

WORCESTER POLYTECHNIC INSTITUTE

Facilitating Multidimensional Analysis of the Merton Partnership's Local Intelligence Data

An Interactive Qualifying Project sponsored by the Merton Partnership

June 28, 2008

Submitted by: James Thompson, Eric Andresen and Joanne Daniel
le8-mepa@wpi.edu

Submitted to: Project Advisors: Steven Taylor and Nikolaos Kazantzis
Project Liaison: Dr. Norman Urquia

Authorship

This project was completed with the equal participation of all members. Without full cooperation and effort provided by each team member, this project could not have been successfully completed.

Acknowledgements

- We would like to thank our friends and families whose support enabled us to take advantage of this wonderful opportunity.
- We would also like to thank our advisors, Steven Taylor and Nikolaos Kazantzis, our ID 2050 professor Fabio Carrera, and the staff at the Gordon Library for their assistance along the way.
- Finally, we would especially like to thank our liaison Norman Urquia and everyone in the Merton Partnership for welcoming us and taking the time to make this project possible.

Abstract

Formed in a merger between one of the city's most affluent wards and its relatively deprived neighbor, the London Borough of Merton is now home to a myriad of dynamic social issues. As their constituencies evolve, local authorities and nonprofits in the area must understand and adapt to these changes. Members of the Merton Partnership, an alliance of these government and nongovernment agencies, collect the local intelligence data needed to inform their understanding of local issues. To distribute and apply these data effectively, the partnership has sponsored the development of a multivariable data display system that allows its users to dynamically analyze and explore local intelligence.

Executive Summary

Within its five-mile radius, the London Borough of Merton comprises communities so diverse that some residents of the western side are expected to live an average of more than five years longer than residents of the eastern side. To address such social issues, local government agencies have allied themselves with local nonprofits in a group called the Merton Partnership. With an action plan addressing issues such as health, safety and economy, the partnership needs well-informed members to make key decisions.

Before this project, the partners' only method of accessing their counterparts' local intelligence data was through pages of static documents hosted on their website. These repositories rarely incorporated visual elements more sophisticated than one or two dimensions, and their interactivity was limited to a site-wide search function. Due in part to the difficulty of updating this system, its sources were also limited, often outdated and never displayed in comparison with one another.

To address the partnership's issues, this project has designed and developed a system that enables users to more easily submit, access and analyze data. The team assessed the potential users' needs and the constraints for a system's development, discovering a demand for dynamic, multivariable comparisons of geographic data stratified by the borough's electoral wards. Following input from the partners, we designed and developed a display system capable of dynamically varying visual elements of a map of the borough, representing up to three datasets from an easily updatable database. This new system can enrich partners' understanding of issues affecting their constituencies as they explore local intelligence data in new ways.

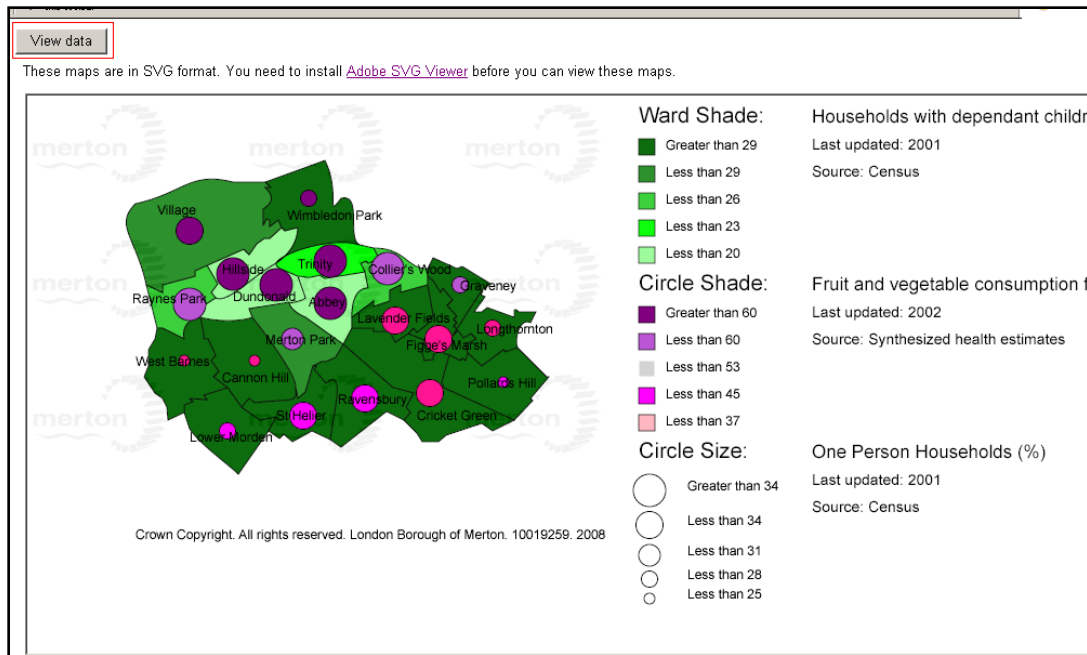


Figure 1 Multivariable displays by ward

In user tests of the system's prototype, partners were already discovering new trends in existing data. The multidimensional data displays we developed will reveal additional complexities in Merton's populations, beyond the static dimensions to which their data displays were once limited. When used properly, our system will allow partners to test hypotheses or seek explanations for aberrations in other datasets, motivating new action toward the betterment of Merton.

Table of Contents

Authorship	2
Acknowledgements	3
Abstract	4
Executive Summary	5
Table of Contents	7
Table of Figures	11
1 Introduction	12
2 Background	14
2.1 The Merton Partnership: Merton’s Local Strategic Partnership	14
2.2 Government organizations in Merton	15
2.2.1 Merton Council	15
2.2.2 Merton emergency services.....	15
2.2.3 Jobcentre Plus.....	16
2.3 Nonprofit organizations in Merton	16
2.3.1 Merton Voluntary Service Council	16
2.3.2 London South Learning and Skills Council	16
2.3.3 Merton Race Equality Partnership	16
2.3.4 The Interfaith Forum	16
2.4 The Community Plan: the partnership’s goals for development	16

2.5	The London Borough of Merton and its social issues.....	17
3	Methodology	19
3.1	Determining needs and constraints for data access.....	20
3.1.1	Identifying the stakeholders and selecting their representatives.....	20
3.1.2	Interviewing representatives of the stakeholders.....	22
3.1.3	Analyzing interview results	25
3.2	Developing a data display system.....	26
3.2.1	Constraining the system's functionality.....	26
3.2.2	Designing the dynamic visual elements.....	27
3.2.3	Collecting data.....	30
3.2.4	Organizing datasets.....	31
3.2.5	Designing the viewers' interface	34
3.2.6	Selecting development tools.....	36
3.2.7	Developing the system	37
3.3	Ensuring the effectiveness of the system	38
3.3.1	User-testing the system	38
3.3.2	Evaluating the system.....	39
3.3.3	Documenting the system	39
4	Data and Analysis.....	40
4.1	Interview findings and applications.....	40

4.1.1	Data interest	40
4.1.2	Feature interest	42
4.1.3	Data possession.....	45
4.1.4	Potential conflicts with existing projects.....	46
4.2	Testing session findings and applications.....	46
5	Conclusions	48
5.1	Accomplishments	48
5.2	Issues and recommendations	48
5.2.1	Expanding supply of data	49
5.2.2	Expanding the system's functionality.....	49
5.2.3	Checking validity of data.....	50
5.3	Implications	50
6	Works Cited	52
Appendix A	Lists of Interviewees.....	54
Appendix B	Documentation: Using ward profiling.....	55
	Introduction page	55
	Display system.....	55
	Update system	58
Appendix C	Code.....	61
	Introduction.aspx.....	61

WardProfile.aspx..... 62

Update.aspx..... 102

VillageDraw.aspx 113

GraphDraw.aspx 130

Table of Figures

Figure 1 Multivariable displays by ward.....	6
Figure 2 Social divisions in Merton.....	18
Figure 3 Timeline of project process	20
Figure 4 Standard decision-maker interview outline.....	23
Figure 5 The system’s available variable visual elements.....	28
Figure 6 Multivariable displays by ward.....	29
Figure 7 Variable bar charts by ward.....	30
Figure 8 Organization of datasets in template.....	33
Figure 9 Data submission and deletion on the update system	34
Figure 10 Interface elements of the system	35
Figure 11 Data interest by number of interviewees	41
Figure 12 Feature interest by number of interviewees	42
Figure 13 Interviewees possessing relevant data.....	45
Figure 14 Partnership decision-maker interviewees.....	54
Figure 15 Council resource manager interviewees	54

1 Introduction

In a 2006 *Teradata Magazine* survey of business executives in more than 100 countries, “45% of respondents said decisions are becoming more complex, 68% said the number of daily decisions has increased over the last year, and 85% agreed that decision-makers need more up-to-date information than in the past” (Terry, 2006). This demand for speedy access to clearly presented data is growing not only in businesses, but in all organizations across the globe. While organizational decision-making has been examined in myriad contexts, none has been as thoroughly analyzed as the highly publicized losses of the American *Challenger* and *Columbia* space shuttles. Speaking from her experience on the investigative boards for both of these incidents, Dr. Sally Ride has argued that the time constraints created by the launch schedule led to contractors’ failure to heed engineers’ hastily prepared warnings about the known safety issues that ultimately contributed to the disasters: “[E]ngineers found themselves having to prove that there'd be catastrophic consequences in order to be listened to” (Dreifus, 2003). Information design expert Edward Tufte expands the argument that organizational decisions can be blamed for the disasters: he argues that because engineers’ risk assessment reports failed to deliver the data that caused their alarm, managers were not convinced to act upon their recommendations (Tufte, 2003). Had managers within NASA and its contractors been offered a more effective display of the relevant data, these tragedies might have been averted. Organizations require efficient access to essential data to make the effective, evidence-based decisions that best serve their constituencies.

The London Borough of Merton is home to a population that spans demographics and, within a five-mile radius, includes some of the most deprived and the most prosperous areas of London. As such, the Merton Partnership, an alliance of local service organizations, faces complex sets of data when making decisions about the borough. The partnership has tried to make it easier for their councilors and officers to locate data by launching its own website in the Spring of 2007. With this move, members have

begun to utilize the internet as a tool for communication and information sharing, but the data each group possesses remains in formats unfriendly to its users.

At the time of the website's launch, its sources for local area information offer only static, non-graphical and one- or two-dimensional depictions of existing datasets. Since its launch, few of these information sources have been updated. The information display system was underdeveloped compared to modern systems for other regions, making it difficult and time-consuming to search through the site to find the specific sets of information needed for decision-making. Because partners' understanding of these datasets informs their decisions, any limits on their ability to access and analyze data in turn limit their ability to serve their constituencies with optimally informed decisions.

The partnership has attempted to address this issue in the past by introducing an information library and ward profile system on their website, as well sponsoring another group of WPI students to create a collection of maps and graphs using 2001 census data. While these approaches offered data-rich sources to assist partnership members, they are difficult to navigate and compare and are becoming dated, as more than seven years have passed since the last census. To overcome these issues, we will need to develop a data display system that is easily operable and expandable.

This project intended to help the Merton Partnership to coordinate the delivery of its members' services by assessing distribution habits and access needs for local intelligence data and introducing a system that improve access to the data partners need to inform their decision-making. The team will determine the needs and constraints for partnership members' access to local intelligence data, develop a system that stores and displays the data in accordance with users' needs and ensure the lasting effectiveness of the system.

2 Background

To design an effective solution to the Merton Partnership's needs, our team familiarized itself not only with the partnership and the community it serves, but also the types of organizations that make up the partnership. These groups consist of nonprofit and governmental organizations that intend to address the social issues in the borough. During the course of our project, we utilized our knowledge of these organizations and their agendas to ensure that our product properly targets these groups.

2.1 The Merton Partnership: Merton's Local Strategic Partnership

In the late 1990s, as the English government worked to improve the quality of life in 88 of its most deprived neighborhoods, it developed a grant called the Neighborhood Renewal Fund (NRF), which was intended to provide communities with the resources needed to alleviate deprivation. To oversee the effective application of NRF resources, the Local Strategic partnership (LSP) program was developed. After establishment of an LSP became a prerequisite for an area's receipt of NRF resources, 88 of these local programs emerged (Neighbourhood Renewal Unit, 2005).

The success of the LSP program in these regions prompted the Local Government Act of 2000, which encouraged the spread of the LSP program beyond NRF areas. It emphasized the program's potential for uniting local organizations with similar aims by developing a sustainable plan around which the groups could operate. An LSP now exists in almost every region of authority in England and Wales.

While each region's LSP may include different member organizations and therefore exist in a different form, all share a common structure for operation: the partnership is led by the local council and holds meetings between representatives of all partners at regular intervals. In addition to the meetings of the partnership as a whole, the group establishes separate teams to handle specific issues, inviting the stakeholding partners to attend. The decisions made within team meetings have a significant impact on local governments' decisions regarding resource allocation and policy decisions, as members

of the partnerships are active and influential members of the community. Recent legislation has bestowed the partnerships with even more sway over community decision-making, as Local Area Agreements now transfer authority over financial and policy decisions from traditional governing bodies directly to the partnership (Neighbourhood Renewal Unit, 2005).

Merton's LSP, the Merton Partnership, consists of the following parties: the leader of Merton Council, the Chief Executive of Merton Council, Member of Parliament (MP) for Wimbledon, MP for Mitcham and Morden, the Borough Commander of Merton Police, the Chair of Merton Chamber of Commerce Jobcentre, the Chair of Merton Voluntary Service Council, the Chief Executive Sutton and Merton Primary Care Trust, the Principal of Merton College London South Learning and Skills Council, the Chair of Merton Race Equality Partnership, and the Interfaith Forum Merton Unity (Idea Knowledge, 2007).

2.2 Government organizations in Merton

Within Merton, government organizations work closely with the partnership to address local issues. The local governing body and service departments focus on various aspects of community life (Merton Partnership, 2007).

2.2.1 Merton Council

The governing body of the Borough of Merton is composed of three councilors from each of the 20 wards. It is led by Councilor David Williams, who oversees the function of its five component departments: the Chief Executive's Department; Corporate Services; Children, Schools and Families; Environment and Regeneration; and Community and Housing.

2.2.2 Merton emergency services

In addition to performing their duties within the borough, the police force and fire brigade also work to develop strategies for reducing crime and making Merton a safer place. Currently, they focus on issues such as vandalism, drug misuse, and antisocial behavior.

2.2.3 Jobcentre Plus

As a part of the Department of Work and Pensions, Jobcentre Plus works to pair job seekers with employers who are recruiting. The organization offers resources to both parties, with the objective of reducing unemployment.

2.3 Nonprofit organizations in Merton

Nonprofits are non-government organizations whose objectives center on social benefits. While such groups exist at the international and national levels, several operate locally, focusing on improving the quality of life Merton (Merton Partnership, 2007).

2.3.1 Merton Voluntary Service Council

A registered charity, the Merton Voluntary Service Council assists volunteer organizations by enabling representation and strategic partnerships within the community. It is a member of a national network of similar organizations.

2.3.2 London South Learning and Skills Council

The Learning and Skills Council is designed to enable those without university educations to effectively meet the needs of employers. In addition to advocating for higher quality of education, the council also plans and funds training for job seekers.

2.3.3 Merton Race Equality Partnership

With the support of stakeholders in sectors such as education, police, religious, and health services, this organization seeks to promote equal opportunities for and cooperation between people of all racial groups.

2.3.4 The Interfaith Forum

The Interfaith Forum is an alliance of representatives from various religious faiths. They work to promote community safety, peace and cohesion.

2.4 The Community Plan: the partnership's goals for development

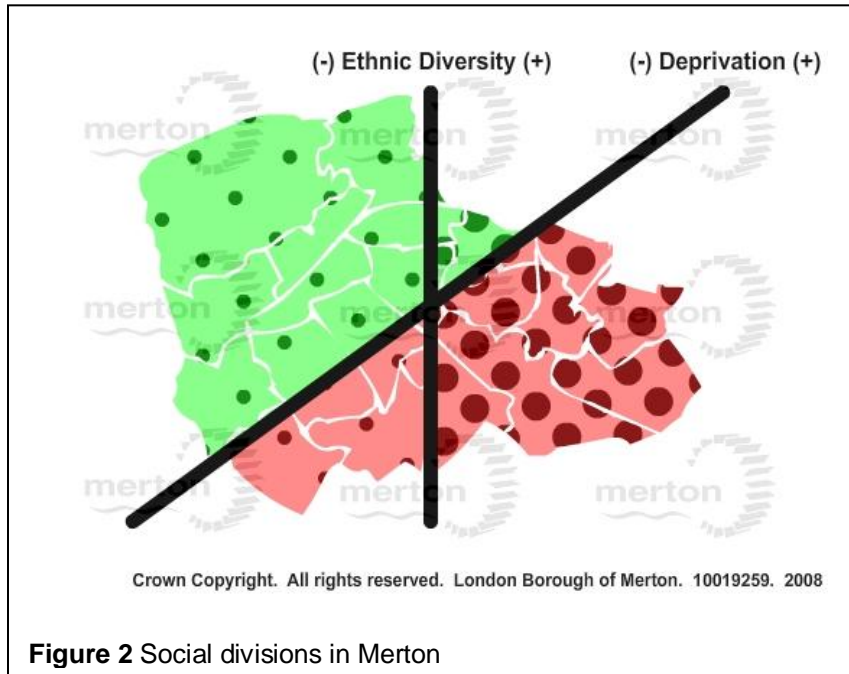
Together, these groups intend to alleviate such problems as limited employment opportunities, poor public health trends, high crime rates, and environmental,

educational and housing problems. To address these issues, the partnership outlined a Community Plan for 2006 through 2015.

The Community Plan acts the partnership's "blueprint" for addressing the needs of the community in five themes: Children and Young People, Older People, Sustainable Communities, Healthier Communities, and Safer and Stronger Communities. In each of these domains, the partnership outlines specific goals for the nine-year period. These goals include benchmarks in efforts to improve areas: to improve housing by offering homes of higher quality and better affordability; to improve the environment by reducing emissions and increasing recycling, as well as increasing community satisfaction with parks and open spaces; to improve the economy by increasing employment and reducing the number of working-age people claiming benefits; and to improve transportation by increasing the usage of sustainable transportation.

2.5 The London Borough of Merton and its social issues

In 1965, the affluent Municipal Borough of Wimbledon merged with its more deprived neighbor of Mitcham, to create Merton in its current form. The socioeconomic divide between the two regions lived on, and it remains a major issue for the borough. Using 2001 census data from the region, Yorkshire and Humber Public Health Observatory has indexed deprivation in the region, using criteria including income, health, education, housing and crime. Indices in parts of southeast Merton range are among the highest in London, while most of the northwest region remains comfortably among the lowest (Indices of Multiple Deprivation 2004). Among the most shocking divides between the two regions is the life expectancy gap: those living in Merton's top four wards are expected to live 5.2 years longer than those in the lowest four wards (Bajekal, Sholes, Pickering, & Purdon, 2002).



While severe differences in deprivation divide the borough diagonally, differences in ethnic diversity levels create a similar, but vertical divide. Census 2001 reports that around a quarter of all Merton residents have an ethnic minority background, with as much as half of the population of some

eastern wards belonging to these groups (Office of National Statistics, 2001). Other data also report rising numbers of residents whose native language is not English. For example, the already significant numbers of Polish-speaking schoolchildren have doubled since the 2001 census (Merton Council, 2007).

The partners' understanding of these issues shapes the Community Plan and thus the operation of the component groups of the partnership. This collective understanding is informed by individuals' access to local intelligence data, the sources of information that reveal momentary trends in local social issues.

3 Methodology

The Merton Partnership formed as a strategic partnership of local groups aiming to improve the quality of life in the London Borough of Merton. This project intended to help the Merton Partnership to coordinate the delivery of its members' services by assessing distribution habits and access needs for local intelligence data and introducing a system that improves access to the data partners need to inform their decision-making.

To accomplish this mission, we divided the project into three stages, which focused on achieving the following goals:

1. To determine the needs and constraints for partnership members' access to local intelligence data;
2. To develop a system that stores and displays the data in accordance with users' needs; and
3. To ensure the lasting effectiveness of the system.

Each of these objectives comprised several tasks completed during the course of our project. For our first objective, we identified and interviewed stakeholders in the project in search of system requirements and data sources, organizing our findings based on our intended use for them. We addressed our second objective by designing and developing a dynamic web page allowing users to select up to three datasets to be compared on variable visual elements of a map of Merton's wards. For our third objective, we tested our system with our original interviewees to confirm that it met their needs, revising it until they were satisfied. We also documented the system to ensure that independent users could adopt it.

We conducted this project in the London Borough of Merton in the seven weeks preceding 28 June, 2008. The following timeline outlines our progress during those weeks:

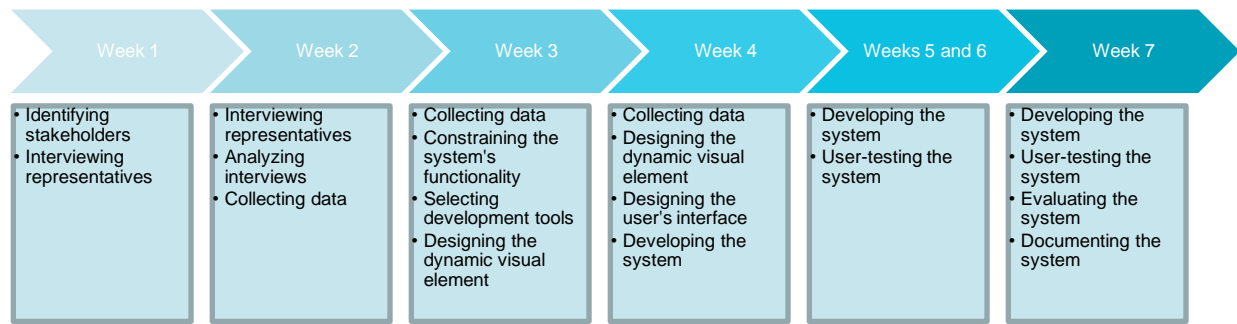


Figure 3 Timeline of project process

We conducted this project in the London Borough of Merton. Because the Merton Partnership exists only as a relationship between its component groups, it has no resources of its own; therefore, we utilized those of one of its chief members, the Merton Council. We conducted most of this project from their facilities in the Merton Civic Center in Morden, Surrey, UK. The final system we developed resides on the council's servers, but is accessible not only to all partnership members, but also to the public.

3.1 Determining needs and constraints for data access

Before developing a solution for the partnership local intelligence needs, we first determined the requirements for the system. To accomplish this objective, we identified the stakeholders in this project, interviewed representatives of each group and analyzed our findings.

3.1.1 Identifying the stakeholders and selecting their representatives

To determine the requirements for our solution, we first identified groups with viewpoints relevant to its development. These stakeholders included all decision-making members of the partnership and council members with constraints on our use of its resources. After identifying the groups in each of these categories, we selected key individuals that would together represent the entire group's views. To create such a sample, we

needed to identify and represent subgroups existing within each of these groups (Bryman, 2001).

3.1.1.1 Identifying the partnership's decision-makers

We stratified the Merton Partnership twice before selecting interviewees to represent its decision-makers. As our mission was to assist the entire partnership in making decisions, we first stratified its membership by the five themes composing its decision-making framework, the Community Plan: Healthier Communities, Safer and Stronger Communities, Older People, Children and Young People, and Sustainable Communities. We also stratified partners by their organization type, including government and non-government organizations. While the council's significant role within the partnership necessitates that many of our interviewees belong to that organization, we represented other groups in our sample when possible.

To select interviewees who could represent each of these groups, we consulted an expert on both qualitative data analysis and the Merton Partnership's membership, Dr. Norman Urquia. He provided a list that included officers from each of the partnerships developed under the Community Plan. He also recommended that we interview additional council staff members who oversee the work of the rest of the council. While these members do not represent a specific Community Plan theme, their work in areas like performance management have familiarized them with the tasks and needs of partnership members across all themes.

3.1.1.2 Identifying the council's resource managers

As we intended to utilize the Merton Council's resources to develop the solution, we needed to interview representative managers of these resources to identify potential constraints. We identified two groups within the council that could such constraints: information technology staff who could assist in our system's creation and members of teams assigned to other data access management projects. Using the staff directory for the council, we found information technology specialists in web content management, databases and Geographic Information System (GIS) mapping. These experts would

be able to identify the constraints of a system such as ours, as it would be developed using their resources. By searching the council website, we also found two existing information access systems. We added their developers to our list of interviewees, as we wanted to ensure that the informational domains did not overlap and to learn methodologies they had found helpful in their work, as these could assist us in completing our second and third objectives.

3.1.2 Interviewing representatives of the stakeholders

After compiling a representative sample of stakeholders, we interviewed them to determine their needs and constraints. We began by interviewing the group of potential users, as an understanding of their needs would help us during our interviews with the second group, who would help define our constraints.

3.1.2.1 Planning interviews with decision-makers

To guide us in collecting similar data from each potential user, we outlined a standard interview. Its format follows:

1. Introduce the group and share our intentions.
2. Ask the interviewee(s) about their role within the partnership.
3. Question the interviewee(s) about their experiences accessing information from other partnership members.
4. Determine what types of information they would like to access.
5. Determine the format in which they would like to see this information.
6. Determine if they have any information to share.
 - a. Determine if they already share this information.
 - b. If they have information but do not share it, determine what prevents them from doing so.
7. Question the interviewee(s) about any additional comments or concerns about the system.

8. Thank them for their time. Let them know that we will contact them again and how to contact us.

Figure 4 Standard decision-maker interview outline

In steps one through three, we intended to make the interviewee think about data access and what role it plays in their work. In steps four and five, we hoped to learn what information would be best to include in our system. In step six, we hoped to find sources of data to use for the system. Steps five, six and seven may also add constraints to our system in the form of special formatting needs, barriers to information sharing and additional concerns about the system.

3.1.2.2 Planning interviews with managers

Unlike the interviews with potential users of the system, our interviews with managers did not share a standard structure. Instead, we would outline a rough idea of the form we envisioned our solution would take, using the findings from other interviews. We would then ask each interviewee whether they could imagine any concerns with it.

When interviewing managers of existing data access projects, we would explain our respective projects and identified similarities. We intended to discover not only overlaps in content, but also information and methodology used in existing projects that could be relevant to our own. If possible, we intended to conform to existing standards of information display to enhance the usability of our system; common interface elements or information categorization practices could improve the system's usability for users familiar with other systems developed by the council.

When interviewing the managers of the council's information technology resources, we planned to begin by asking what resources were standard for use at the council in several areas: any web content management systems or cascading style sheets that would allow us to automatically conform to the council's established visual identity, server space and a database on which to store the data necessary to drive our display system, popular end-user data storage and manipulation tools on which existing data may be stored, and mapping software we could use to develop our solution. Because

this solution would ultimately reside on council hardware, we would need to conform to the council's policies and standards for visual identity, accessibility and software compatibility. We also intended to encourage continuing communication with these experts on such policies, since we anticipated that our solution could develop new conflicts as we altered its design.

3.1.2.3 *Requesting interviews*

After preparing our questions, we contacted each of our representatives to request a brief, informal meeting to discuss their experiences with and desires for information access within the partnership. We began by using email, as it allowed the representative to respond when time permitted. If an email yielded no response within two days, we contacted them by telephone to schedule the interview, as this method demands more immediate attention than an email. If we made contact but were unable to schedule an interview, we attempted to conduct the interview by telephone while we had the representative's attention. By using multiple methods of contacting representatives, we hoped to increase the rate of response (Valenzuela & Shrivastava, 2002).

3.1.2.4 *Conducting Interviews*

When a representative agreed to an interview, we used Microsoft Outlook contact management software to determine the representatives' availability and to schedule interviews with them, as was standard within the council. We considered standardizing the setting to add uniformity to the interviews, but opted to give our interviewees freedom to choose the time and setting instead. This would increase their comfort level and encourage their participation.

All three of our group members attended each interview. While one conversed with the interviewee following our planned procedure, the other two members took notes about every response the interviewee made (Bryman, 2001).

3.1.3 Analyzing interview results

Following each interview, our team reviewed the results. Together, the two note-takers created a typewritten summary of their notes on each interviewee's responses. In the same document, we also noted interesting or unique issues that arose during the interview. This process allowed us to consider the outcomes of individual interviews before summarizing our findings in the overall analysis. We also used these summaries to recall users' specific concerns when we contacted them later.

3.1.3.1 Analyzing interviews with decision-makers

After completing all interviews of potential users' representatives, we combined their individual summaries into three tables of findings. Two tables tallied the number of interviewees expressing interest in a specific type of data or requesting features of the solution. The third table included an initial list of potential data sources as mentioned in the interviews, including both interviewees who offered data they possessed and those who mentioned an interest in data they alleged to be held by other groups.

These tables dictated the priority for the inclusion of data and features in the solution we develop. As our mission was to meet the data access needs of all partners represented in these interviews, we weighted each interviewee's opinion equally. For example, a type of data with twice as many tallies as another would receive twice the priority as we developed the system.

3.1.3.2 Analyzing interviews with managers

Unlike our analysis of the interviews with potential users, we did not combine the results of our interviews with managers; instead, we used them individually while planning the next phase of our project, the development of the information display system. We used resource managers' comments to constrain ourselves to the standards and policies they presented. Concerns of managers of other projects helped us set the domain of our project, helping us avoid interference with their work

3.2 Developing a data display system

With an understanding of potential users' needs and managers' constraints, we designed and developed a system that would store the data we had collected and display it in useful ways. Using our findings from the interview analyses, we determined the domain of our project and designed a system that would fulfill the interviewees' needs while meeting their constraints. With the assistance of the managers we interviewed, we then selected tools that would enable us to realize our design. Finally, we developed the system based on our design.

3.2.1 Constraining the system's functionality

With a time constraint of seven weeks and the intent to design a simple and approachable system, we needed to establish a scope for our system's functionality. Using our analysis of interviews with decision-makers, we determined a scope for the data we could provide for these partners.

Data collectors have controlled the detail level of their data by dividing London boroughs into regions of varying size: community forums, wards, super output areas and postal code regions. Some data also exist at the level of the entire borough. By comparing our interviewees' needs and constraints with the availability of data at these levels, we selected the electoral ward as an optimal detail level. As the council operates on this level, many existing datasets and decision-makings frameworks have been constructed at this level. The ward level also allows us to avoid the privacy concerns of displaying very small regions without compromising the users' ability to recognize trends within the borough.

We also determined that because our solution would need to dynamically process new data, all data would need to exist as a set of 20 numeric values, with one for each ward. Requiring numeric values not only ensures that data are objective measures of real phenomena, but it also allows us to use standard graphical displays to facilitate comparisons between all datasets.

3.2.2 Designing the dynamic visual elements

The key element in our system is the display system. Because users' needs demanded geographic displays, we chose a map of the boroughs wards for our primary display. To simplify the image and prevent confusion between its variable elements, we included no landmarks or other details within the regions. Instead, accompanying documentation referred users unfamiliar with the borough's geography breakdown to more detailed static maps.

To represent our datasets, we needed elements on the display system to change in relation to an individual ward's value within the range of the dataset. Since our interviewees requested comparisons, we aimed to include at least two datasets; allowing users to choose more would increase the system's usefulness, but only if the variable visual element was distinct enough to be easily detectable when displayed simultaneously with all the other visual elements.

To select the variable elements we would use, we listed common variable visual elements seen in other dynamic display systems: size, shape, position, color, shade of color and texture (Gapminder, 2007). As we used geographic relationships to position the ward, we cannot use it to vary these visual elements. Similarly, we cannot alter the size or shape of the elements, as these properties identify them as wards. We also eliminated texture, as the development tools offered by our resource managers could not easily change the texture of a shape. By changing the color of a ward on a gradient, we could effectively signify the variance of one dataset. Because our tests of this method proved so effective, we considered the possibility of using multiple colors per ward. If we striped the wards in alternating, contrasting colors, we could then change the shade of the colors to signify the variance of two variables. However, striping in three colors yielded less distinct results.

Because striping would prevent us from introducing a third variable, we used an alternative. Instead of introducing a different color with striped wards, we added a symbol to the center of each ward. Like the stripes, we colored the symbol to contrast

with the background, while varying the shade of that color to indicate changes in the corresponding dataset.

In addition to shade, this symbol could be manipulated in ways that the ward region could not, allowing the possibility of depicting more than two datasets. We eliminated position, as it was used to indicate the ward to which the symbol belonged. Because we intended to display quantitative data along a range of values, we also eliminated shape. While creating shapes that varied in a progression from one value to another might be possible by changing its number of sides, this progression was difficult for our test viewers to recognize at a glance. To avoid interference with the other visual elements, we formed the symbol into the simplest shape possible, a circle. Size, however, could be changed dynamically, creating a third variable visual element that could be displayed simultaneously with the other two.

This left us with the following variable visual elements:

1. **Variable ward color:** To represent our first dataset, we varied the shade of the background color of each ward. Lighter values would indicate values nearer to the start of the range, while darker values would indicate the values nearer to its end.
2. **Variable symbol color:** To represent a second dataset, we placed a circle within each ward and colored it with a color that contrasted with the ward color.
3. **Variable symbol size:** To represent a third dataset, we varied the size of that circle. Smaller circles would indicate values nearer to the start of the range, while larger circles would indicate the values nearer to its end.

Figure 5 The system's available variable visual elements

In all three variable visual elements, we divided the variance into five stages, either color shades or circle sizes. Together, these stages represented a steady change from light to dark or small to large. Each of these stages corresponded to a range of values in the dataset, as defined by the set's contributor in the spreadsheet. While this required the contributor to select ranges that create accurate representations of the

data, it also enables that contributor to adjust values as needed, compensating for outliers in a way that a system with automatically generated ranges could not. For datasets with regular distributions or users uncomfortable with manipulating these ranges, our templates include default values calculated by spreading the ranges evenly between the dataset's minimum and maximum values.

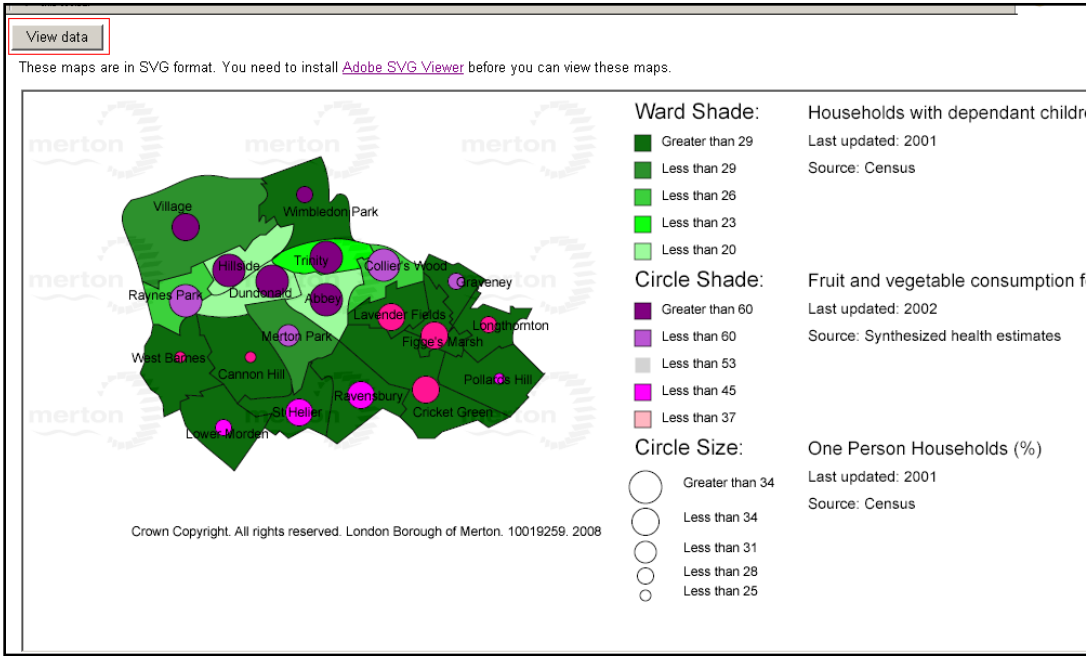


Figure 6 Multivariable displays by ward

In addition to the primary geographic displays, we also included bar charts of the data to allow visual comparisons of the exact values without the geographic element. We colored the bars were colored with respect to the datasets' colors in the map to create an easy transition between the two. As we intended this display to enable more detailed comparisons, we labeled each bar with its exact value instead of labeling the numeric axis. To ensure readability of the bar labels, we used horizontal bars in favor of vertical ones; this allowed space for the name labels of each bar to appear horizontally as well.

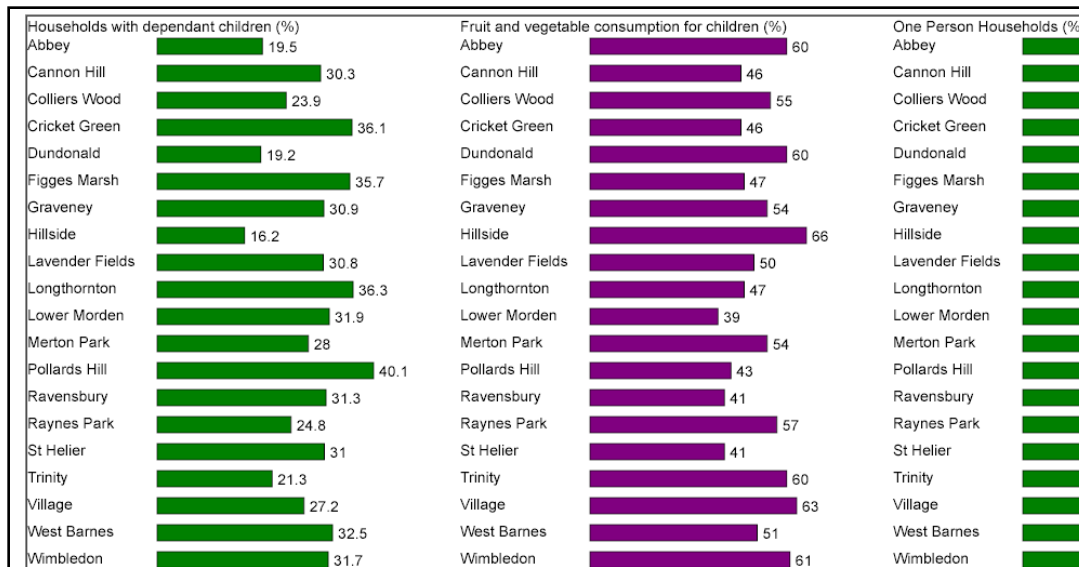


Figure 7 Variable bar charts by ward

Finally, we needed to select two contrasting colors for the ward background and the symbol. In doing so, we referred to our analysis of a manger’s interview, which stated that in order to place our system on the council’s website, we would need to make it accessible to people with visual impairments. To fulfill this requirement, we needed to select colors that clearly contrasted, even for those with low vision or color blindness. Using a tool to convert colors pairs to their equivalent for color blind viewers, we assessed the various pairings of primary and secondary colors and selected several color candidates, including green, blue, purple, red and gray. Through user tests, we found no patterns of preference between the different sets, though some users held individual preferences. To respond, we allowed users to select a color for each of the two colored variable elements. This would not only allow users to make aesthetic decisions while viewing and publishing their maps, but it also allows users with limited vision to select the sets best suited to their needs.

3.2.3 Collecting data

With the system designed, we needed data to populate it. We considered two possible methods of data collection: in the first, we would collect any data we could find about the borough and include as much as possible in our solution; in the second, we would

collect only data requested by the interviewees. Because our system would need data in a standardized form to make the dynamic comparisons requested by interviewees, we would have to limit the contents in some ways, including both the format of the data and the restriction to ward-based data divisions we already established. While this ensures that our solution can dynamically handle new datasets, it does not limit the types of issues the data can describe.

After choosing constraints, we conducted our search for data, much like our search for interviewees. We emailed representative of groups our interviewees thought possessed data, followed by telephone calls to those who did not respond. Upon contact, we requested any shareable data meeting our constraints. For the initial population of the database, we accepted data in any form. Later, our system would require a standard form of input to handle submissions automatically.

3.2.4 Organizing datasets

To add the data to our display system, we needed to organize our datasets into a standard form to allow them to be easily or automatically added to a database. To satisfy this requirement, we placed each dataset into a single row in a spreadsheet containing all of the information our system would require. For this task, we used Microsoft Excel spreadsheet software, the council standard.

With values established for each of the datasets, we could offer template spreadsheets to data contributors who wished to continue to update and expand the display system's database. Instead of distributing them before the initial population of our database, our group formatted the data into these spreadsheets. We decided that this would reduce contributors' initial commitment and allow us to showcase the working system to motivate users to independently format and contribute data. After placing a live version of the system on the council website, we distributed templates to current contributors and made them available for future contributors.

Using our interview analyses and our design for the display, we determined the necessary fields of information for each dataset:

- **Dataset title:** This field identifies the set of data in its row. The system uses it both as a key, or unique identifier for the dataset, and as a string of text allowing the end-user to identify and select it (Chapple, 2008). This title must clarify the format of the data in this set.
- **Ward data:** Each of these 20 fields contain a number, constituting the data in the dataset. As they are be used by the system to generate graphics, these fields must contain only numeric values. Any clarifications about their format must be made in the title field.
- **Data ranges:** These four fields allow the contributor of the data to specify the range into which the data falls. These numeric values are used by the display system to categorize the data into comparable ranges.
- **Source:** As many interviewees requested source information to allow them to confidently reference data in reports, these fields are included for each dataset. Each will contain a text string that will allow the reader to identify the source.
- **Year:** Like the source field, the year provides the user with information about the source of the data. The year can act as a caveat against outdated data and data which may be incompatible for viable comparisons. Each field contains a single four-digit number representing a year.

	A	B	C
1	<i>Dataset title</i>	Fruit and vegetable consumption for adults (%)	
2	<i>Abbey</i>		31
3	<i>Cannon Hill</i>		28
4	<i>Colliers Wood</i>		30
5	<i>Cricket Green</i>		26
6	<i>Dundonald</i>		31
7	<i>Figges Marsh</i>		28
8	<i>Graveney</i>		34
9	<i>Hillside</i>		37
10	<i>Lavender Fields</i>		26
11	<i>Longthornton</i>		29
12	<i>Lower Morden</i>		25
13	<i>MertonPark</i>		33
14	<i>Pollards Hill</i>		26
15	<i>Ravensbury</i>		25
16	<i>Raynes Park</i>		31
17	<i>St. Helier</i>		24
18	<i>Trinity</i>		29
19	<i>Village</i>		41
20	<i>West Barnes</i>		29
21	<i>Wimbledon</i>		33
22	<i>Range 1</i>		23
23	<i>Range 2</i>		27
24	<i>Range 3</i>		31
25	<i>Range 4</i>		35
26	<i>Source</i>	PCT synthetic health estimates	
27	<i>Year</i>		2002
28			

Figure 8 Organization of datasets in template

To submit these templates, we created a password-protected page available only to partnership contributors. It allows them to select an Excel file from their local network to add to the database. When an Excel file is submitted to the database, any datasets it contains will replace existing ones with the same title. Datasets with new titles will expand the database. Administrators may also remove datasets from the system.

Please Enter the path where the Excel sheet is located: Delete statistic:

C:\PCT_data_example.xls .Please Select One...

Is this the correct data?

Information	Fruit and vegetable consumption for children (%)	Fruit and vegetable consumption for adults (%)	Estimate of smoking (%)	Estimate of binge drinking (%)	Estimate of obesity (%)
Abbey	60	31	26	19	17
CannonHill	46	28	20	15	21
ColliersWood	55	30	28	17	19
CricketGreen	46	26	33	12	23
Dundonald	60	31	24	22	16
FiggesMarsh	47	28	31	12	22
Graveney	54	34	22	12	20
Hillside	66	37	22	21	14
LavenderFields	50	26	30	15	21
Longthornton	47	29	23	11	22
LowerMorden	39	25	22	14	22
MertonPark	54	33	17	15	17
PollardsHill	43	26	25	12	23
Ravensbury	41	25	30	13	22
RaynesPark	57	31	21	16	16
StHelier	41	24	32	13	22
Trinity	60	29	26	22	16
Village	63	41	13	14	13
WestBarnes	51	29	19	14	19

Figure 9 Data submission and deletion on the update system

3.2.5 Designing the viewers' interface

The user interface allows a viewer to interact with the system and controls what the user is able to display with it. Before we could create the system, we needed to design the interface in a way that would satisfy users' needs. To accomplish this task, we referred to our analysis of interviews with potential users. Because our project intended to ensure that the system is inviting and accessible for a broad range of users, we needed to ensure that the interface was simple and intuitive.

Referring to our interview analyses and our display design, we began by listing the elements we needed to include in the interface:

- **Dynamic geographic display:** This display depicted regions with variable visual elements capable of indicating variation in up to three datasets.
- **Non-geographic display:** These displays chart each dataset on a different graph, with each ward on a different bar.

- **Dataset range keys:** For each dataset used as a variable visual element, we needed a key representing the how that visual element would change as the variable changed.
- **Dataset selection menus:** To allow the user to select datasets for each of the variable visual elements, we provided three dropdown menus populated with a list of all data currently in the system. These datasets were named using the database entry containing the set's information title.
- **Color selection menus:** We will include one dropdown menu for each shaded visual element, allowing users to select its color.
- **Add/remove dataset buttons:** Because a comparison using all three available datasets would not suit all situations, we wanted to allow users to select the number of datasets included in the display. While it would have been possible to keep all three dropdown menus visible at all times, this could unnecessarily encourage the use of all three. By requiring users to add new datasets individually, the system forces them to consider whether they need additional datasets instead of asking which new datasets they need.

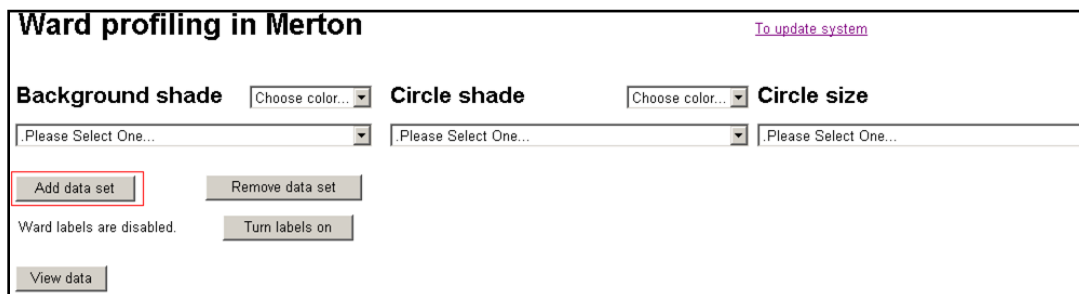


Figure 10 Interface elements of the system

We decided that the system will never prompt the user to select a variable visual element to associate with a dataset; this will be automatic. Ward shade will always signify the first dataset, shaded dots will appear only if a second set is used, and the dots will only change size if a third dataset is used. This automation simplifies the user's experience, as it selects more intuitive variable visual elements first. A map with one variable should not introduce symbols as new visual elements when it can vary the

existing wards. Similarly, a map with two variables should not introduce size as a new variable element when the viewer is already using color in comparisons for the first set.

3.2.6 Selecting development tools

To begin developing a functional system from our designs, we selected development tools that will enable us to create the three major elements of our system: the web pages, the dynamic displays and the database. Our selection of these tools was heavily constrained by standards maintained by the council. As a government organization, their duty to ensure equal accessibility obligates them to maximize compatibility whenever possible.

Developing the web pages required use of Hypertext Markup Language (HTML), a common language for generating web pages. While HTML allowed us to format text and position controls such as buttons and dropdown menus, we needed a more powerful scripting language to generate the dynamic content our system required. While common scripting languages include JavaScript and PHP: Hypertext Preprocessor (PHP), these require end users to download and update a plug-in. Requiring a user to take this sort of action violates the council's accessibility policies (Chisholm, Vanderheiden, & Jacobs, 1999). To avoid this conflict, we opted to use Microsoft's Active Server Pages (ASP), a scripting engine that runs directly from the server. This method allows developers to create dynamic web pages without requiring viewers to download any plug-ins. While several versions of ASP exist, we used ASP.NET 2.0 to conform to council standards.

While HTML and ASP offer basic image manipulation functionality, we needed a more powerful visual display protocol to develop our dynamic geographic display. We investigated third-party mapping packages such as GeoReveal and MapInfo, but they did not meet our needs. An interview with the manager of GIS systems for the council also revealed that licensing a mapping package for our purposes could cost the council thousands of pounds. Furthermore, the packages would neither allow for the comparison of multiple datasets we required, nor would they allow users to contribute

data to its database as easily as our interviewees demanded. For these reasons, we decided to develop a customized display solution using Scalable Vector Graphics (SVG). SVG is the protocol used for displaying the images generated by the third-party software we investigated.

After selecting these languages, we needed an Integrated Development Environment (IDE) with a graphical user interface to help us edit, compile and test our code. To maximize our productivity and minimize expenses, we selected a free IDE targeted at novice web developers, Microsoft Visual Web Developer 2005. Using this environment, we were able to use ASP to develop our project. The project's front-end, the display seen by the end user, used HTML and SVG. We used C# to program back-end, or the code behind the function of control objects like our buttons and drop-down menus.

Finally, we needed to select a language for the database that holds our datasets. There are a plethora of languages available for implementing a database, ranging from common Structured Query Language (SQL) to the obscure Gellish English. While specialized languages like Gellish English excel in a specific niche of applications like language interpretation, SQL is more suited for general applications. As such, SQL has become a standard in many businesses and organizations, including the Merton Council. For this reason and because our data's simple numeric format requires no special functionality, we selected SQL for use in our project.

3.2.7 Developing the system

Using our design for the system and our selected development tools, we created our dynamic image, the web pages controlling and documenting it and the database housing its data. In this process, we consulted developers' resources and integrated code from open sources. To maintain a standard visual identity across its website, the council uses LiveLink, a content management system that formats pages to standards determined by its web development managers. To conform to these standards, we used default formatting throughout our pages. HTML tags, such as headings, allowed

us to differentiate between types of text without explicitly defining their formats. A complete copy of the code can be found in Appendix C.

3.3 Ensuring the effectiveness of the system

Before publishing our site to the web, we needed to ensure that it fulfilled the needs of our users, met council publishing standards and could be understood by new users. To do so, we tested our system with potential users, seeking approval from the web information manager and write accompanying documentation. As issues arose during these phases, we made necessary revisions to the design of our system.

3.3.1 User-testing the system

Throughout the system's development, we elicited feedback from the same set of representative potential users we interviewed for our first objective. We contacted them using the same methods proven successful during the interviews. To ensure that we were making positive progression from the beginning, we began to schedule these interviews as soon as we had produced a reasonable facsimile of our final system. Though this meant that the system was missing features or data in most of our user tests, we needed to ensure that the features that had been implemented were satisfactory for users.

At the beginning of each test session, we explained which features we had implemented so far and how to use them, in a manner emulating the final documentation. After this explanation, we allowed testers to interact with the system, encouraging them to ask questions and state concerns while doing so. As one group member guided our testers and responded to their questions, the other two members recorded the questions and responses, as we had done in the interview stage. We used this feedback to make alterations to the system to respond to users' needs. Unlike the interview results, we considered user test data individually and attempted to address every concern they presented.

3.3.2 Evaluating the system

In addition to the feedback of potential users, we also sought feedback about the developing system from the managers we had interviewed. Because features of the evolving system would stray from the design we had originally posed to this group, we needed to ensure that major changes continued to comply with council policies and standards. Much like the user testing, we conducted these feedback sessions by presenting the manager with a working system and explained any new features. Based on their feedback, we would then redesign any new features with conflicts.

3.3.3 Documenting the system

Once the final system had been developed, tested and placed on council web space, we wrote the documentation that would accompany it on the website. In these documents, we included the explanations we used to introduce the system to testers, along with answers to any questions they asked. Throughout the documentation, we used descriptive, task-oriented headings and step-by-step lists to allow users to skim for solutions to individual issues. We intended this documentation to be sufficient guidance to introduce a new user to the system. Documentation would also allow council members to maintain the system after its launch. A copy of this maintenance documentation can be found in Appendix B.

The data viewer's documentation outlined why and how to use the system. In the first section, we introduced the system and its features, giving examples of applications for its functionalities. We then explained how to use each feature of the system, following the logical order in which most viewers would use them.

Similarly, the data contributor's documentation outlined how to add data to the system. We began by explaining how the database is structured and the importance of this standard. Next, we explained how to fill out and submit a spreadsheet of data. Finally, we reminded contributors of the importance of ensuring their data's validity and updating when new data becomes available.

4 Data and Analysis

Throughout our project, we gathered information from the partnership members about their needs. This information came primarily from two sources, interviews and user testing sessions. We applied our findings from these sources to our work in the data collection, system design and development, as well as the methods for gathering additional information.

4.1 Interview findings and applications

Our first and primary data source in this project was the analysis of interviews we conducted during the first objective phase. We used these findings to determine partners' needs and constraints for data access, to design the data display system and to ensure the effectiveness of that system.

4.1.1 Data interest

Through our interview analyses, we learned what data the partners wanted out of the project. Health data was requested directly or mentioned with interest in eight of the sixteen interviews with representatives of potential users; its demand was greater than that for any other type of data. Demographic data was also in high demand, as the 2001 census data was still widely used among the partnership. Other types of data were mentioned only in one or two interviews, indicating that their demand was significantly lower. Examples included court data, data of all types stratified by ethnicity and up-to-date indices of individual maintenance works within the borough.

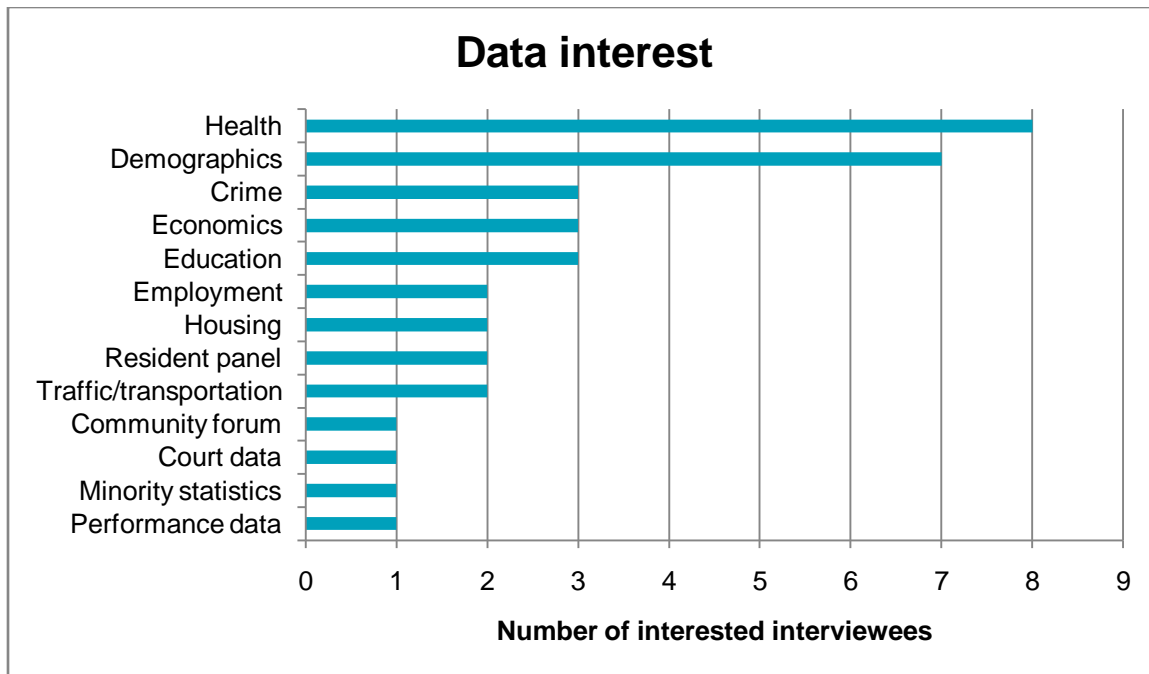


Figure 11 Data interest by number of interviewees

While groups generally possessed data within their field, many expressed a need for secondary groups of data that would help them explore correlations between trends in their field and others. For example, the representative from Children, Schools and Families branch expressed interest in comparing quality of education data to the health data he wanted from the Healthier Communities branch. He hoped to find trends in children’s health that could explain problems in school performance that he was seeking to rectify. He noted that he could provide nutrition information and consumption habits for school lunches, which would in turn help Healthier Communities by providing them with more information about children’s eating habits.

Some groups’ requests conflicted with constraints expressed by other groups, such as confidentiality or data ownership. Data capable of indicating personal information near the individual level. Others requested data that had never been collected and therefore was not held by any group within the partnership.

4.1.2 Feature interest

We also learned which formats the interviewees preferred. The overwhelming consensus pointed to graphical representations that would allow comparisons between data submitted by other partners.

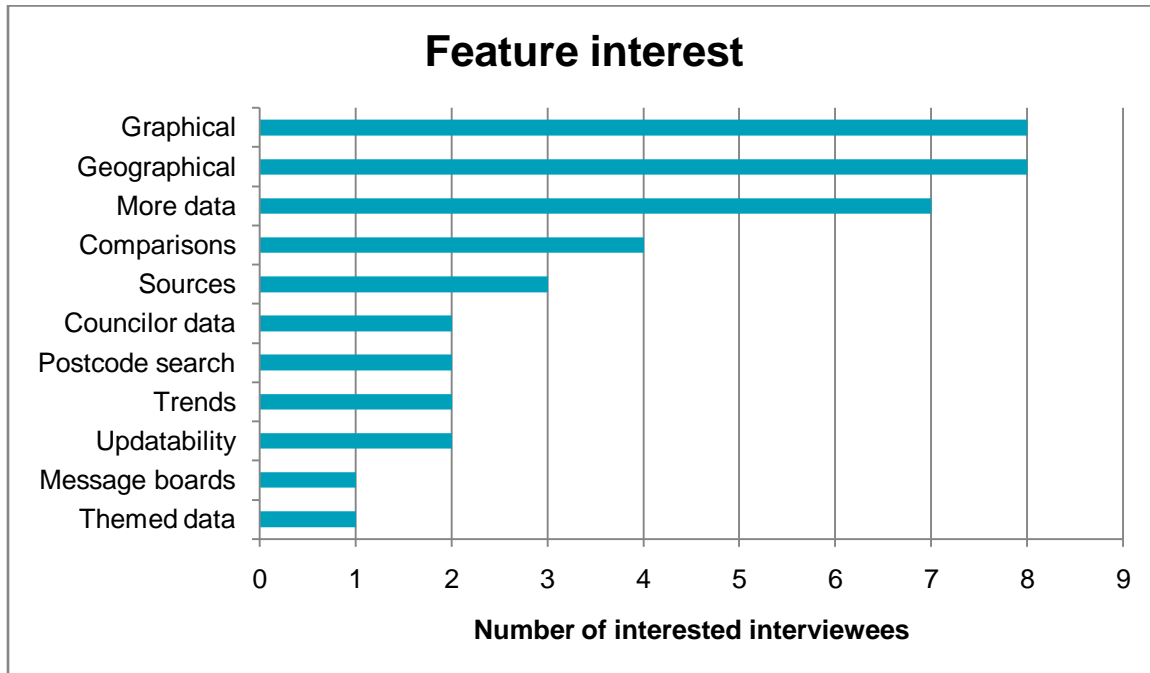


Figure 12 Feature interest by number of interviewees

Analysis of our interviews revealed users' interest in three relevant functionalities: geographical displays of data, comparisons of region-level figures between datasets, and a data contribution system simple enough to encourage widespread use and frequent updates. To comply with the users' display needs, our system needed to incorporate at least three variables into a visual, including space and two or more statistical datasets. Developing an update system allowing contributors to automatically submit data by uploading their Excel spreadsheets would satisfy the need for a simple contribution system.

4.1.2.1 System constraint applications

Due to information privacy concerns, information that exposes personal details about a specific person should not be available to the public (Information Commissioner's Office,

1998); therefore, we chose to reject regions like postal codes, whose information could be cross-referenced to identify a specific person.

We rejected borough-level data using information from our interview analyses. These data include statistics about the borough in its entirety. While this enables comparisons to other boroughs and exhibits general trends about the entire region, our interviewees expressed interest in making comparisons about regions within the borough. We also found that a team we interviewed was already working on an information display system for borough-level data.

We also rejected community forums for our data's detail level. These new areas have been roughly defined around the five main town centers in Merton. Because the community forums are so new, data collection in the areas is still limited. Interviewees also rarely mentioned these areas, so centering our system on them could decrease its appeal.

By comparing interviewee interests to data availability, we selected ward-based data from the two remaining regions. The ward and the Super Output Area (SOA) are both subdivisions of the borough; the borough of Merton comprises 20 wards and 124 SOAs (Office of National Statistics, 2001). While the level of detail provided by the smaller SOAs would allow users to make more detailed comparisons of data, fewer datasets are available. The complexity of including 124 items in each dataset would also make adding, updating and viewing data more difficult. Because interviewees expressed greater interest in ease of updates and quantity of data than in detail level, we selected wards as our level of detail. Additionally, because each ward is composed of a unique set of SOA regions, we can convert crucial SOA-level data to ward-level by totaling or averaging the values.

While collecting data indiscriminately offers the advantage of a larger dataset for the final system, it also presents several problems. As we have already decided to display data on a single level of detail, including data collected at other levels could require complex conversions, making additions to the system's database difficult, or impossible,

making the collection of such data a wasted effort. We must also avoid interfering with projects already being developed by other partnership members. Finally, we were constrained by the availability of data, as we must rely on partnership members to offer data that is not already accessible. Due to these concerns, we decided to focus on collecting data relevant to interviewees' interests, but include functionality to expand our system with additional datasets.

4.1.2.2 System design applications

Our interviews also shaped our design for the system.

While seven of the groups expressed a desire for more data to be easily available online, most of the data holders we interviewed stated that their time was so limited that they would not have time to submit data. This indicated that the system would need to accept data directly from existing sources whenever possible and from sources that could be adapted quickly and easily from existing sources otherwise. In order to accommodate this, we developed Excel templates for any data that did not already have a compatible source. Using this approach, data checking must occur prior to the Excel templates being submitted into the system.

While three of sixteen interviewees requested sources, dates and caveats about the data, six interviewees mentioned that their primary use for data would be in writing reports, a process that would also require these sorts of metadata. On the interface of our system, sources and years for each dataset appeared prominently, allowing users to cite them as references or ensure the validity of the data. Users are also able and instructed to export images generated by the system for use in reports. While our interview with the council's web information manager revealed that images of ward boundaries are copyrighted, he informed us that we could use the images if we added a council watermark to them and noted the copyright information.

Potential users of the data on our display system expressed interest in graphical displays of data with dimensions such as space, time and themes. To respond to this need, our system uses a map of the borough of Merton to display data, automatically

giving a spatial dimension. Because our system lacks multiple time periods' datasets, the dimension of time cannot be fully implemented at this point. The system could be expanded into the temporal dimension later, as our database can archive datasets from multiple periods. To fulfill the need for thematic data, we included data from multiple sources including information applicable specifically to the Community Plan themes. For example, we included data from the annual residents survey that directed special questions at the younger and older people of the borough. These data will apply directly to work in the Children and Young People and Older People themes.

4.1.3 Data possession

Most of our interviewees claimed to possess no contributable data, citing reasons ranging in legitimacy. Some legitimate concerns included violating confidentiality policies and sharing data collected by other groups. Others argued that they had insufficient time and resources to provide the requested data despite admitting to successfully complying with explicit demands from other groups.

Interviewee	Type of data	Notes
Jatinder Bhuhi	Health	Has access to but does not possess raw data; data are in high demand
Norman Urquia	Annual Residents Survey	Data are in high demand
Chris Williams	Crime/Police	
Nick Smart/Sue Tanton	Economics	
Simon Deakin	Children and education	
Gary Shaw	GIS	
Steve Lawrence	Miscellaneous	Has access to 126 databases
Abbi Scott	Miscellaneous	Already started gathering information

Figure 13 Interviewees possessing relevant data

By comparing the table of interviewees' interest in data types with the list of potential data sources, we found sources for data in our interviewees' areas of interest that met our constraints: ward-based, non-personal, quantifiable data that could be compared in a useful way to other datasets. If they held any, we asked to acquire it; since the data would usually be digitized, this process was as simple as emailing a spreadsheet to us.

While finding and providing these data in the final system would be ideal, our primary goal was to showcase a tool that these potential contributors could use to display their

information. These groups may be compelled to submit their data once they have seen a system that could support their data more easily. For this purpose, we used data from the Annual Residents Survey, a collection of local opinions on a range of topics; local crime data from the Metropolitan Police Service; synthetic health estimates from the local Primary Care Trust; and Office of National Statistics figures, such as those from the 2001 census data.

4.1.4 Potential conflicts with existing projects

Asking interviewees about their concerns with our plans revealed that other teams within the council have been working on projects in similar areas. To avoid conflicts, we interviewed the managers of two such projects. If necessary, we planned to use this information to help us constrain the domain of our own project.

Abbi Scott of the council's Policy and Partnerships team was gathering data to make borough-level statistics available to the partnership. After discussing the bounds of our respective projects, we found no conflicts as long as our data remain at different regional levels. Instead, the projects complement one another, as general trends at the borough-level can be explored in more detail at the ward-level. As such, we included links to her information in our project's documentation, inviting users to also investigate borough-level information.

Simon Morgan of the council's Performance and Business Improvement team was gathering performance data from potential contributors to our database. He developed a program for automatic submission of data to a master Excel sheet. Although there was some overlap with our update system, the difference in content ensured that there will be no conflicts; with our focus on information display, the projects can exist simultaneously without overlap. Because his project uses a spreadsheet as its ultimate storage tool, we cannot emulate his update system in our database-driven project.

4.2 Testing session findings and applications

As a secondary data source, we used feedback from the testing sessions we held during the system's development. While the interviews provided most of the information

constraining our project, we used these sessions to gather additional concerns as we introduced new features.

In several early demonstrations of our system, test users noted that some users might not be able to identify wards by their shape and position alone. To address this problem, we added ward names to the design. Because the names added an additional visual element to the already busy multivariable display, we also provided a button allowing users to toggle off the names when they are not needed.

In other tests of the system, users noted that they would like to add maps of their findings to reports, such as the ones mentioned in the initial interviews. Exporting the dynamic image is as simple as copying and pasting the object into an image editor such as Microsoft Paint, but users wanted to include the keys to explain the connection between variables and their representations for viewers of their reports. To fulfill this demand, we moved the keys from below the drop-down menus in the HTML to beside the map in the SVG image. This placed them within a map image when its SVG image is copied. We also added the references and update years for each dataset, since a report's reader might benefit from these as well.

While testing the spreadsheet templates, our testers unanimously responded that they would prefer rows to hold each ward's data and columns to hold each dataset. They needed to use Excel's sort functions on their datasets, which are available to columns but not rows. We solved this issue by transposing existing spreadsheets of data and redesigning the template.

5 Conclusions

With our system developed and accessible on the council website, partnership members have a new tool for data sharing and analysis to prepare them for decision-making. While this system offers the potential to reveal heretofore undiscovered or unexplored relationships between sets of data, partnership members must keep its database populated with up-to-date, relevant information to maintain its usefulness.

5.1 Accomplishments

At the launch of this project, the partnership's sole means of information sharing were the static web pages on its official website. These displays rarely incorporated graphics, and when comparisons were offered, no user input was requested. The result was a set of one- and two-dimensional comparisons drawn from a small set of outdated information. While multiple datasets existed on the website, it offered no comparisons between them.

To address these issues, our project has provided a system that enables users to more easily analyze and share data. The team assessed the potential users' needs and the constraints for a system's development. We discovered a demand for dynamic, multidimensional comparisons of geographic data stratified by ward. Following input from our users, we designed and developed a system capable of dynamically varying visual elements representing up to four variables, taken from an easily updatable database. This new system can inform partners' understanding of their constituencies as they explore sets of data in new ways.

5.2 Issues and recommendations

While our system presents new opportunities to users, it also presents new responsibilities. Our limited development time and technical expertise has prevented us from implementing every ideal function. We recommend continued development on the system to resolve these issues as its user base grows.

5.2.1 Expanding supply of data

A key finding in our interviews with partners was the demand for more data from other groups. While we attempted to acquire health data throughout the latter five weeks of our project, groups only offered a meager selection of outdated and overused datasets. To address the problem, we focused on the ease with which our system's database could be updated.

We recommend adding data sharing to the criteria by which a team is judged in its contributions to partnership goals. In addition to assessing their ability to meet benchmarks established by the community plan, performance monitors could assess data sharing efforts. Displaying these performance indicators could motivate teams who previously focused on other priorities.

5.2.2 Expanding the system's functionality

As user tests have already indicated the interest in our ward-based data comparison system, adding new features would further improve the benefit this system would offer. Time constraints prevented us from developing all features requested by our interviewees, and two worthy candidates for implementation remained, time-based trends and additional geographic levels. While both of these features would require further development of our code, the council information technology staff possesses the technical skills to alter the system if necessary. A second group of WPI students could also expand the system, implementing all of our recommendations.

5.2.2.1 Additional levels of geography

Though we constrained our project to ward-level for the sake of achieving the most beneficial result from our limited time, users have requested geographic datasets at other levels. While SOA-level sets, for example, could be converted to ward-level for use in our system, a potentially beneficial functionality would allow users to analyze these datasets at their native level. These comparisons could present ward-level trends in even greater detail where both levels of data exist.

5.2.2.2 Additional periods of time

As the system ages, it will accumulate multiple years' datasets, presenting the system's maintainers with the option of replacing old values or retaining them along with new ones. As the number of datasets grows, archiving multiple years of data would require increasing amounts of server space, but would be as simple as appending a year to each dataset's title. While overwriting them would allow the system to continue to provide up-to-date statistics, maintaining the historical data could prepare the database for the additional of a time variable to the system. If the system ever underwent a major redesign, functionality could be added to animate the progression of datasets and trends over time.

5.2.3 Checking validity of data

As datasets are added to the database, the system currently relies upon the contributor to ensure their validity, including the ranges for comparison. In this context, errors exist in two forms, factual and syntactical.

Syntactical errors in the spreadsheet include entries of nonnumeric values, such as slashes and quotation marks, and improperly ordered ranges. In a dataset, these errors will prevent the system from displaying them. As such, we recommend that contributors test their datasets using the display system after uploading any data.

Factual errors occur when a user inputs a value that the system can correctly process and display, but is not an accurate figure for the data item it is meant to represent. These errors offer a far greater threat to the system's usefulness. While syntactical errors will prevent access to one dataset until the error is resolved, a factual error might go undetected indefinitely, depending on its severity. If factual errors create misleading trends in data, unknowing users could base decisions upon this faulty information. This danger reinforces the importance that users

5.3 Implications

As partners' access to additional datasets grows, so will their ability to recognize subtle trends local intelligence data. These displays of multidimensional data will reveal

additional complexities in Merton's social issues, beyond the two dimensions to which their graphical displays were once limited. When used properly, our system will not provide definitive answers to their questions or proof of their theories, but rather it will allow partners to conduct initial tests of a hypothesis or seek explanations for aberrations in other datasets.

6 Works Cited

Bajekal, M., Sholes, S., Pickering, K., & Purdon, S. (2002). *Synthetic Estimates of Healthy Lifestyle Behaviours*. London: Department of Health.

Bryman, A. (2001). *Social Research Methods*. Oxford: Oxford University Press.

Chisholm, W., Vanderheiden, G., & Jacobs, I. (1999, May 5). *Web Content Accessibility Guidelines 1.0*. Retrieved May 15, 2008, from World Wide Web Consortium: <http://www.w3.org/TR/WCAG10/>

Dreifus, C. (2003, August 26). A Conversation with Sally Ride; Painful Questions From an Ex-Astronaut. *New York Times* .

Idea Knowledge. (2007). *Merton Partnership Members*. Retrieved March 27, 2008, from Idea Knowledge: <http://www.idea-knowledge.gov.uk/idk/aio/5676732>

Information Commissioner's Office. (1998). *The Data Protection Act*. Retrieved May 12, 2008, from Information Commissioner's Office: http://www.ico.gov.uk/what_we_cover/data_protection.aspx

Merton Council. (2007, April 13). *Performance Statistics*. Retrieved April 7, 2007, from Department fact sheet - Environment and Regeneration: http://www.merton.gov.uk/community/lsp/partnership-information/fact-sheets/mp-is-dept-er/performance_statistics.htm#crimesafety

Merton Partnership. (2007). *Members of the Merton Partnership*. Retrieved May 9, 2008, from Merton Partnership: <http://www.mertonpartnership.org/mp-hom/mp-organisations.htm>

Merton Partnership. (2006, June 14). *The Community Plan 2006-2015*. Retrieved May 12, 2008, from Merton Council: <http://www.merton.gov.uk/communityplan>

Neighbourhood Renewal Unit. (2005). *Local Strategic Partnerships*. Retrieved April 7, 2008, from Communities and Local Government:
<http://www.neighbourhood.gov.uk/page.asp?id=531>

Office of National Statistics. (2001). *Census Data 2001*. Retrieved April 29, 2008, from Office of National Statistics: www.statistics.gov.uk/neighbourhood/

Terry, D. (2006, September 18). *More Decisions, More Complexity, More Data: Teradata Survey Validates Global Phenomenon*. Retrieved March 29, 2008, from Teradata: <http://www.teradata.com/t/page/157049/>

Tufte, E. (2003, March 18). *PowerPoint Does Rocket Science: Assessing the Quality and Credibility of Technical Reports*. Retrieved May 29, 2008, from The Work of Edward Tufte and Graphics Press: http://www.edwardtufte.com/bboard/q-and-a-fetch-msg?msg_id=0001yB&topic_id=1&topic=Ask+E%2eT%2

Valenzuela, D., & Shrivastava, P. (2002). *Interview as a Method for Qualitative Research*. Retrieved May 7, 2008, from Arizona State University:
<http://www.public.asu.edu/~kroel/www500/Interview%20Fri.pdf>

Yorkshire and Humber Public Health Observatory. (2004, July 15). *Indices of Multiple Deprivation 2004*. Retrieved May 7, 2008, from Communities and Local Government:
<http://www.communities.gov.uk/archived/general-content/communities/indicesofdeprivation/216309/>

Appendix A Lists of Interviewees

Interviewee	Organization	Community Plan Theme	Other role
Abdool Kara	Merton Council		Assistant Chief Executive
Paul Bateman	Merton Council		Representative to councilors
Suzanne Barrows	Merton Council		Performance monitoring
Daniel Moore	Merton Council		Scrutiny
Robert Roach	Merton Council		Community forums
Ingrid Lackajis	Merton Council	Older people	
Jatinder Bhuhi	Merton Council	Healthier Communities	
Chris Williams	Merton Council	Safer and Stronger Communities	
Paul Ballatt	Merton Council	Younger People	
Sue Tanton	Merton Council	Sustainable Communities	
Patricia Anderson	Merton Unity Network	Sustainable Communities, Safer and Stronger Communities	
Simon Deakin	National Health Service	Healthier Communities	

Figure 14 Partnership decision-maker interviewees

Interviewee	Technical Specialty	Information Display Project
Gary Shaw	Geographic Information Systems	
Gavin Compton	Web Information Manager	
Steve Lawrenson	Information Technology	
Ben Harris, Abbi Scott and Joanna Richards		Borough-level data
Simon Morgan		Performance data

Figure 15 Council resource manager interviewees

Appendix B Documentation: Using ward profiling

Introduction page

Upon arrival to the web page housing the ward profiling system, an introductory screen greets the user. This screen contains a simple explanation of what the system is, what the purpose of the system is, and a brief explanation of the progression of using the system. A link to the display system is located in the bottom right corner of the introduction page.

Welcome to ward profiling

Welcome to the Merton Council's geographic information display system. This tool will allow you to view and compare up to three sets of data on a map of Merton's electoral wards. These data range from demographic Census data to opinions gathered in the Annual Residents Survey and can be easily updated and expanded as new information becomes available.

On the next page, you will find dropdown menus allowing you to select the data you wish to compare. In the first dataset, light to dark shading of the green background of each ward will signify the values' variance from low to high values. The second dataset's values will be signified by similar shading of purple dots at the center of each ward. As values in the third dataset range from low to high, the size of these dots will change from small to large. For the exact ranges that these changes represent, refer to the legend beneath the dropdown menu from which you selected a variable.

[Continue to information system](#)

Display system

The page begins with the title in the upper left corner of the page, and a link to the update system in the upper right. This link should be deleted when the display system is put on the internet so as to block the public from changing any data.

Ward profiling in Merton

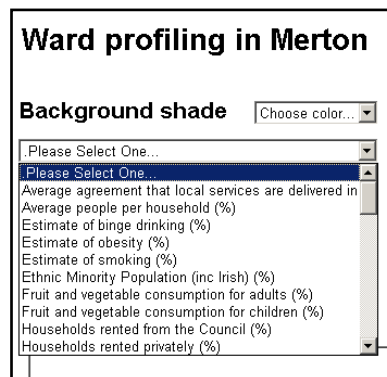
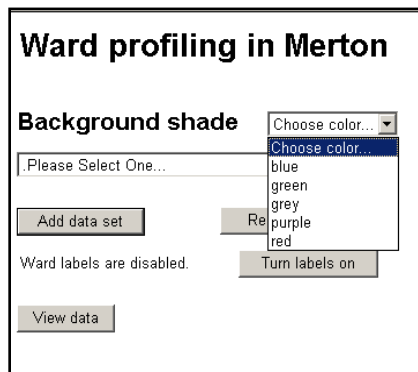
[To update system](#)

Background shade

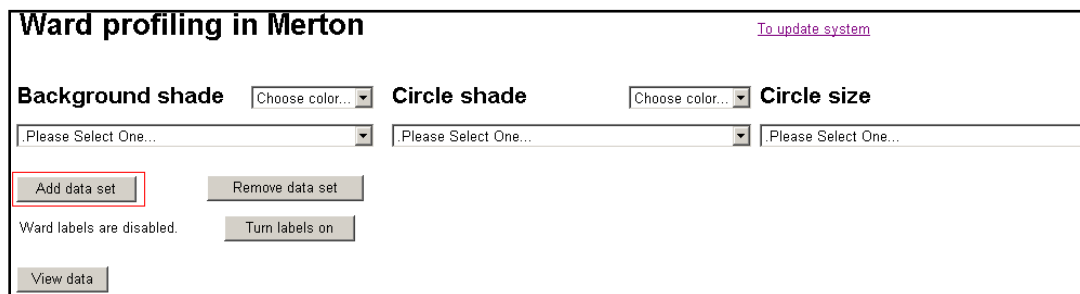
Ward labels are disabled.

Continuing down the page, there is originally a single place where the user can enter a data set and the color that the data should be represented in. This data set will be

displayed as the background shade of the wards, as denoted by the label. The drop down menu for choosing a data set lists the available options alphabetically. The user can also choose from blue, green, grey, purple, and red color scales.



The next two buttons, 'Add data set' and 'Remove data set', allow the user to specify how many variables they would like to see compared on the map and graphs. The system supports a maximum of three variables other than geography and requires a minimum of one. Each separate data set included is represented in a different way on the map. The first variable is represented by the background color, the second by the shade of a circle that appears in the middle of each ward, and the third by the size of said circles. These separate display options are always added and deleted in the same order.



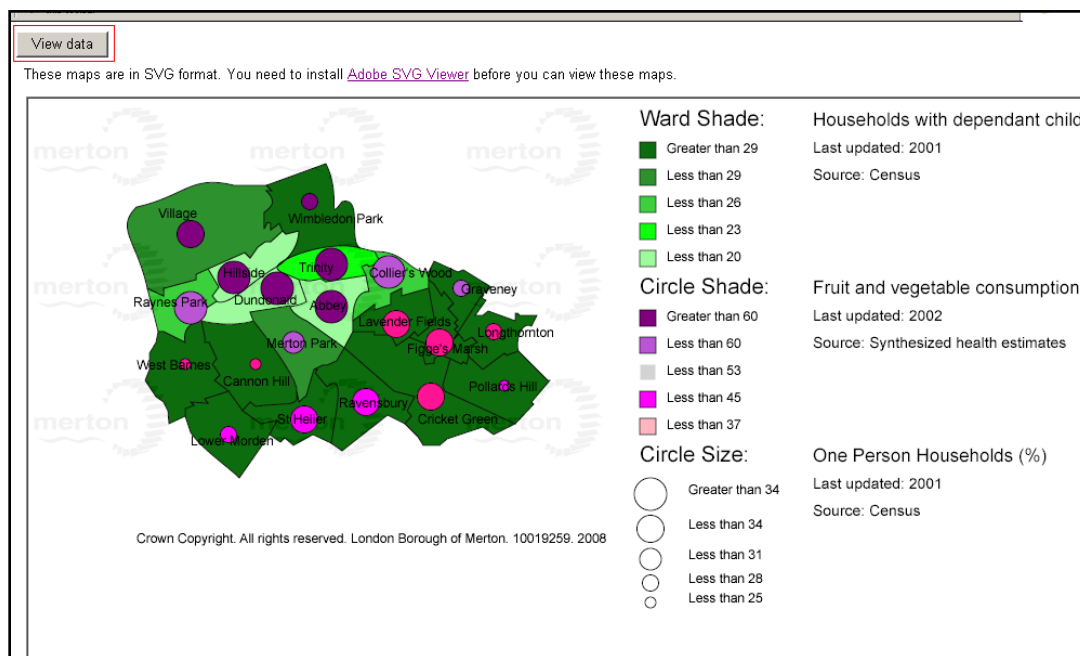
After the 'add' and 'remove' buttons is an option to turn ward labels on and off. The ward labels name each ward individually, in case the user is not familiar with all the names.

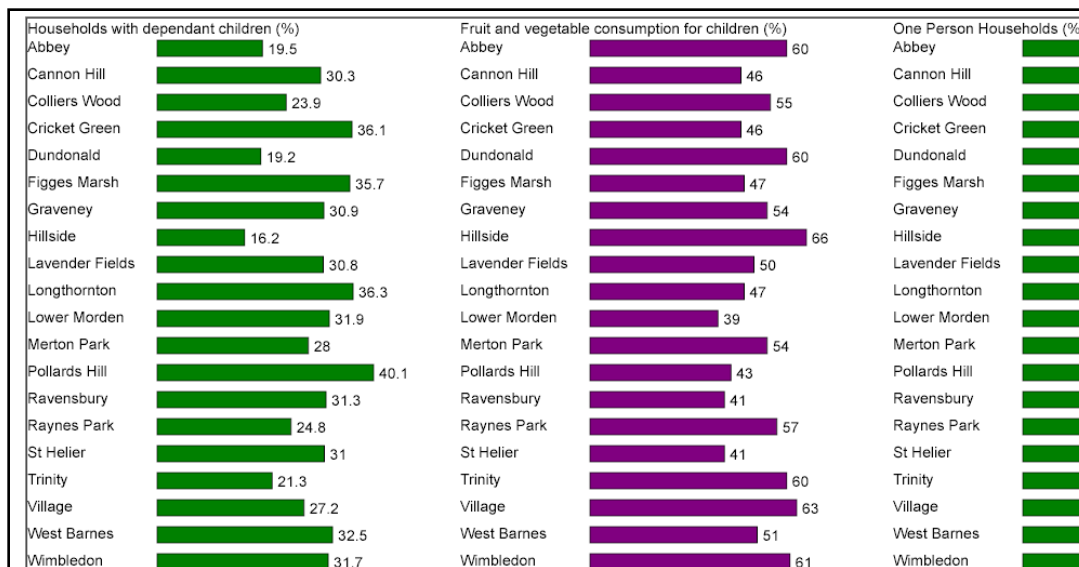
Ward profiling in Merton [To update system](#)

Background shade
Circle shade
Circle size

Ward labels are enabled.

Once the user has entered all of the information and specifications that he/she wants to compare, the 'View data' button will generate the appropriate maps, graphs and tables. The user can scroll down to see all of these displays, in that order. Right clicking on the map or graphs and choosing 'Copy SVG' copies the visual representation. It can then be pasted in a basic picture-viewing program, such as Microsoft Paint.





Information title	Households with dependant children (%)	Please Select One...	Fruit and vegetable consumption for children (%)	Please Select One...
Abbey	19.5	Abbey	60	Abbey
Cannon Hill	30.3	Cannon Hill	46	Cannon Hill
Colliers Wood	23.9	Colliers Wood	55	Colliers Wood
Cricket Green	36.1	Cricket Green	46	Cricket Green
Dundonald	19.2	Dundonald	60	Dundonald
Figges Marsh	35.7	Figges Marsh	47	Figges Marsh
Graveney	30.9	Graveney	54	Graveney
Hillside	16.2	Hillside	66	Hillside
Lavender Fields	30.8	Lavender Fields	50	Lavender Fields
Longthornton	36.3	Longthornton	47	Longthornton
Lower Morden	31.9	Lower Morden	39	Lower Morden
Merton Park	28	Merton Park	54	Merton Park
Pollards Hill	40.1	Pollards Hill	43	Pollards Hill
Ravensbury	31.3	Ravensbury	41	Ravensbury
Raynes Park	24.8	Raynes Park	57	Raynes Park
St. Helier	31	St. Helier	41	St. Helier
Trinity	21.3	Trinity	60	Trinity
Village	27.2	Village	63	Village
West Barnes	32.5	West Barnes	51	West Barnes
Wimbledon	31.7	Wimbledon	61	Wimbledon

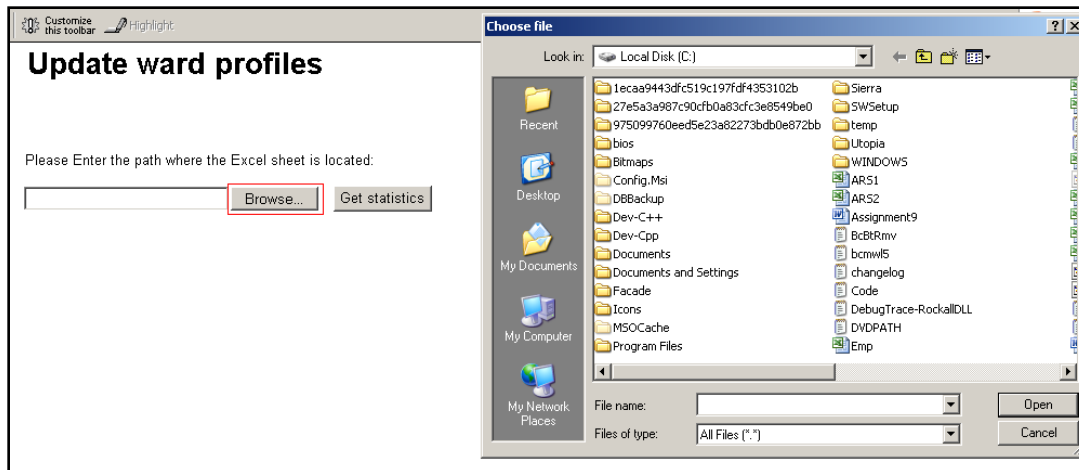
Update system

Update ward profiles

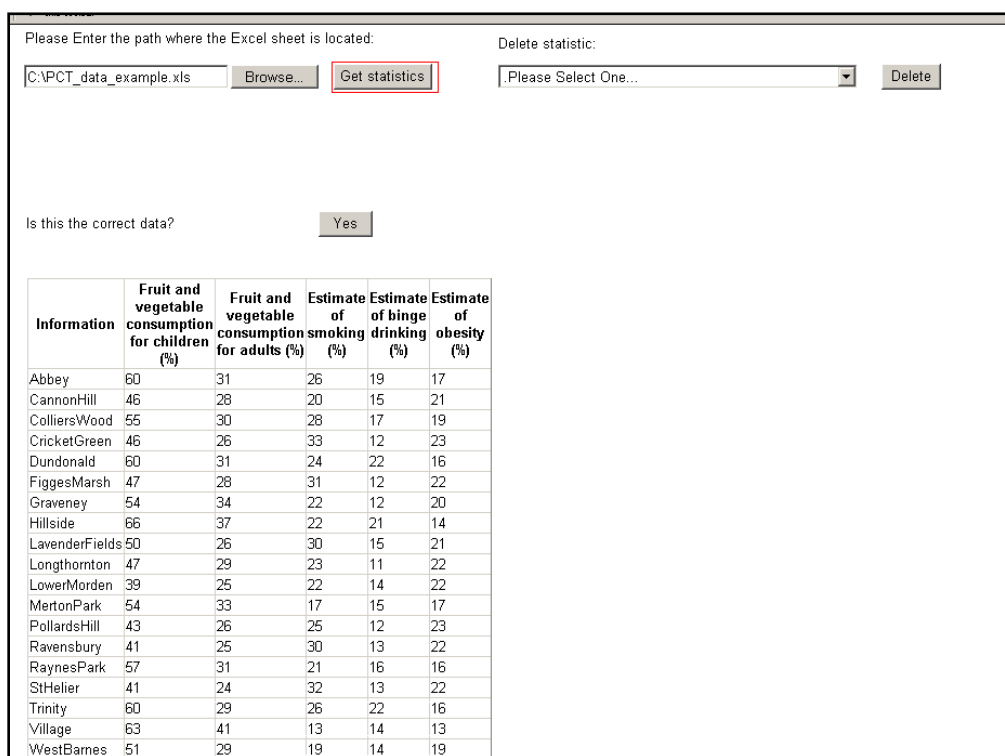
[To display system](#)

Please Enter the path where the Excel sheet is located: Delete statistic:

The update system is comprised of two distinct functionalities, adding to the database and deleting from the database. By clicking the browse button, the user can specify the location of the information to be put into the database.



Pressing the 'Get statistics' button generates a table populated by the data in the specified excel sheet, and an option to confirm that the data is correct.



The confirm ('Yes') button adds the data into the database.

Is this the correct data? Update successful.

Information	Fruit and vegetable consumption for children (%)	Fruit and vegetable consumption for adults (%)	Estimate of smoking (%)	Estimate of binge drinking (%)	Estimate of obesity (%)
Abbey	60	31	26	19	17
CannonHill	46	28	20	15	21
ColliersWood	55	30	28	17	19
CricketGreen	46	26	33	12	23
Dundonald	60	31	24	22	16

On the other side of the page, the user can specify a data set in the drop down menu to delete, and click the 'Delete' button.

[To display system](#)

sted:

Get statistics

Delete statistic:

Please Select One...

- Please Select One...
- Average agreement that local services are delivered in
- Average people per household (%)
- Estimate of binge drinking (%)
- Estimate of obesity (%)
- Estimate of smoking (%)
- Ethnic Minority Population (inc Irish) (%)
- Fruit and vegetable consumption for adults (%)
- Fruit and vegetable consumption for children (%)
- Households rented from the Council (%)
- Households rented privately (%)

[To display system](#)

Delete statistic:

Estimate of smoking (%)

Are you sure you want to delete Estimate of smoking (%)?

A confirmation is needed to make sure that the delete button was not pressed accidentally, so the 'Confirm' button needs to be pressed to actually delete the data set from the database. The 'Deny' button will reset the field and nothing will happen.

Update ward profiles

[To display system](#)

Please Enter the path where the Excel sheet is located:

Delete statistic:

Please Select One...

Are you sure you want to delete Estimate of smoking (%)?

Estimate of smoking (%) has been deleted.

Appendix C Code

Introduction.aspx

```
<%@ Page Language="C#" %>

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">

<script runat="server">

    protected void Button1_Click(object sender, EventArgs e)
    {

    }

    //When the page loads
    protected void Page_Load(object sender, EventArgs e)
    {
        //Set the label to have a hyperlink to the actual information system
        - done this way to avoid javascript
        ContinueLbl.Text = "<a href='WardProfile.aspx'>Continue to
information system</a>";
    }
</script>

<html xmlns="http://www.w3.org/1999/xhtml" >
<head runat="server">
    <meta content="text/VBScript" http-equiv="content-script-type" />
    <title>Welcome</title>
</head>
<body>
    <form id="form1" runat="server">
    <div>
    <h1>
        Welcome to ward profiling
    </h1>
        &nbsp;

    <p>Welcome to the Merton Council's geographic information display system.
This tool will allow you to view and compare up to three sets of data on a
map of Merton's electoral wards. These data range from demographic Census
data to opinions gathered in the Annual Residents Survey and can be easily
updated and expanded as new information becomes available. </p>

    <p>On the next page, you will find dropdown menus allowing you to select
the data you wish to compare. In the first dataset, light to dark shading of
the green background of each ward will signify the values' variance from low
to high values. The second dataset's values will be signified by similar
shading of purple dots at the center of each ward. As values in the third
dataset range from low to high, the size of these dots will change from small
```

to large. For the exact ranges that these changes represent, refer to the legend beneath the dropdown menu from which you selected a variable.</p>
 <asp:Label ID="ContinueLbl" runat="server" Style="z-index: 101; left: 621px; position: absolute; top: 319px" Text="Label" Width="238px"></asp:Label>
 </div>
 </form>
</body>
</html>

WardProfile.aspx

```

<%@Page Language ="C#"%>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">

<html xmlns="http://www.w3.org/1999/xhtml" >
<head>
<script runat="server">
    //NOTE: Some comments will end with an (AJ) tag. This means that this
method was taken
    //to avoid javascript, so as to conform to the Partnership's development
plan. This does not
    //mean the developer is against javascript, and it also doesn't mean that
the implementation is better.

    //Function to go through the possible selection combinations and put
together the correct
    //string of variable declarations to pass to VillageDraw.aspx (the file
that has the
    //SVG)
    protected string getDataString(string start)
    {
        //Retreive Source Data From the Database for Var1
        System.Data.SqlClient.SqlConnection src = new
System.Data.SqlClient.SqlConnection("Data
Source=.\SQLEXPRESS;AttachDbFilename=|DataDirectory|\\Wards.mdf;Integrated
Security=True;User Instance=True");
        System.Data.DataSet srclds = new System.Data.DataSet();
        string SelectCommandSrc = "SELECT * FROM Sources WHERE Statistic = '"
+ Var1.Selected.Value.ToString() + "'";
        System.Data.SqlClient.SqlDataAdapter DataCommandSrc = new
System.Data.SqlClient.SqlDataAdapter(SelectCommandSrc, src);
        DataCommandSrc.Fill(srclds);

        //Pass the wards' statistics and the limits, in that order, in
alphabetical order
        //The Var1...Var3 drop down boxes are lined up with the rows of
the grid box (but not the original database).
        //This could be changed to directly access
        string dataField = start;
        dataField += ("Abbey=" +
GridView1.Rows[Var1.SelectedIndex].Cells[1].Text.ToString())

```

```

        + "&CannonHill=" +
GridView1.Rows[Var1.SelectedIndex].Cells[2].Text.ToString()
        + "&ColliersWood=" +
GridView1.Rows[Var1.SelectedIndex].Cells[3].Text.ToString()
        + "&CricketGreen=" +
GridView1.Rows[Var1.SelectedIndex].Cells[4].Text.ToString()
        + "&Dundonald=" +
GridView1.Rows[Var1.SelectedIndex].Cells[5].Text.ToString()
        + "&FiggesMarsh=" +
GridView1.Rows[Var1.SelectedIndex].Cells[6].Text.ToString()
        + "&Graveney=" +
GridView1.Rows[Var1.SelectedIndex].Cells[7].Text.ToString()
        + "&Hillside=" +
GridView1.Rows[Var1.SelectedIndex].Cells[8].Text.ToString()
        + "&LavenderFields=" +
GridView1.Rows[Var1.SelectedIndex].Cells[9].Text.ToString()
        + "&Longthornton=" +
GridView1.Rows[Var1.SelectedIndex].Cells[10].Text.ToString()
        + "&LowerMorden=" +
GridView1.Rows[Var1.SelectedIndex].Cells[11].Text.ToString()
        + "&MertonPark=" +
GridView1.Rows[Var1.SelectedIndex].Cells[12].Text.ToString()
        + "&PollardsHill=" +
GridView1.Rows[Var1.SelectedIndex].Cells[13].Text.ToString()
        + "&Ravensbury=" +
GridView1.Rows[Var1.SelectedIndex].Cells[14].Text.ToString()
        + "&RaynesPark=" +
GridView1.Rows[Var1.SelectedIndex].Cells[15].Text.ToString()
        + "&StHelier=" +
GridView1.Rows[Var1.SelectedIndex].Cells[16].Text.ToString()
        + "&Trinity=" +
GridView1.Rows[Var1.SelectedIndex].Cells[17].Text.ToString()
        + "&Village=" +
GridView1.Rows[Var1.SelectedIndex].Cells[18].Text.ToString()
        + "&WestBarnes=" +
GridView1.Rows[Var1.SelectedIndex].Cells[19].Text.ToString()
        + "&Wimbledon=" +
GridView1.Rows[Var1.SelectedIndex].Cells[20].Text.ToString()
        + "&first=" + ((float.Parse(GridView2.Rows[Var1.SelectedIndex -
1].Cells[1].Text.ToString())) + 0.01).ToString()
        + "&second=" + GridView2.Rows[Var1.SelectedIndex -
1].Cells[2].Text.ToString()
        + "&third=" + GridView2.Rows[Var1.SelectedIndex -
1].Cells[3].Text.ToString()
        + "&fourth=" + GridView2.Rows[Var1.SelectedIndex -
1].Cells[4].Text.ToString()
        + "&Var1Select=" + Var1.SelectedValue.ToString());

float max =
float.Parse(GridView1.Rows[Var1.SelectedIndex].Cells[1].Text.ToString());

for (int i = 2; i <= 20; i++)
{

```

```

        if
        (float.Parse(GridView1.Rows[Var1.SelectedIndex].Cells[i].Text.ToString()) >
max)
        {
            max =
float.Parse(GridView1.Rows[Var1.SelectedIndex].Cells[i].Text.ToString());
        }
    }

    dataField += "&max=" + max.ToString();

    //Try to get the source information that corresponds to the selected
data set
    try
    {
        dataField += "&Var1LastUpdate=" +
srclds.Tables[0].Rows[0]["LastUpdate"].ToString()
        + "&Var1Source=" + srclds.Tables[0].Rows[0]["Source"];
    }
    //If there is none specified, pass 'Unknown'
    catch (IndexOutOfRangeException)
    {
        dataField += "&Var1LastUpdate=Unknown" +
"&Var1Source=Unknown";
    }
    //Move on
    finally
    {
        //Retrieve Database source data for the second variable
        System.Data.SqlClient.SqlConnection src2 = new
System.Data.SqlClient.SqlConnection("Data
Source=.\SQLEXPRESS;AttachDbFilename=|DataDirectory|\\Wards.mdf;Integrated
Security=True;User Instance=True");
        System.Data.DataSet src2ds = new System.Data.DataSet();
        string SelectCommandSrc2 = "SELECT * FROM Sources WHERE
Statistic = '" + Var2.SelectedValue.ToString() + "'";
        System.Data.SqlClient.SqlDataAdapter DataCommandSrc2 = new
System.Data.SqlClient.SqlDataAdapter(SelectCommandSrc2, src2);
        DataCommandSrc2.Fill(src2ds);

        if (Var2.Visible && Var2.SelectedIndex != 0)
        {
            //append string appropriately
            dataField += ("&problem=0"
+ "&AbbeyCircle=" +
GridView1.Rows[Var2.SelectedIndex].Cells[1].Text.ToString()
+ "&CannonHillCircle=" +
GridView1.Rows[Var2.SelectedIndex].Cells[2].Text.ToString()
+ "&ColliersWoodCircle=" +
GridView1.Rows[Var2.SelectedIndex].Cells[3].Text.ToString())

```

```

        + "&CricketGreenCircle=" +
GridView1.Rows[Var2.SelectedIndex].Cells[4].Text.ToString()
        + "&DundonaldCircle=" +
GridView1.Rows[Var2.SelectedIndex].Cells[5].Text.ToString()
        + "&FiggesMarshCircle=" +
GridView1.Rows[Var2.SelectedIndex].Cells[6].Text.ToString()
        + "&GraveneyCircle=" +
GridView1.Rows[Var2.SelectedIndex].Cells[7].Text.ToString()
        + "&HillsideCircle=" +
GridView1.Rows[Var2.SelectedIndex].Cells[8].Text.ToString()
        + "&LavenderFieldsCircle=" +
GridView1.Rows[Var2.SelectedIndex].Cells[9].Text.ToString()
        + "&LongthorntonCircle=" +
GridView1.Rows[Var2.SelectedIndex].Cells[10].Text.ToString()
        + "&LowerMordenCircle=" +
GridView1.Rows[Var2.SelectedIndex].Cells[11].Text.ToString()
        + "&MertonParkCircle=" +
GridView1.Rows[Var2.SelectedIndex].Cells[12].Text.ToString()
        + "&PollardsHillCircle=" +
GridView1.Rows[Var2.SelectedIndex].Cells[13].Text.ToString()
        + "&RavensburyCircle=" +
GridView1.Rows[Var2.SelectedIndex].Cells[14].Text.ToString()
        + "&RaynesParkCircle=" +
GridView1.Rows[Var2.SelectedIndex].Cells[15].Text.ToString()
        + "&StHelierCircle=" +
GridView1.Rows[Var2.SelectedIndex].Cells[16].Text.ToString()
        + "&TrinityCircle=" +
GridView1.Rows[Var2.SelectedIndex].Cells[17].Text.ToString()
        + "&VillageCircle=" +
GridView1.Rows[Var2.SelectedIndex].Cells[18].Text.ToString()
        + "&WestBarnesCircle=" +
GridView1.Rows[Var2.SelectedIndex].Cells[19].Text.ToString()
        + "&WimbledonCircle=" +
GridView1.Rows[Var2.SelectedIndex].Cells[20].Text.ToString()
        + "&circle1=" + GridView2.Rows[Var2.SelectedIndex -
1].Cells[1].Text.ToString()
        + "&circle2=" + GridView2.Rows[Var2.SelectedIndex -
1].Cells[2].Text.ToString()
        + "&circle3=" + GridView2.Rows[Var2.SelectedIndex -
1].Cells[3].Text.ToString()
        + "&circle4=" + GridView2.Rows[Var2.SelectedIndex -
1].Cells[4].Text.ToString()
        + "&Var2Select=" + Var2.SelectedValue.ToString());

        float max2 =
float.Parse(GridView1.Rows[Var2.SelectedIndex].Cells[1].Text.ToString());

        for (int i = 2; i <= 20; i++)
        {
            if
(float.Parse(GridView1.Rows[Var2.SelectedIndex].Cells[i].Text.ToString()) >
max2)
            {

```

```

        max2 =
float.Parse(GridView1.Rows[Var2.SelectedIndex].Cells[i].Text.ToString());
    }
}

    dataField += "&max2=" + max2.ToString();
}

//Try to get source data again
try
{
    dataField += "&Var2LastUpdate=" +
src2ds.Tables[0].Rows[0]["LastUpdate"].ToString()
    + "&Var2Source=" + src2ds.Tables[0].Rows[0]["Source"];
}
//If not there - pass unknown
catch (IndexOutOfRangeException)
{
    dataField += "&Var2LastUpdate=Unknown" +
"&Var2Source=Unknown";
}
finally
{
    if (!(Var2.Visible && Var2.SelectedIndex != 0))
    {
        //Let Update.aspx know that no other data is coming
        dataField += "&problem=1&max2=0";
    }

    //Grab source data for third variable
    System.Data.SqlClient.SqlConnection src3 = new
System.Data.SqlClient.SqlConnection("Data
Source=.\SQLEXPRESS;AttachDbFilename=|DataDirectory|\\Wards.mdf;Integrated
Security=True;User Instance=True");
    System.Data.DataSet src3ds = new System.Data.DataSet();
    string SelectCommandSrc3 = "SELECT * FROM Sources WHERE
Statistic = '" + Var3.SelectedValue.ToString() + "'";
    System.Data.SqlClient.SqlDataAdapter DataCommandSrc3 =
new System.Data.SqlClient.SqlDataAdapter(SelectCommandSrc3, src3);
    DataCommandSrc3.Fill(src3ds);

    if ((Var3.Visible) && Var3.SelectedIndex != 0)
    {

        //Add third variable statistics to the end of the
string
        dataField += ("&radBad=0"
+ "&AbbeyRad=" +
GridView1.Rows[Var3.SelectedIndex].Cells[1].Text.ToString()
+ "&CannonHillRad=" +
GridView1.Rows[Var3.SelectedIndex].Cells[2].Text.ToString()
+ "&ColliersWoodRad=" +
GridView1.Rows[Var3.SelectedIndex].Cells[3].Text.ToString()

```

```

        + "&CricketGreenRad=" +
GridView1.Rows[Var3.SelectedIndex].Cells[4].Text.ToString()
        + "&DundonaldRad=" +
GridView1.Rows[Var3.SelectedIndex].Cells[5].Text.ToString()
        + "&FiggesMarshRad=" +
GridView1.Rows[Var3.SelectedIndex].Cells[6].Text.ToString()
        + "&GraveneyRad=" +
GridView1.Rows[Var3.SelectedIndex].Cells[7].Text.ToString()
        + "&HillsideRad=" +
GridView1.Rows[Var3.SelectedIndex].Cells[8].Text.ToString()
        + "&LavenderFieldsRad=" +
GridView1.Rows[Var3.SelectedIndex].Cells[9].Text.ToString()
        + "&LongthorntonRad=" +
GridView1.Rows[Var3.SelectedIndex].Cells[10].Text.ToString()
        + "&LowerMordenRad=" +
GridView1.Rows[Var3.SelectedIndex].Cells[11].Text.ToString()
        + "&MertonParkRad=" +
GridView1.Rows[Var3.SelectedIndex].Cells[12].Text.ToString()
        + "&PollardsHillRad=" +
GridView1.Rows[Var3.SelectedIndex].Cells[13].Text.ToString()
        + "&RavensburyRad=" +
GridView1.Rows[Var3.SelectedIndex].Cells[14].Text.ToString()
        + "&RaynesParkRad=" +
GridView1.Rows[Var3.SelectedIndex].Cells[15].Text.ToString()
        + "&StHelierRad=" +
GridView1.Rows[Var3.SelectedIndex].Cells[16].Text.ToString()
        + "&TrinityRad=" +
GridView1.Rows[Var3.SelectedIndex].Cells[17].Text.ToString()
        + "&VillageRad=" +
GridView1.Rows[Var3.SelectedIndex].Cells[18].Text.ToString()
        + "&WestBarnesRad=" +
GridView1.Rows[Var3.SelectedIndex].Cells[19].Text.ToString()
        + "&WimbledonRad=" +
GridView1.Rows[Var3.SelectedIndex].Cells[20].Text.ToString()
        + "&firstRad=" + GridView2.Rows[Var3.SelectedIndex -
1].Cells[1].Text.ToString()
        + "&secondRad=" + GridView2.Rows[Var3.SelectedIndex -
1].Cells[2].Text.ToString()
        + "&thirdRad=" + GridView2.Rows[Var3.SelectedIndex -
1].Cells[3].Text.ToString()
        + "&fourthRad=" + GridView2.Rows[Var3.SelectedIndex -
1].Cells[4].Text.ToString()
        + "&Var3Select=" + Var3.SelectedValue.ToString());

        float max3 =
float.Parse(GridView1.Rows[Var3.SelectedIndex].Cells[1].Text.ToString());

        for (int i = 2; i <= 20; i++)
        {
            if
(float.Parse(GridView1.Rows[Var3.SelectedIndex].Cells[i].Text.ToString()) >
max3)
            {

```

```

                max3 =
float.Parse(GridView1.Rows[Var3.SelectedIndex].Cells[i].Text.ToString());
            }
        }

        dataField += "&max3=" + max3.ToString();

    }

    //Same deal with the retrieving of source information
    try
    {
        dataField += "&Var3LastUpdate=" +
src3ds.Tables[0].Rows[0]["LastUpdate"].ToString()
        + "&Var3Source=" + src3ds.Tables[0].Rows[0]["Source"];
    }
    catch (IndexOutOfRangeException)
    {
        dataField += "&Var3LastUpdate=Unknown" +
"&Var3Source=Unknown";
    }
    finally
    {
        if (!(Var3.Visible) && Var3.SelectedIndex != 0)
        {
            //Let Update.aspx know nothing's coming
            dataField += "&radBad=1&max3=0";
        }

        //Specify whether or not to draw ward names
        if (ToggleOff.Visible == true)
        {
            dataField += "&wards=1";
        }
        else
        {
            dataField += "&wards=0";
        }
    }
}

}

//Set color for first variable
if (Color1.SelectedIndex == 0)
{
    dataField += "&V1Color=green";
}
else
{
    dataField += "&V1Color=" + Color1.SelectedValue.ToString();
}

if (Color2.SelectedIndex == 0)

```

```

        {
            dataField += "&V2Color=purple";
        }
        else
        {
            dataField += "&V2Color=" + Color2.Selected.Value.ToString();
        }

        //Find and pass the max data for each set to be used in the GraphDraw
file.

        //return final string, whatever that may be.
        return dataField;

    }

    //If they click the add variable button
protected void buttonAdd_Click(object sender, EventArgs e)
{
    //If the second one is already out, make the third variable visible
    if (Var2.Visible)
    {
        Var3.Visible = true;
        title3.Visible = true;
    }
    //If not, make the second one visible.
    else
    {
        Var2.Visible = true;
        title2.Visible = true;
        Color2.Visible = true;
    }

    //Don't forget to manage toggles
    if (Label2.Visible)
    {
        if (Label2.Text == "Ward labels are disabled.")
        {
            ToggleOn.Visible = true;
        }
        else
        {
            ToggleOff.Visible = true;
        }
    }
}

protected void Var2_SelectedIndexChanged(object sender, EventArgs e)
{
}

protected void Var3_SelectedIndexChanged(object sender, EventArgs e)

```

```

{
}

protected void Var4_SelectedIndexChanged(object sender, EventArgs e)
{
}

protected void Var5_SelectedIndexChanged(object sender, EventArgs e)
{
}

protected void SqlDataSource1_Selecting(object sender,
SqlDataSourceSelectingEventArgs e)
{
}

protected void GridView1_SelectedIndexChanged(object sender, EventArgs e)
{
}

protected void Var1_SelectedIndexChanged(object sender, EventArgs e)
{
}

//Upon loading the page...
protected void Page_Load(object sender, EventArgs e)
{
    //Set the link to the Update system
    SwitchLbl.Text = "<a href='Update.aspx'>To update system </a>";
}

protected void ImageButton1_Click(object sender, ImageClickEventArgs e)
{
}

//Remove variable button
protected void Button2_Click(object sender, EventArgs e)
{
    //Take care of toggles
    if (Label2.Visible)
    {
        if (Label2.Text == "Ward labels are disabled.")
        {
            ToggleOn.Visible = true;
        }
    }
    else
    {

```

```

        ToggleOff.Visible = true;
    }
}

//if the third variable is out there, get rid of it
if (Var3.Visible)
{
    Var3.Visible = false;
    title3.Visible = false;
    Var3.SelectedIndex = 0;
}
//If not, lose the second one
else if (Var2.Visible)
{
    Var2.Visible = false;
    title2.Visible = false;
    Color2.Visible = false;
    Var2.SelectedIndex = 0;
}
//Otherwise, there's only one left, so there's nothing to do.
}

protected void DetailsView2_PageIndexChanging(object sender,
DetailsViewPageEventArgs e)
{
}

protected void Button4_Click(object sender, EventArgs e)
{
}

protected void ToggleOn_Click(object sender, EventArgs e)
{
    //Switch buttons so the user can't toggle to the same thing
    ToggleOn.Visible = false;
    ToggleOff.Visible = true;
    //Remind the user what status is currently being upheld.
    Label2.Text = "Ward labels are enabled.";
}

protected void ToggleOff_Click(object sender, EventArgs e)
{
    //Switch buttons so the user can't toggle to the same thing
    ToggleOn.Visible = true;
    ToggleOff.Visible = false;
    //Remind the user what status is currently being upheld.
    Label2.Text = "Ward labels are disabled.";
}

protected void Variable1View_PageIndexChanging(object sender,
DetailsViewPageEventArgs e)
{
}

```

```

}

//View Tables Button - big time (AJ)
protected void Button3_Click2(object sender, EventArgs e)
{
    if (Var1.SelectedIndex == 0 || (Var2.Visible && Var2.SelectedIndex ==
0) || (Var3.Visible && Var3.SelectedIndex == 0))
    {
        WarningLbl.Visible = true;
    }
    else
    {
        WarningLbl.Visible = false;

        WardFrame.Attributes["src"] = getDataString("VillageDraw.aspx?");
        Graphs.Attributes["src"] = getDataString("GraphDraw.aspx?");

        //Give the user the option to now view the map (AJ)
        //This was originally done with embed and iframe, but object
        //needed to be used. Object didn't work for this at all...
        DwnLdLbl.Visible = true;
        //Label1.Visible = true;
        Label2.Visible = true;
        if (Label2.Text == "Ward labels are disabled.")
        {
            ToggleOn.Visible = true;
        }
        else
        {
            ToggleOff.Visible = true;
        }

        //Set link to updated selected variables
        //Label1.Text = "<a href = '" + getDataString() + "'> View map
</a>";

        //Create Tables
        //Open database connection
        System.Data.SqlClient.SqlConnection con = new
System.Data.SqlClient.SqlConnection("Data
Source=.\SQLEXPRESS;AttachDbFilename=|DataDirectory|\\Wards.mdf;Integrated
Security=True;User Instance=True");
        System.Data.DataSet ds = new System.Data.DataSet();
        string SelectCommand = "SELECT * FROM WardInformation WHERE
InformationTitle = '" + Var1.SelectedValue.ToString() + "'";
        System.Data.SqlClient.SqlDataAdapter DataCommand = new
System.Data.SqlClient.SqlDataAdapter(SelectCommand, con);
        DataCommand.Fill(ds);

        //Set each cell individually (AJ times 1000000)
        System.Web.UI.HtmlControls.HtmlContainerControl thead =
(System.Web.UI.HtmlControls.HtmlContainerControl)FormView1.FindControl("hd");
        thead.InnerText = "Information title";

```

```

        System.Web.UI.HtmlControls.HtmlContainerControl thead2 =
(System.Web.UI.HtmlControls.HtmlContainerControl) FormView1.FindControl ("hd2")
;
        thead2.InnerText =
(string) ds.Tables[0].Rows[0] ["InformationTitle"];

        System.Web.UI.HtmlControls.HtmlContainerControl AbbeyLbl =
(System.Web.UI.HtmlControls.HtmlContainerControl) FormView1.FindControl ("Abbey
Lbl");
        AbbeyLbl.InnerText = "Abbey";

        System.Web.UI.HtmlControls.HtmlContainerControl AbbeyStat =
(System.Web.UI.HtmlControls.HtmlContainerControl) FormView1.FindControl ("Abbey
");
        AbbeyStat.InnerText = ds.Tables[0].Rows[0] ["Abbey"].ToString();

        System.Web.UI.HtmlControls.HtmlContainerControl CannonHillLbl =
(System.Web.UI.HtmlControls.HtmlContainerControl) FormView1.FindControl ("CHLbl
");
        CannonHillLbl.InnerText = "Cannon Hill";

        System.Web.UI.HtmlControls.HtmlContainerControl CannonHillStat =
(System.Web.UI.HtmlControls.HtmlContainerControl) FormView1.FindControl ("CH");
        CannonHillStat.InnerText =
ds.Tables[0].Rows[0] ["CannonHill"].ToString();

        System.Web.UI.HtmlControls.HtmlContainerControl ColliersWoodLbl =
(System.Web.UI.HtmlControls.HtmlContainerControl) FormView1.FindControl ("CWLbl
");
        ColliersWoodLbl.InnerText = "Colliers Wood";

        System.Web.UI.HtmlControls.HtmlContainerControl ColliersWoodStat
=
(System.Web.UI.HtmlControls.HtmlContainerControl) FormView1.FindControl ("CW");
        ColliersWoodStat.InnerText =
ds.Tables[0].Rows[0] ["ColliersWood"].ToString();

        System.Web.UI.HtmlControls.HtmlContainerControl CricketGreenLbl =
(System.Web.UI.HtmlControls.HtmlContainerControl) FormView1.FindControl ("CrGLb
l");
        CricketGreenLbl.InnerText = "Cricket Green";

        System.Web.UI.HtmlControls.HtmlContainerControl CricketGreenStat
=
(System.Web.UI.HtmlControls.HtmlContainerControl) FormView1.FindControl ("CG");
        CricketGreenStat.InnerText =
ds.Tables[0].Rows[0] ["CricketGreen"].ToString();

        System.Web.UI.HtmlControls.HtmlContainerControl DundonaldLbl =
(System.Web.UI.HtmlControls.HtmlContainerControl) FormView1.FindControl ("DDLbl
");
        DundonaldLbl.InnerText = "Dundonald";

```

```

        System.Web.UI.HtmlControls.HtmlContainerControl DundonaldStat =
(System.Web.UI.HtmlControls.HtmlContainerControl)FormView1.FindControl("DD");
        DundonaldStat.InnerText =
ds.Tables[0].Rows[0]["Dundonald"].ToString();

        System.Web.UI.HtmlControls.HtmlContainerControl FiggesMarshLbl =
(System.Web.UI.HtmlControls.HtmlContainerControl)FormView1.FindControl("FMLbl
");
        FiggesMarshLbl.InnerText = "Figges Marsh";

        System.Web.UI.HtmlControls.HtmlContainerControl FiggesMarshStat =
(System.Web.UI.HtmlControls.HtmlContainerControl)FormView1.FindControl("FM");
        FiggesMarshStat.InnerText =
ds.Tables[0].Rows[0]["FiggesMarsh"].ToString();

        System.Web.UI.HtmlControls.HtmlContainerControl GraveneyLbl =
(System.Web.UI.HtmlControls.HtmlContainerControl)FormView1.FindControl("GLbl"
);
        GraveneyLbl.InnerText = "Graveney";

        System.Web.UI.HtmlControls.HtmlContainerControl GraveneyStat =
(System.Web.UI.HtmlControls.HtmlContainerControl)FormView1.FindControl("G");
        GraveneyStat.InnerText =
ds.Tables[0].Rows[0]["Graveney"].ToString();

        System.Web.UI.HtmlControls.HtmlContainerControl HillsideLbl =
(System.Web.UI.HtmlControls.HtmlContainerControl)FormView1.FindControl("HLbl"
);
        HillsideLbl.InnerText = "Hillside";

        System.Web.UI.HtmlControls.HtmlContainerControl HillsideStat =
(System.Web.UI.HtmlControls.HtmlContainerControl)FormView1.FindControl("H");
        HillsideStat.InnerText =
ds.Tables[0].Rows[0]["Hillside"].ToString();

        System.Web.UI.HtmlControls.HtmlContainerControl LavenderFieldsLbl
=
(System.Web.UI.HtmlControls.HtmlContainerControl)FormView1.FindControl("LFLbl
");
        LavenderFieldsLbl.InnerText = "Lavender Fields";

        System.Web.UI.HtmlControls.HtmlContainerControl
LavenderFieldsStat =
(System.Web.UI.HtmlControls.HtmlContainerControl)FormView1.FindControl("LF");
        LavenderFieldsStat.InnerText =
ds.Tables[0].Rows[0]["LavenderFields"].ToString();

        System.Web.UI.HtmlControls.HtmlContainerControl LongthorntonLbl =
(System.Web.UI.HtmlControls.HtmlContainerControl)FormView1.FindControl("LTLbl
");
        LongthorntonLbl.InnerText = "Longthornton";

```

```

        System.Web.UI.HtmlControls.HtmlContainerControl LongthorntonStat
=
(System.Web.UI.HtmlControls.HtmlContainerControl) FormView1.FindControl("LT");
        LongthorntonStat.InnerText =
ds.Tables[0].Rows[0]["Longthornton"].ToString();

        System.Web.UI.HtmlControls.HtmlContainerControl LowerMordenLbl =
(System.Web.UI.HtmlControls.HtmlContainerControl) FormView1.FindControl("LMLbl
");
        LowerMordenLbl.InnerText = "Lower Morden";

        System.Web.UI.HtmlControls.HtmlContainerControl LowerMordenStat =
(System.Web.UI.HtmlControls.HtmlContainerControl) FormView1.FindControl("LM");
        LowerMordenStat.InnerText =
ds.Tables[0].Rows[0]["LowerMorden"].ToString();

        System.Web.UI.HtmlControls.HtmlContainerControl MertonParkLbl =
(System.Web.UI.HtmlControls.HtmlContainerControl) FormView1.FindControl("MPLbl
");
        MertonParkLbl.InnerText = "Merton Park";

        System.Web.UI.HtmlControls.HtmlContainerControl MertonParkStat =
(System.Web.UI.HtmlControls.HtmlContainerControl) FormView1.FindControl("MP");
        MertonParkStat.InnerText =
ds.Tables[0].Rows[0]["MertonPark"].ToString();

        System.Web.UI.HtmlControls.HtmlContainerControl PollardsHillLbl =
(System.Web.UI.HtmlControls.HtmlContainerControl) FormView1.FindControl("PHLbl
");
        PollardsHillLbl.InnerText = "Pollards Hill";

        System.Web.UI.HtmlControls.HtmlContainerControl PollardsHillStat
=
(System.Web.UI.HtmlControls.HtmlContainerControl) FormView1.FindControl("PH");
        PollardsHillStat.InnerText =
ds.Tables[0].Rows[0]["PollardsHill"].ToString();

        System.Web.UI.HtmlControls.HtmlContainerControl RavensburyLbl =
(System.Web.UI.HtmlControls.HtmlContainerControl) FormView1.FindControl("RBLbl
");
        RavensburyLbl.InnerText = "Ravensbury";

        System.Web.UI.HtmlControls.HtmlContainerControl RavensburyStat =
(System.Web.UI.HtmlControls.HtmlContainerControl) FormView1.FindControl("RB");
        RavensburyStat.InnerText =
ds.Tables[0].Rows[0]["Ravensbury"].ToString();

        System.Web.UI.HtmlControls.HtmlContainerControl RaynesParkLbl =
(System.Web.UI.HtmlControls.HtmlContainerControl) FormView1.FindControl("RPLbl
");
        RaynesParkLbl.InnerText = "Raynes Park";

        System.Web.UI.HtmlControls.HtmlContainerControl RaynesParkStat =
(System.Web.UI.HtmlControls.HtmlContainerControl) FormView1.FindControl("RP");

```

```

RaynesParkStat.InnerText =
ds.Tables[0].Rows[0]["RaynesPark"].ToString();

System.Web.UI.HtmlControls.HtmlContainerControl StHelierLbl =
(System.Web.UI.HtmlControls.HtmlContainerControl)FormView1.FindControl("SHLbl");
StHelierLbl.InnerText = "St. Helier";

System.Web.UI.HtmlControls.HtmlContainerControl StHelierStat =
(System.Web.UI.HtmlControls.HtmlContainerControl)FormView1.FindControl("SH");
StHelierStat.InnerText =
ds.Tables[0].Rows[0]["StHelier"].ToString();

System.Web.UI.HtmlControls.HtmlContainerControl TrinityLbl =
(System.Web.UI.HtmlControls.HtmlContainerControl)FormView1.FindControl("TLLbl");
TrinityLbl.InnerText = "Trinity";

System.Web.UI.HtmlControls.HtmlContainerControl TrinityStat =
(System.Web.UI.HtmlControls.HtmlContainerControl)FormView1.FindControl("T");
TrinityStat.InnerText =
ds.Tables[0].Rows[0]["Trinity"].ToString();

System.Web.UI.HtmlControls.HtmlContainerControl VillageLbl =
(System.Web.UI.HtmlControls.HtmlContainerControl)FormView1.FindControl("VLLbl");
VillageLbl.InnerText = "Village";

System.Web.UI.HtmlControls.HtmlContainerControl VillageStat =
(System.Web.UI.HtmlControls.HtmlContainerControl)FormView1.FindControl("V");
VillageStat.InnerText =
ds.Tables[0].Rows[0]["Village"].ToString();

System.Web.UI.HtmlControls.HtmlContainerControl WestBarnesLbl =
(System.Web.UI.HtmlControls.HtmlContainerControl)FormView1.FindControl("WBLLbl");
WestBarnesLbl.InnerText = "West Barnes";

System.Web.UI.HtmlControls.HtmlContainerControl WestBarnesStat =
(System.Web.UI.HtmlControls.HtmlContainerControl)FormView1.FindControl("WB");
WestBarnesStat.InnerText =
ds.Tables[0].Rows[0]["WestBarnes"].ToString();

System.Web.UI.HtmlControls.HtmlContainerControl WimbledonLbl =
(System.Web.UI.HtmlControls.HtmlContainerControl)FormView1.FindControl("WLBL");
WimbledonLbl.InnerText = "Wimbledon";

System.Web.UI.HtmlControls.HtmlContainerControl WimbledonStat =
(System.Web.UI.HtmlControls.HtmlContainerControl)FormView1.FindControl("W");
WimbledonStat.InnerText =
ds.Tables[0].Rows[0]["Wimbledon"].ToString();

if (Var2.Visible && Var2.SelectedIndex != 0)

```

```

        {
            //Second Table
            //New connection
            System.Data.SqlClient.SqlConnection con2 = new
System.Data.SqlClient.SqlConnection("Data
Source=.\SQLEXPRESS;AttachDbFilename=|DataDirectory|\\Wards.mdf;Integrated
Security=True;User Instance=True");
            System.Data.DataSet ds2 = new System.Data.DataSet();
            string SelectCommand2 = "SELECT * FROM WardInformation WHERE
InformationTitle = '" + Var2.Selected.Value.ToString() + "'";
            System.Data.SqlClient.SqlDataAdapter DataCommand2 = new
System.Data.SqlClient.SqlDataAdapter(SelectCommand2, con2);
            DataCommand2.Fill(ds2);

            //Manually populate again
            System.Web.UI.HtmlControls.HtmlContainerControl Var2thead =
(System.Web.UI.HtmlControls.HtmlContainerControl)FormView2.FindControl("Var2h
d");
            Var2thead.InnerText = "Information title";

            System.Web.UI.HtmlControls.HtmlContainerControl Var2thead2 =
(System.Web.UI.HtmlControls.HtmlContainerControl)FormView2.FindControl("Var2h
d2");
            Var2thead2.InnerText =
(string)ds2.Tables[0].Rows[0]["InformationTitle"];

            System.Web.UI.HtmlControls.HtmlContainerControl Var2AbbeyLbl
=
(System.Web.UI.HtmlControls.HtmlContainerControl)FormView2.FindControl("Var2A
bbeyLbl");
            Var2AbbeyLbl.InnerText = "Abbey";

            System.Web.UI.HtmlControls.HtmlContainerControl Var2AbbeyStat
=
(System.Web.UI.HtmlControls.HtmlContainerControl)FormView2.FindControl("Var2A
bbey");
            Var2AbbeyStat.InnerText =
ds2.Tables[0].Rows[0]["Abbey"].ToString();

            System.Web.UI.HtmlControls.HtmlContainerControl
Var2CannonHillLbl =
(System.Web.UI.HtmlControls.HtmlContainerControl)FormView2.FindControl("Var2C
HLbl");
            Var2CannonHillLbl.InnerText = "Cannon Hill";

            System.Web.UI.HtmlControls.HtmlContainerControl
Var2CannonHillStat =
(System.Web.UI.HtmlControls.HtmlContainerControl)FormView2.FindControl("Var2C
H");
            Var2CannonHillStat.InnerText =
ds2.Tables[0].Rows[0]["CannonHill"].ToString();

            System.Web.UI.HtmlControls.HtmlContainerControl
Var2ColliersWoodLbl =

```

```

(System.Web.UI.HtmlControls.HtmlContainerControl)FormView2.FindControl("Var2C
WLbl");
        Var2ColliersWoodLbl.InnerText = "Colliers Wood";

        System.Web.UI.HtmlControls.HtmlContainerControl
Var2ColliersWoodStat =
(System.Web.UI.HtmlControls.HtmlContainerControl)FormView2.FindControl("Var2C
W");
        Var2ColliersWoodStat.InnerText =
ds2.Tables[0].Rows[0]["ColliersWood"].ToString();

        System.Web.UI.HtmlControls.HtmlContainerControl
Var2CricketGreenLbl =
(System.Web.UI.HtmlControls.HtmlContainerControl)FormView2.FindControl("Var2C
rGLbl");
        Var2CricketGreenLbl.InnerText = "Cricket Green";

        System.Web.UI.HtmlControls.HtmlContainerControl
Var2CricketGreenStat =
(System.Web.UI.HtmlControls.HtmlContainerControl)FormView2.FindControl("Var2C
G");
        Var2CricketGreenStat.InnerText =
ds2.Tables[0].Rows[0]["CricketGreen"].ToString();

        System.Web.UI.HtmlControls.HtmlContainerControl
Var2DundonaldLbl =
(System.Web.UI.HtmlControls.HtmlContainerControl)FormView2.FindControl("Var2D
DLbl");
        Var2DundonaldLbl.InnerText = "Dundonald";

        System.Web.UI.HtmlControls.HtmlContainerControl
Var2DundonaldStat =
(System.Web.UI.HtmlControls.HtmlContainerControl)FormView2.FindControl("Var2D
D");
        Var2DundonaldStat.InnerText =
ds2.Tables[0].Rows[0]["Dundonald"].ToString();

        System.Web.UI.HtmlControls.HtmlContainerControl
Var2FiggesMarshLbl =
(System.Web.UI.HtmlControls.HtmlContainerControl)FormView2.FindControl("Var2F
MLbl");
        Var2FiggesMarshLbl.InnerText = "Figges Marsh";

        System.Web.UI.HtmlControls.HtmlContainerControl
Var2FiggesMarshStat =
(System.Web.UI.HtmlControls.HtmlContainerControl)FormView2.FindControl("Var2F
M");
        Var2FiggesMarshStat.InnerText =
ds2.Tables[0].Rows[0]["FiggesMarsh"].ToString();

        System.Web.UI.HtmlControls.HtmlContainerControl
Var2GraveneyLbl =
(System.Web.UI.HtmlControls.HtmlContainerControl)FormView2.FindControl("Var2G
Lbl");

```

```

        Var2GraveneyLbl.InnerText = "Graveney";

        System.Web.UI.HtmlControls.HtmlContainerControl
Var2GraveneyStat =
(System.Web.UI.HtmlControls.HtmlContainerControl)FormView2.FindControl("Var2G
");
        Var2GraveneyStat.InnerText =
ds2.Tables[0].Rows[0]["Graveney"].ToString();

        System.Web.UI.HtmlControls.HtmlContainerControl
Var2HillsideLbl =
(System.Web.UI.HtmlControls.HtmlContainerControl)FormView2.FindControl("Var2H
Lbl");
        Var2HillsideLbl.InnerText = "Hillside";

        System.Web.UI.HtmlControls.HtmlContainerControl
Var2HillsideStat =
(System.Web.UI.HtmlControls.HtmlContainerControl)FormView2.FindControl("Var2H
");
        Var2HillsideStat.InnerText =
ds2.Tables[0].Rows[0]["Hillside"].ToString();

        System.Web.UI.HtmlControls.HtmlContainerControl
Var2LavenderFieldsLbl =
(System.Web.UI.HtmlControls.HtmlContainerControl)FormView2.FindControl("Var2L
FLbl");
        Var2LavenderFieldsLbl.InnerText = "Lavender Fields";

        System.Web.UI.HtmlControls.HtmlContainerControl
Var2LavenderFieldsStat =
(System.Web.UI.HtmlControls.HtmlContainerControl)FormView2.FindControl("Var2L
F");
        Var2LavenderFieldsStat.InnerText =
ds2.Tables[0].Rows[0]["LavenderFields"].ToString();

        System.Web.UI.HtmlControls.HtmlContainerControl
Var2LongthorntonLbl =
(System.Web.UI.HtmlControls.HtmlContainerControl)FormView2.FindControl("Var2L
TLbl");
        Var2LongthorntonLbl.InnerText = "Longthornton";

        System.Web.UI.HtmlControls.HtmlContainerControl
Var2LongthorntonStat =
(System.Web.UI.HtmlControls.HtmlContainerControl)FormView2.FindControl("Var2L
T");
        Var2LongthorntonStat.InnerText =
ds2.Tables[0].Rows[0]["Longthornton"].ToString();

        System.Web.UI.HtmlControls.HtmlContainerControl
Var2LowerMordenLbl =
(System.Web.UI.HtmlControls.HtmlContainerControl)FormView2.FindControl("Var2L
MLbl");
        Var2LowerMordenLbl.InnerText = "Lower Morden";

```

```

        System.Web.UI.HtmlControls.HtmlContainerControl
Var2LowerMordenStat =
(System.Web.UI.HtmlControls.HtmlContainerControl)FormView2.FindControl("Var2LM");
        Var2LowerMordenStat.InnerText =
ds2.Tables[0].Rows[0]["LowerMorden"].ToString();

        System.Web.UI.HtmlControls.HtmlContainerControl
Var2MertonParkLbl =
(System.Web.UI.HtmlControls.HtmlContainerControl)FormView2.FindControl("Var2MPLbl");
        Var2MertonParkLbl.InnerText = "Merton Park";

        System.Web.UI.HtmlControls.HtmlContainerControl
Var2MertonParkStat =
(System.Web.UI.HtmlControls.HtmlContainerControl)FormView2.FindControl("Var2MP");
        Var2MertonParkStat.InnerText =
ds2.Tables[0].Rows[0]["MertonPark"].ToString();

        System.Web.UI.HtmlControls.HtmlContainerControl
Var2PollardsHillLbl =
(System.Web.UI.HtmlControls.HtmlContainerControl)FormView2.FindControl("Var2PHLbl");
        Var2PollardsHillLbl.InnerText = "Pollards Hill";

        System.Web.UI.HtmlControls.HtmlContainerControl
Var2PollardsHillStat =
(System.Web.UI.HtmlControls.HtmlContainerControl)FormView2.FindControl("Var2PH");
        Var2PollardsHillStat.InnerText =
ds2.Tables[0].Rows[0]["PollardsHill"].ToString();

        System.Web.UI.HtmlControls.HtmlContainerControl
Var2RavensburyLbl =
(System.Web.UI.HtmlControls.HtmlContainerControl)FormView2.FindControl("Var2RBLbl");
        Var2RavensburyLbl.InnerText = "Ravensbury";

        System.Web.UI.HtmlControls.HtmlContainerControl
Var2RavensburyStat =
(System.Web.UI.HtmlControls.HtmlContainerControl)FormView2.FindControl("Var2RB");
        Var2RavensburyStat.InnerText =
ds2.Tables[0].Rows[0]["Ravensbury"].ToString();

        System.Web.UI.HtmlControls.HtmlContainerControl
Var2RaynesParkLbl =
(System.Web.UI.HtmlControls.HtmlContainerControl)FormView2.FindControl("Var2RPLbl");
        Var2RaynesParkLbl.InnerText = "Raynes Park";

        System.Web.UI.HtmlControls.HtmlContainerControl
Var2RaynesParkStat =

```

```

(System.Web.UI.HtmlControls.HtmlContainerControl) FormView2.FindControl("Var2R
P");
        Var2RaynesParkStat.InnerText =
ds2.Tables[0].Rows[0]["RaynesPark"].ToString();

        System.Web.UI.HtmlControls.HtmlContainerControl
Var2StHelierLbl =
(System.Web.UI.HtmlControls.HtmlContainerControl) FormView2.FindControl("Var2S
HLbl");
        Var2StHelierLbl.InnerText = "St. Helier";

        System.Web.UI.HtmlControls.HtmlContainerControl
Var2StHelierStat =
(System.Web.UI.HtmlControls.HtmlContainerControl) FormView2.FindControl("Var2S
H");
        Var2StHelierStat.InnerText =
ds2.Tables[0].Rows[0]["StHelier"].ToString();

        System.Web.UI.HtmlControls.HtmlContainerControl
Var2TrinityLbl =
(System.Web.UI.HtmlControls.HtmlContainerControl) FormView2.FindControl("Var2T
Lbl");
        Var2TrinityLbl.InnerText = "Trinity";

        System.Web.UI.HtmlControls.HtmlContainerControl
Var2TrinityStat =
(System.Web.UI.HtmlControls.HtmlContainerControl) FormView2.FindControl("Var2T
");
        Var2TrinityStat.InnerText =
ds2.Tables[0].Rows[0]["Trinity"].ToString();

        System.Web.UI.HtmlControls.HtmlContainerControl
Var2VillageLbl =
(System.Web.UI.HtmlControls.HtmlContainerControl) FormView2.FindControl("Var2V
Lbl");
        Var2VillageLbl.InnerText = "Village";

        System.Web.UI.HtmlControls.HtmlContainerControl
Var2VillageStat =
(System.Web.UI.HtmlControls.HtmlContainerControl) FormView2.FindControl("Var2V
");
        Var2VillageStat.InnerText =
ds2.Tables[0].Rows[0]["Village"].ToString();

        System.Web.UI.HtmlControls.HtmlContainerControl
Var2WestBarnesLbl =
(System.Web.UI.HtmlControls.HtmlContainerControl) FormView2.FindControl("Var2W
BLbl");
        Var2WestBarnesLbl.InnerText = "West Barnes";

        System.Web.UI.HtmlControls.HtmlContainerControl
Var2WestBarnesStat =
(System.Web.UI.HtmlControls.HtmlContainerControl) FormView2.FindControl("Var2W
B");

```

```

                Var2WestBarnesStat.InnerText =
ds2.Tables[0].Rows[0]["WestBarnes"].ToString();

                System.Web.UI.HtmlControls.HtmlContainerControl
Var2WimbletonLbl =
(System.Web.UI.HtmlControls.HtmlContainerControl)FormView2.FindControl("Var2W
Lbl");
                Var2WimbletonLbl.InnerText = "Wimbleton";

                System.Web.UI.HtmlControls.HtmlContainerControl
Var2WimbletonStat =
(System.Web.UI.HtmlControls.HtmlContainerControl)FormView2.FindControl("Var2W
");
                Var2WimbletonStat.InnerText =
ds2.Tables[0].Rows[0]["Wimbleton"].ToString();
        }

        if (Var3.Visible && Var3.SelectedIndex != 0)
        {
                //Third Table
                //Third Connection
                System.Data.SqlClient.SqlConnection con3 = new
System.Data.SqlClient.SqlConnection("Data
Source=.\SQLEXPRESS;AttachDbFilename=|DataDirectory|\\Wards.mdf;Integrated
Security=True;User Instance=True");
                System.Data.DataSet ds3 = new System.Data.DataSet();
                string SelectCommand3 = "SELECT * FROM WardInformation WHERE
InformationTitle = '" + Var3.SelectedValue.ToString() + "'";
                System.Data.SqlClient.SqlDataAdapter DataCommand3 = new
System.Data.SqlClient.SqlDataAdapter(SelectCommand3, con3);
                DataCommand3.Fill(ds3);

                //Third time manually creating table. Fun?
                System.Web.UI.HtmlControls.HtmlContainerControl Var2thead =
(System.Web.UI.HtmlControls.HtmlContainerControl)FormView3.FindControl("Var2h
d");
                Var2thead.InnerText = "Information title";

                System.Web.UI.HtmlControls.HtmlContainerControl Var2thead2 =
(System.Web.UI.HtmlControls.HtmlContainerControl)FormView3.FindControl("Var2h
d2");
                Var2thead2.InnerText =
(string)ds3.Tables[0].Rows[0]["InformationTitle"];

                System.Web.UI.HtmlControls.HtmlContainerControl Var2AbbeyLbl
=
(System.Web.UI.HtmlControls.HtmlContainerControl)FormView3.FindControl("Var2A
bbeyLbl");
                Var2AbbeyLbl.InnerText = "Abbey";

                System.Web.UI.HtmlControls.HtmlContainerControl Var2AbbeyStat
=
(System.Web.UI.HtmlControls.HtmlContainerControl)FormView3.FindControl("Var2A
bbey");

```

```

                Var2AbbeyStat.InnerText =
ds3.Tables[0].Rows[0]["Abbey"].ToString();

                System.Web.UI.HtmlControls.HtmlContainerControl
Var2CannonHillLbl =
(System.Web.UI.HtmlControls.HtmlContainerControl)FormView3.FindControl("Var2C
HLbl");
                Var2CannonHillLbl.InnerText = "Cannon Hill";

                System.Web.UI.HtmlControls.HtmlContainerControl
Var2CannonHillStat =
(System.Web.UI.HtmlControls.HtmlContainerControl)FormView3.FindControl("Var2C
H");
                Var2CannonHillStat.InnerText =
ds3.Tables[0].Rows[0]["CannonHill"].ToString();

                System.Web.UI.HtmlControls.HtmlContainerControl
Var2ColliersWoodLbl =
(System.Web.UI.HtmlControls.HtmlContainerControl)FormView3.FindControl("Var2C
WLbl");
                Var2ColliersWoodLbl.InnerText = "Colliers Wood";

                System.Web.UI.HtmlControls.HtmlContainerControl
Var2ColliersWoodStat =
(System.Web.UI.HtmlControls.HtmlContainerControl)FormView3.FindControl("Var2C
W");
                Var2ColliersWoodStat.InnerText =
ds3.Tables[0].Rows[0]["ColliersWood"].ToString();

                System.Web.UI.HtmlControls.HtmlContainerControl
Var2CricketGreenLbl =
(System.Web.UI.HtmlControls.HtmlContainerControl)FormView3.FindControl("Var2C
rGLbl");
                Var2CricketGreenLbl.InnerText = "Cricket Green";

                System.Web.UI.HtmlControls.HtmlContainerControl
Var2CricketGreenStat =
(System.Web.UI.HtmlControls.HtmlContainerControl)FormView3.FindControl("Var2C
G");
                Var2CricketGreenStat.InnerText =
ds3.Tables[0].Rows[0]["CricketGreen"].ToString();

                System.Web.UI.HtmlControls.HtmlContainerControl
Var2DundonaldLbl =
(System.Web.UI.HtmlControls.HtmlContainerControl)FormView3.FindControl("Var2D
DLbl");
                Var2DundonaldLbl.InnerText = "Dundonald";

                System.Web.UI.HtmlControls.HtmlContainerControl
Var2DundonaldStat =
(System.Web.UI.HtmlControls.HtmlContainerControl)FormView3.FindControl("Var2D
D");
                Var2DundonaldStat.InnerText =
ds3.Tables[0].Rows[0]["Dundonald"].ToString();

```

```

        System.Web.UI.HtmlControls.HtmlContainerControl
Var2FiggesMarshLbl =
(System.Web.UI.HtmlControls.HtmlContainerControl)FormView3.FindControl("Var2F
MLbl");
        Var2FiggesMarshLbl.InnerText = "Figges Marsh";

        System.Web.UI.HtmlControls.HtmlContainerControl
Var2FiggesMarshStat =
(System.Web.UI.HtmlControls.HtmlContainerControl)FormView3.FindControl("Var2F
M");
        Var2FiggesMarshStat.InnerText =
ds3.Tables[0].Rows[0]["FiggesMarsh"].ToString();

        System.Web.UI.HtmlControls.HtmlContainerControl
Var2GraveneyLbl =
(System.Web.UI.HtmlControls.HtmlContainerControl)FormView3.FindControl("Var2G
Lbl");
        Var2GraveneyLbl.InnerText = "Graveney";

        System.Web.UI.HtmlControls.HtmlContainerControl
Var2GraveneyStat =
(System.Web.UI.HtmlControls.HtmlContainerControl)FormView3.FindControl("Var2G
");
        Var2GraveneyStat.InnerText =
ds3.Tables[0].Rows[0]["Graveney"].ToString();

        System.Web.UI.HtmlControls.HtmlContainerControl
Var2HillsideLbl =
(System.Web.UI.HtmlControls.HtmlContainerControl)FormView3.FindControl("Var2H
Lbl");
        Var2HillsideLbl.InnerText = "Hillside";

        System.Web.UI.HtmlControls.HtmlContainerControl
Var2HillsideStat =
(System.Web.UI.HtmlControls.HtmlContainerControl)FormView3.FindControl("Var2H
");
        Var2HillsideStat.InnerText =
ds3.Tables[0].Rows[0]["Hillside"].ToString();

        System.Web.UI.HtmlControls.HtmlContainerControl
Var2LavenderFieldsLbl =
(System.Web.UI.HtmlControls.HtmlContainerControl)FormView3.FindControl("Var2L
FLbl");
        Var2LavenderFieldsLbl.InnerText = "Lavender Fields";

        System.Web.UI.HtmlControls.HtmlContainerControl
Var2LavenderFieldsStat =
(System.Web.UI.HtmlControls.HtmlContainerControl)FormView3.FindControl("Var2L
F");
        Var2LavenderFieldsStat.InnerText =
ds3.Tables[0].Rows[0]["LavenderFields"].ToString();

```

```

        System.Web.UI.HtmlControls.HtmlContainerControl
Var2LongthorntonLbl =
(System.Web.UI.HtmlControls.HtmlContainerControl)FormView3.FindControl("Var2L
TLbl");
        Var2LongthorntonLbl.InnerText = "Longthornton";

        System.Web.UI.HtmlControls.HtmlContainerControl
Var2LongthorntonStat =
(System.Web.UI.HtmlControls.HtmlContainerControl)FormView3.FindControl("Var2L
T");
        Var2LongthorntonStat.InnerText =
ds3.Tables[0].Rows[0]["Longthornton"].ToString();

        System.Web.UI.HtmlControls.HtmlContainerControl
Var2LowerMordenLbl =
(System.Web.UI.HtmlControls.HtmlContainerControl)FormView3.FindControl("Var2L
MLbl");
        Var2LowerMordenLbl.InnerText = "Lower Morden";

        System.Web.UI.HtmlControls.HtmlContainerControl
Var2LowerMordenStat =
(System.Web.UI.HtmlControls.HtmlContainerControl)FormView3.FindControl("Var2L
M");
        Var2LowerMordenStat.InnerText =
ds3.Tables[0].Rows[0]["LowerMorden"].ToString();

        System.Web.UI.HtmlControls.HtmlContainerControl
Var2MertonParkLbl =
(System.Web.UI.HtmlControls.HtmlContainerControl)FormView3.FindControl("Var2M
PLbl");
        Var2MertonParkLbl.InnerText = "Merton Park";

        System.Web.UI.HtmlControls.HtmlContainerControl
Var2MertonParkStat =
(System.Web.UI.HtmlControls.HtmlContainerControl)FormView3.FindControl("Var2M
P");
        Var2MertonParkStat.InnerText =
ds3.Tables[0].Rows[0]["MertonPark"].ToString();

        System.Web.UI.HtmlControls.HtmlContainerControl
Var2PollardsHillLbl =
(System.Web.UI.HtmlControls.HtmlContainerControl)FormView3.FindControl("Var2P
HLbl");
        Var2PollardsHillLbl.InnerText = "Pollards Hill";

        System.Web.UI.HtmlControls.HtmlContainerControl
Var2PollardsHillStat =
(System.Web.UI.HtmlControls.HtmlContainerControl)FormView3.FindControl("Var2P
H");
        Var2PollardsHillStat.InnerText =
ds3.Tables[0].Rows[0]["PollardsHill"].ToString();

        System.Web.UI.HtmlControls.HtmlContainerControl
Var2RavensburyLbl =

```

```

(System.Web.UI.HtmlControls.HtmlContainerControl)FormView3.FindControl("Var2R
BLbl");
        Var2RavensburyLbl.InnerText = "Ravensbury";

        System.Web.UI.HtmlControls.HtmlContainerControl
Var2RavensburyStat =
(System.Web.UI.HtmlControls.HtmlContainerControl)FormView3.FindControl("Var2R
B");
        Var2RavensburyStat.InnerText =
ds3.Tables[0].Rows[0]["Ravensbury"].ToString();

        System.Web.UI.HtmlControls.HtmlContainerControl
Var2RaynesParkLbl =
(System.Web.UI.HtmlControls.HtmlContainerControl)FormView3.FindControl("Var2R
PLbl");
        Var2RaynesParkLbl.InnerText = "Raynes Park";

        System.Web.UI.HtmlControls.HtmlContainerControl
Var2RaynesParkStat =
(System.Web.UI.HtmlControls.HtmlContainerControl)FormView3.FindControl("Var2R
P");
        Var2RaynesParkStat.InnerText =
ds3.Tables[0].Rows[0]["RaynesPark"].ToString();

        System.Web.UI.HtmlControls.HtmlContainerControl
Var2StHelierLbl =
(System.Web.UI.HtmlControls.HtmlContainerControl)FormView3.FindControl("Var2S
HLbl");
        Var2StHelierLbl.InnerText = "St. Helier";

        System.Web.UI.HtmlControls.HtmlContainerControl
Var2StHelierStat =
(System.Web.UI.HtmlControls.HtmlContainerControl)FormView3.FindControl("Var2S
H");
        Var2StHelierStat.InnerText =
ds3.Tables[0].Rows[0]["StHelier"].ToString();

        System.Web.UI.HtmlControls.HtmlContainerControl
Var2TrinityLbl =
(System.Web.UI.HtmlControls.HtmlContainerControl)FormView3.FindControl("Var2T
Lbl");
        Var2TrinityLbl.InnerText = "Trinity";

        System.Web.UI.HtmlControls.HtmlContainerControl
Var2TrinityStat =
(System.Web.UI.HtmlControls.HtmlContainerControl)FormView3.FindControl("Var2T
");
        Var2TrinityStat.InnerText =
ds3.Tables[0].Rows[0]["Trinity"].ToString();

        System.Web.UI.HtmlControls.HtmlContainerControl
Var2VillageLbl =
(System.Web.UI.HtmlControls.HtmlContainerControl)FormView3.FindControl("Var2V
Lbl");

```

```

        Var2VillageLbl.InnerText = "Village";

        System.Web.UI.HtmlControls.HtmlContainerControl
Var2VillageStat =
(System.Web.UI.HtmlControls.HtmlContainerControl)FormView3.FindControl("Var2V
");
        Var2VillageStat.InnerText =
ds3.Tables[0].Rows[0]["Village"].ToString();

        System.Web.UI.HtmlControls.HtmlContainerControl
Var2WestBarnesLbl =
(System.Web.UI.HtmlControls.HtmlContainerControl)FormView3.FindControl("Var2W
BLbl");
        Var2WestBarnesLbl.InnerText = "West Barnes";

        System.Web.UI.HtmlControls.HtmlContainerControl
Var2WestBarnesStat =
(System.Web.UI.HtmlControls.HtmlContainerControl)FormView3.FindControl("Var2W
B");
        Var2WestBarnesStat.InnerText =
ds3.Tables[0].Rows[0]["WestBarnes"].ToString();

        System.Web.UI.HtmlControls.HtmlContainerControl
Var2WimbledonLbl =
(System.Web.UI.HtmlControls.HtmlContainerControl)FormView3.FindControl("Var2W
Lbl");
        Var2WimbledonLbl.InnerText = "Wimbledon";

        System.Web.UI.HtmlControls.HtmlContainerControl
Var2WimbledonStat =
(System.Web.UI.HtmlControls.HtmlContainerControl)FormView3.FindControl("Var2W
");
        Var2WimbledonStat.InnerText =
(Math.Round((double)ds3.Tables[0].Rows[0]["Wimbledon"], 2)).ToString();
    }

    //Make the appropriate Tables visible to the user.
    FormView1.Visible = true;

    if (Var2.Visible == true && Var2.SelectedIndex != 0)
    {
        FormView2.Visible = true;
    }
    else
    {
        FormView2.Visible = false;
    }
    if (Var3.Visible == true && Var3.SelectedIndex != 0)
    {
        FormView3.Visible = true;
    }
    else
    {

```

```

        FormView3.Visible = false;
    }
}

protected void DropDownList1_SelectedIndexChanged(object sender,
EventArgs e)
{
}
</script>

<meta content="text/JScript" http-equiv="content-script-type" />
<title>Ward Profile</title>
</head>
<body>

<h1>
    Ward profiling in Merton
</h1>
<form method="post" action="WardProfile.aspx" runat="server">

    <asp:Label ID="SwitchLbl" runat="server" Style="z-index: 102; left:
692px; position: absolute;
        top: 26px" Text="Label" Width="151px"></asp:Label>

    <asp:Label ID="WarningLbl" runat="server" Style="z-index: 103; left: 7px;
position: absolute;
        top: 57px" Text="Please choose a data set for each available slot,
or remove a dataset." Visible="False" ForeColor="Red"
Width="496px"></asp:Label>

    <h2>
    <asp:Label ID="title1" runat="server" Style="z-index: 104; left: 15px;
position: absolute;
        top: 84px" Text="Background shade"></asp:Label>
    </h2>

    <asp:DropDownList ID="Color1" runat="server"
OnSelectedIndexChanged="DropDownList1_SelectedIndexChanged"
        Style="z-index: 100; left: 234px; position: absolute; top: 87px">
    <asp:ListItem>Choose color...</asp:ListItem>
    <asp:ListItem>blue</asp:ListItem>
    <asp:ListItem>green</asp:ListItem>
    <asp:ListItem>grey</asp:ListItem>
    <asp:ListItem>purple</asp:ListItem>
    <asp:ListItem>red</asp:ListItem>
    </asp:DropDownList>

```



```

    <asp:DropDownList ID="Var3" runat="server" Height="50px" Style="z-index:
109;
        left: 15px; position: absolute; top: 290px" Width="333px"
Visible="False" OnSelectedIndexChanged="Var3_SelectedIndexChanged"
DataSourceID="SqlDataSource1" DataTextField="InformationTitle"
DataValueField="InformationTitle" >
    </asp:DropDownList>

    <br />
    <br />
    <br />
    <br />

    <asp:Button ID="buttonAdd" runat="server" OnClientClick
="buttonAdd_Click" OnClick="buttonAdd_Click" Style="z-index: 110;
        left: 15px; position: absolute; top: 329px" Text="Add data set" />

    <asp:Button ID="Button2" runat="server" OnClick="Button2_Click"
Style="z-index: 111;
        left: 169px; position: absolute; top: 329px" Text="Remove data set"
/>

    <asp:Button ID="Button3" runat="server" OnClick="Button3_Click2"
Style="z-index: 112;
        left: 12px; position: absolute; top: 453px" Text="View data" />

    <h1>
    &nbsp;  </h1>

    <asp:Label ID="Label2" runat="server" Style="z-index: 114; left: 12px;
position: absolute;
        top: 397px" Text="Ward labels are disabled."
Width="164px"></asp:Label>

    <asp:Button ID="ToggleOn" runat="server" OnClick="ToggleOn_Click"
Style="z-index: 115;
        left: 224px; position: absolute; top: 394px" Text="Turn labels on" />

    <asp:Button ID="ToggleOff" runat="server" Style="z-index: 116; left:
224px; position: absolute;
        top: 394px" Text="Turn labels off" OnClick="ToggleOff_Click"
Visible="False" />

    <br />
    <br />
    <br />

    <asp:Label ID="DwnLdLbl" runat="server" Style="z-index: 117; left: 25px;
position: absolute;
        top: 512px" Text= "These maps are in SVG format. You need to install
<a href=' http://www.adobe.com/svg/viewer/install/'>Adobe SVG Viewer</a>
before you can view these maps." Visible="False" Width="657px"></asp:Label>

```

```
<br />
```

```
<iframe id = 'WardFrame' width = '1250' height = '650' runat = "server" style="z-index: 124; left: 25px; position: absolute; top: 547px; height: 520px;"></iframe>
```

```
<br />
```

```
<iframe id ='Graphs' width = '1250' height = '650' runat = "server" style="z-index: 124; left: 25px; position: absolute; top: 1104px; height: 520px;"></iframe>
```

```
<asp:SqlDataSource ID="SqlDataSource1" runat="server" ConnectionString="<%= $ ConnectionStrings:ConnectionString %>" SelectCommand="SELECT * FROM [WardInformation]" OnSelecting="SqlDataSource1_Selecting"></asp:SqlDataSource>
```

```
<asp:SqlDataSource ID="SqlDataSource2" runat="server" ConnectionString="<%= $ ConnectionStrings:ConnectionString %>" SelectCommand="SELECT * FROM [Limits]"></asp:SqlDataSource>
```

```
<asp:SqlDataSource ID="SqlDataSource3" runat="server" ConnectionString="<%= $ ConnectionStrings:ConnectionString %>" SelectCommand="SELECT * FROM [Sources]"></asp:SqlDataSource>
```

```
<asp:GridView ID="GridView3" runat="server" AutoGenerateColumns="False" DataSourceID="SqlDataSource3" Style="z-index: 118; left: 791px; position: absolute; top: 584px" Visible="False">  
<Columns>  
    <asp:BoundField DataField="Statistic" HeaderText="Statistic" SortExpression="Statistic" />  
    <asp:BoundField DataField="Source" HeaderText="Source" SortExpression="Source" />  
    <asp:BoundField DataField="LastUpdate" HeaderText="LastUpdate" SortExpression="LastUpdate" />  
</Columns>  
</asp:GridView>
```

```
<asp:GridView ID="GridView2" runat="server" AutoGenerateColumns="False" DataKeyNames="Statistic" DataSourceID="SqlDataSource2" Style="z-index: 119; left: 379px; position: absolute; top: 603px" Visible="False">  
<Columns>  
    <asp:BoundField DataField="Statistic" HeaderText="Statistic" ReadOnly="True" SortExpression="Statistic" />  
    <asp:BoundField DataField="LFirst" HeaderText="First" SortExpression="First" />  
    <asp:BoundField DataField="LSecond" HeaderText="Second" SortExpression="Second" />  
    <asp:BoundField DataField="LThird" HeaderText="Third" SortExpression="Third" />  
</Columns>  
</asp:GridView>
```

```

        <asp:BoundField DataField="LFourth" HeaderText="Fourth"
SortExpression="Fourth" />
    </Columns>
</asp:GridView>

<asp:GridView ID="GridView1" runat="server" AutoGenerateColumns="False"
    DataSourceID="SqlDataSource1"
    OnSelectedIndexChanged="GridView1_SelectedIndexChanged"
    Style="z-index: 120; left: 741px; position: absolute; top: 684px"
    DataKeyNames="InformationTitle" Visible="False">
    <Columns>
        <asp:BoundField DataField="InformationTitle"
HeaderText="InformationTitle" ReadOnly="True"
            SortExpression="InformationTitle" />
        <asp:BoundField DataField="Abbey" HeaderText="Abbey"
SortExpression="Abbey" />
        <asp:BoundField DataField="CannonHill" HeaderText="CannonHill"
SortExpression="CannonHill" />
        <asp:BoundField DataField="ColliersWood"
HeaderText="ColliersWood" SortExpression="ColliersWood" />
        <asp:BoundField DataField="CricketGreen"
HeaderText="CricketGreen" SortExpression="CricketGreen" />
        <asp:BoundField DataField="Dundonald" HeaderText="Dundonald"
SortExpression="Dundonald" />
        <asp:BoundField DataField="FiggesMarsh" HeaderText="FiggesMarsh"
SortExpression="FiggesMarsh" />
        <asp:BoundField DataField="Graveney" HeaderText="Graveney"
SortExpression="Graveney" />
        <asp:BoundField DataField="Hillside" HeaderText="Hillside"
SortExpression="Hillside" />
        <asp:BoundField DataField="LavenderFields"
HeaderText="LavenderFields" SortExpression="LavenderFields" />
        <asp:BoundField DataField="Longthornton"
HeaderText="Longthornton" SortExpression="Longthornton" />
        <asp:BoundField DataField="LowerMorden" HeaderText="LowerMorden"
SortExpression="LowerMorden" />
        <asp:BoundField DataField="MertonPark" HeaderText="MertonPark"
SortExpression="MertonPark" />
        <asp:BoundField DataField="PollardsHill"
HeaderText="PollardsHill" SortExpression="PollardsHill" />
        <asp:BoundField DataField="Ravensbury" HeaderText="Ravensbury"
SortExpression="Ravensbury" />
        <asp:BoundField DataField="RaynesPark" HeaderText="RaynesPark"
SortExpression="RaynesPark" />
        <asp:BoundField DataField="StHelier" HeaderText="StHelier"
SortExpression="StHelier" />
        <asp:BoundField DataField="Trinity" HeaderText="Trinity"
SortExpression="Trinity" />
        <asp:BoundField DataField="Village" HeaderText="Village"
SortExpression="Village" />
        <asp:BoundField DataField="WestBarnes" HeaderText="WestBarnes"
SortExpression="WestBarnes" />
        <asp:BoundField DataField="Wimbledon" HeaderText="Wimbledon"
SortExpression="Wimbledon" />
    </Columns>
</asp:GridView>

```

```

        </Columns>
    </asp:GridView>

<br />
<br />
<br />

    <asp:FormView ID="FormView1" CellSpacing = "1" runat="server"
DataSourceID="SqlDataSource1" style="z-index: 121; left: 25px; position:
absolute; top: 1697px" Visible="False" Height="544px">
    <ItemTemplate>
        <table border = "1" rules = "all">
            <thead class = "t1" runat="server">
                <tr>
                    <td id="hd" class = "t1" runat="server">
                        <%# Eval("InformationTitle")%>
                    </td>
                    <td id = "hd2" class = "t1" runat="server">
                        <%# Eval ("InformationTitle") %>
                    </td>
                </thead>
                <tbody class = "t1" runat="server">
                    <tr>
                        <td id = "AbbeyLbl" class = "t1" runat = "server">
                            <%# Eval("Abbey")%>
                        </td>
                        <td id = "Abbey" class = "t1" runat = "server">
                            <%# Eval("Abbey")%>
                        </td> </tr>
                    <tr>
                        <td id = "CHLbl" class = "t1" runat = "server">
                            <%# Eval ("CannonHill")%>
                        </td>
                        <td id = "CH" class = "t1" runat = "server">
                            <%# Eval ("CannonHill")%>
                        </td> </tr>
                    <tr>
                        <td id = "CWLbl" class = "t1" runat = "server">
                            <%# Eval ("ColliersWood")%>
                        </td>
                        <td id = "CW" class = "t1" runat = "server">
                            <%# Eval ("ColliersWood")%>
                        </td> </tr>
                    <tr>
                        <td id = "CrGLbl" class = "t1" runat = "server">
                            <%# Eval ("CricketGreen")%>
                        </td>
                        <td id = "CG" class = "t1" runat = "server">
                            <%# Eval ("CricketGreen")%>
                        </td> </tr>
                    <tr>
                        <td id = "DDLbl" class = "t1" runat = "server">
                            <%# Eval ("Dundonald")%>

```

```

    </td>
<td id = "DD" class = "t1" runat = "server">
    <%# Eval ("Dundonald")%>
</td> </tr>
<tr>
<td id = "FMLbl" class = "t1" runat = "server">
    <%# Eval ("FiggesMarsh")%>
    </td>
<td id = "FM" class = "t1" runat = "server">
    <%# Eval ("FiggesMarsh")%>
</td> </tr>
<tr>
<td id = "GLbl" class = "t1" runat = "server">
    <%# Eval ("Graveney")%>
    </td>
<td id = "G" class = "t1" runat = "server">
    <%# Eval ("Graveney")%>
</td> </tr>
<tr>
<td id = "HLbl" class = "t1" runat = "server">
    <%# Eval ("Hillside")%>
    </td>
<td id = "H" class = "t1" runat = "server">
    <%# Eval ("Hillside")%>
</td> </tr>
<tr>
<td id = "LFLbl" class = "t1" runat = "server">
    <%# Eval ("LavenderFields")%>
    </td>
<td id = "LF" class = "t1" runat = "server">
    <%# Eval ("LavenderFields")%>
</td> </tr>
<tr>
<td id = "LTLbl" class = "t1" runat = "server">
    <%# Eval ("Longthornton")%>
    </td>
<td id = "LT" class = "t1" runat = "server">
    <%# Eval ("Longthornton")%>
</td> </tr>
<tr>
<td id = "LMLbl" class = "t1" runat = "server">
    <%# Eval ("LowerMorden")%>
    </td>
<td id = "LM" class = "t1" runat = "server">
    <%# Eval ("LowerMorden")%>
</td> </tr>
<tr>
<td id = "MPLbl" class = "t1" runat = "server">
    <%# Eval ("MertonPark")%>
    </td>
<td id = "MP" class = "t1" runat = "server">
    <%# Eval ("MertonPark")%>
</td> </tr>
<tr>

```

```

<td id = "PHLlbl" class = "t1" runat = "server">
  <%# Eval ("PollardsHill")%>
</td>
<td id = "PH" class = "t1" runat = "server">
  <%# Eval ("PollardsHill")%>
</td> </tr>
<tr>
<td id = "RBLlbl" class = "t1" runat = "server">
  <%# Eval ("Ravensbury")%>
</td>
<td id = "RB" class = "t1" runat = "server">
  <%# Eval ("Ravensbury")%>
</td> </tr>
<tr>
<td id = "RPLlbl" class = "t1" runat = "server">
  <%# Eval ("RaynesPark")%>
</td>
<td id = "RP" class = "t1" runat = "server">
  <%# Eval ("RaynesPark")%>
</td> </tr>
<tr>
<td id = "SHLlbl" class = "t1" runat = "server">
  <%# Eval ("StHelier")%>
</td>
<td id = "SH" class = "t1" runat = "server">
  <%# Eval ("StHelier")%>
</td> </tr>
<tr>
<td id = "TLlbl" class = "t1" runat = "server">
  <%# Eval ("Trinity")%>
</td>
<td id = "T" class = "t1" runat = "server">
  <%# Eval ("Trinity")%>
</td> </tr>
<tr>
<td id = "VLlbl" class = "t1" runat = "server">
  <%# Eval ("Village")%>
</td>
<td id = "V" class = "t1" runat = "server">
  <%# Eval ("Village")%>
</td> </tr>
<tr>
<td id = "WBLlbl" class = "t1" runat = "server">
  <%# Eval ("WestBarnes")%>
</td>
<td id = "WB" class = "t1" runat = "server">
  <%# Eval ("WestBarnes")%>
</td> </tr>
<tr>
<td id = "WLlbl" class = "t1" runat = "server">
  <%# Eval ("Wimbledon")%>
</td>
<td id = "W" class = "t1" runat = "server">
  <%# Eval ("Wimbledon")%>

```



```

<tr>
<td id = "Var2DDLbl" class = "t1" runat = "server">
  <%# Eval ("Dundonald")%>
</td>
<td id = "Var2DD" class = "t1" runat = "server">
  <%# Eval ("Dundonald")%>
</td> </tr>
<tr>
<td id = "Var2FMLbl" class = "t1" runat = "server">
  <%# Eval ("FiggesMarsh")%>
</td>
<td id = "Var2FM" class = "t1" runat = "server">
  <%# Eval ("FiggesMarsh")%>
</td> </tr>
<tr>
<td id = "Var2GLbl" class = "t1" runat = "server">
  <%# Eval ("Graveney")%>
</td>
<td id = "Var2G" class = "t1" runat = "server">
  <%# Eval ("Graveney")%>
</td> </tr>
<tr>
<td id = "Var2HLbl" class = "t1" runat = "server">
  <%# Eval ("Hillside")%>
</td>
<td id = "Var2H" class = "t1" runat = "server">
  <%# Eval ("Hillside")%>
</td> </tr>
<tr>
<td id = "Var2LFLbl" class = "t1" runat = "server">
  <%# Eval ("LavenderFields")%>
</td>
<td id = "Var2LF" class = "t1" runat = "server">
  <%# Eval ("LavenderFields")%>
</td> </tr>
<tr>
<td id = "Var2LTLbl" class = "t1" runat = "server">
  <%# Eval ("Longthornton")%>
</td>
<td id = "Var2LT" class = "t1" runat = "server">
  <%# Eval ("Longthornton")%>
</td> </tr>
<tr>
<td id = "Var2LMLbl" class = "t1" runat = "server">
  <%# Eval ("LowerMorden")%>
</td>
<td id = "Var2LM" class = "t1" runat = "server">
  <%# Eval ("LowerMorden")%>
</td> </tr>
<tr>
<td id = "Var2MPLbl" class = "t1" runat = "server">
  <%# Eval ("MertonPark")%>
</td>
<td id = "Var2MP" class = "t1" runat = "server">

```

```

        <%# Eval ("MertonPark")%>
</td> </tr>
<tr>
<td id = "Var2PHLlbl" class = "t1" runat = "server">
    <%# Eval ("PollardsHill")%>
</td>
<td id = "Var2PH" class = "t1" runat = "server">
    <%# Eval ("PollardsHill")%>
</td> </tr>
<tr>
<td id = "Var2RBLlbl" class = "t1" runat = "server">
    <%# Eval ("Ravensbury")%>
</td>
<td id = "Var2RB" class = "t1" runat = "server">
    <%# Eval ("Ravensbury")%>
</td> </tr>
<tr>
<td id = "Var2RPLlbl" class = "t1" runat = "server">
    <%# Eval ("RaynesPark")%>
</td>
<td id = "Var2RP" class = "t1" runat = "server">
    <%# Eval ("RaynesPark")%>
</td> </tr>
<tr>
<td id = "Var2SHLlbl" class = "t1" runat = "server">
    <%# Eval ("StHelier")%>
</td>
<td id = "Var2SH" class = "t1" runat = "server">
    <%# Eval ("StHelier")%>
</td> </tr>
<tr>
<td id = "Var2TLlbl" class = "t1" runat = "server">
    <%# Eval ("Trinity")%>
</td>
<td id = "Var2T" class = "t1" runat = "server">
    <%# Eval ("Trinity")%>
</td> </tr>
<tr>
<td id = "Var2VLlbl" class = "t1" runat = "server">
    <%# Eval ("Village")%>
</td>
<td id = "Var2V" class = "t1" runat = "server">
    <%# Eval ("Village")%>
</td> </tr>
<tr>
<td id = "Var2WBLlbl" class = "t1" runat = "server">
    <%# Eval ("WestBarnes")%>
</td>
<td id = "Var2WB" class = "t1" runat = "server">
    <%# Eval ("WestBarnes")%>
</td> </tr>
<tr>
<td id = "Var2WLlbl" class = "t1" runat = "server">
    <%# Eval ("Wimbledon")%>

```



```

<td id = "Var2CG" class = "t1" runat = "server">
  <%# Eval ("CricketGreen")%>
</td> </tr>
<tr>
<td id = "Var2DDLbl" class = "t1" runat = "server">
  <%# Eval ("Dundonald")%>
</td>
<td id = "Var2DD" class = "t1" runat = "server">
  <%# Eval ("Dundonald")%>
</td> </tr>
<tr>
<td id = "Var2FMLbl" class = "t1" runat = "server">
  <%# Eval ("FiggesMarsh")%>
</td>
<td id = "Var2FM" class = "t1" runat = "server">
  <%# Eval ("FiggesMarsh")%>
</td> </tr>
<tr>
<td id = "Var2GLbl" class = "t1" runat = "server">
  <%# Eval ("Graveney")%>
</td>
<td id = "Var2G" class = "t1" runat = "server">
  <%# Eval ("Graveney")%>
</td> </tr>
<tr>
<td id = "Var2HLbl" class = "t1" runat = "server">
  <%# Eval ("Hillside")%>
</td>
<td id = "Var2H" class = "t1" runat = "server">
  <%# Eval ("Hillside")%>
</td> </tr>
<tr>
<td id = "Var2LFLbl" class = "t1" runat = "server">
  <%# Eval ("LavenderFields")%>
</td>
<td id = "Var2LF" class = "t1" runat = "server">
  <%# Eval ("LavenderFields")%>
</td> </tr>
<tr>
<td id = "Var2LTLbl" class = "t1" runat = "server">
  <%# Eval ("Longthornton")%>
</td>
<td id = "Var2LT" class = "t1" runat = "server">
  <%# Eval ("Longthornton")%>
</td> </tr>
<tr>
<td id = "Var2LMLbl" class = "t1" runat = "server">
  <%# Eval ("LowerMorden")%>
</td>
<td id = "Var2LM" class = "t1" runat = "server">
  <%# Eval ("LowerMorden")%>
</td> </tr>
<tr>
<td id = "Var2MPLbl" class = "t1" runat = "server">

```

```

        <%# Eval ("MertonPark")%>
    </td>
<td id = "Var2MP" class = "t1" runat = "server">
    <%# Eval ("MertonPark")%>
</td> </tr>
<tr>
<td id = "Var2PHLlbl" class = "t1" runat = "server">
    <%# Eval ("PollardsHill")%>
</td>
<td id = "Var2PH" class = "t1" runat = "server">
    <%# Eval ("PollardsHill")%>
</td> </tr>
<tr>
<td id = "Var2RBLlbl" class = "t1" runat = "server">
    <%# Eval ("Ravensbury")%>
</td>
<td id = "Var2RB" class = "t1" runat = "server">
    <%# Eval ("Ravensbury")%>
</td> </tr>
<tr>
<td id = "Var2RPLlbl" class = "t1" runat = "server">
    <%# Eval ("RaynesPark")%>
</td>
<td id = "Var2RP" class = "t1" runat = "server">
    <%# Eval ("RaynesPark")%>
</td> </tr>
<tr>
<td id = "Var2SHLlbl" class = "t1" runat = "server">
    <%# Eval ("StHelier")%>
</td>
<td id = "Var2SH" class = "t1" runat = "server">
    <%# Eval ("StHelier")%>
</td> </tr>
<tr>
<td id = "Var2TLlbl" class = "t1" runat = "server">
    <%# Eval ("Trinity")%>
</td>
<td id = "Var2T" class = "t1" runat = "server">
    <%# Eval ("Trinity")%>
</td> </tr>
<tr>
<td id = "Var2VLlbl" class = "t1" runat = "server">
    <%# Eval ("Village")%>
</td>
<td id = "Var2V" class = "t1" runat = "server">
    <%# Eval ("Village")%>
</td> </tr>
<tr>
<td id = "Var2WBLlbl" class = "t1" runat = "server">
    <%# Eval ("WestBarnes")%>
</td>
<td id = "Var2WB" class = "t1" runat = "server">
    <%# Eval ("WestBarnes")%>
</td> </tr>

```

```

        <tr>
        <td id = "Var2WLbl" class = "t1" runat = "server">
            <%# Eval ("Wimbledon")%>
        </td>
        <td id = "Var2W" class = "t1" runat = "server">
            <%# Eval ("Wimbledon")%>
        </td> </tr>

        </tbody>

        </table>
    </ItemTemplate>
</asp:FormView>
</form>
</body>
</html>

```

Update.aspx

```

<%@ Page Language="C#" %>

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">

<script runat="server">

    //At the beginning of the page
    protected void Page_Load(object sender, EventArgs e)
    {
        //Clear away confirmations from previous happenings
        DelConfirm.Visible = false;
        UpdtConfirm.Visible = false;

        //Set the switch label's text to include a link to the update system
        (not for public).
        SwitchLbl.Text = "<a href='WardProfile.aspx'>To display system </a>";
    }

    //User has checked that the information was correct and wants to proceed
    with the update
    protected void Confirm_Click(object sender, EventArgs e)
    {
        //New connection to database
        System.Data.SqlClient.SqlConnection con = new
        System.Data.SqlClient.SqlConnection("Data
        Source=.\SQLEXPRESS;AttachDbFilename=|DataDirectory|\\Wards.mdf;Integrated
        Security=True;User Instance=True");

        //If they're updating the values...
        if (checkLbl.Text == "Stat")
        {
            //For each statistic
            try

```

```

    {
        for (int i = 1; i < GridView1.Rows[1].Cells.Count; i++)
        {
            //Delete variable that's already in wardInformation
            System.Data.SqlClient.SqlCommand del = new
System.Data.SqlClient.SqlCommand("wardDelete", con);
            del.Parameters.Add("@Info",
System.Data.SqlDbType.VarChar).Value =
GridView1.HeaderRow.Cells[i].Text.ToString();
            del.CommandType =
System.Data.CommandType.StoredProcedure;

            //Add new Variable to wardInformation. Pass statistics
for each ward as parameters

            System.Data.SqlClient.SqlCommand cmd = new
System.Data.SqlClient.SqlCommand("wardChange", con);
            cmd.Parameters.Add("@Info",
System.Data.SqlDbType.VarChar).Value =
GridView1.HeaderRow.Cells[i].Text.ToString();
            cmd.Parameters.Add("@Abbey",
System.Data.SqlDbType.Float).Value =
Math.Round(float.Parse(GridView1.Rows[0].Cells[i].Text.ToString()), 2);
            cmd.Parameters.Add("@CannonHill",
System.Data.SqlDbType.Float).Value =
Math.Round(float.Parse(GridView1.Rows[1].Cells[i].Text.ToString()), 2);
            cmd.Parameters.Add("@ColliersWood",
System.Data.SqlDbType.Float).Value =
Math.Round(float.Parse(GridView1.Rows[2].Cells[i].Text.ToString()), 2);
            cmd.Parameters.Add("@CricketGreen",
System.Data.SqlDbType.Float).Value =
Math.Round(float.Parse(GridView1.Rows[3].Cells[i].Text.ToString()), 2);
            cmd.Parameters.Add("@Dundonald",
System.Data.SqlDbType.Float).Value =
Math.Round(float.Parse(GridView1.Rows[4].Cells[i].Text.ToString()), 2);
            cmd.Parameters.Add("@FiggesMarsh",
System.Data.SqlDbType.Float).Value =
Math.Round(float.Parse(GridView1.Rows[5].Cells[i].Text.ToString()), 2);
            cmd.Parameters.Add("@Graveney",
System.Data.SqlDbType.Float).Value =
Math.Round(float.Parse(GridView1.Rows[6].Cells[i].Text.ToString()), 2);
            cmd.Parameters.Add("@Hillside",
System.Data.SqlDbType.Float).Value =
Math.Round(float.Parse(GridView1.Rows[7].Cells[i].Text.ToString()), 2);
            cmd.Parameters.Add("@LavenderFields",
System.Data.SqlDbType.Float).Value =
Math.Round(float.Parse(GridView1.Rows[8].Cells[i].Text.ToString()), 2);
            cmd.Parameters.Add("@Longthornton",
System.Data.SqlDbType.Float).Value =
Math.Round(float.Parse(GridView1.Rows[9].Cells[i].Text.ToString()), 2);
            cmd.Parameters.Add("@LowerMorden",
System.Data.SqlDbType.Float).Value =
Math.Round(float.Parse(GridView1.Rows[10].Cells[i].Text.ToString()), 2);

```

```

        cmd.Parameters.Add("@MertonPark",
System.Data.SqlDbType.Float).Value =
Math.Round(float.Parse(GridView1.Rows[11].Cells[i].Text.ToString()), 2);
        cmd.Parameters.Add("@PollardsHill",
System.Data.SqlDbType.Float).Value =
Math.Round(float.Parse(GridView1.Rows[12].Cells[i].Text.ToString()), 2);
        cmd.Parameters.Add("@Ravensbury",
System.Data.SqlDbType.Float).Value =
Math.Round(float.Parse(GridView1.Rows[13].Cells[i].Text.ToString()), 2);
        cmd.Parameters.Add("@RaynesPark",
System.Data.SqlDbType.Float).Value =
Math.Round(float.Parse(GridView1.Rows[14].Cells[i].Text.ToString()), 2);
        cmd.Parameters.Add("@StHelier",
System.Data.SqlDbType.Float).Value =
Math.Round(float.Parse(GridView1.Rows[15].Cells[i].Text.ToString()), 2);
        cmd.Parameters.Add("@Trinity",
System.Data.SqlDbType.Float).Value =
Math.Round(float.Parse(GridView1.Rows[16].Cells[i].Text.ToString()), 2);
        cmd.Parameters.Add("@Village",
System.Data.SqlDbType.Float).Value =
Math.Round(float.Parse(GridView1.Rows[17].Cells[i].Text.ToString()), 2);
        cmd.Parameters.Add("@WestBarnes",
System.Data.SqlDbType.Float).Value =
Math.Round(float.Parse(GridView1.Rows[18].Cells[i].Text.ToString()), 2);
        cmd.Parameters.Add("@Wimbledon",
System.Data.SqlDbType.Float).Value =
Math.Round(float.Parse(GridView1.Rows[19].Cells[i].Text.ToString()), 2);
        cmd.CommandType =
System.Data.CommandType.StoredProcedure;

        //Delete Ranges for Variable from Limits
        System.Data.SqlClient.SqlCommand limDel = new
System.Data.SqlClient.SqlCommand("limitDelete", con);
        limDel.Parameters.Add("@Info",
System.Data.SqlDbType.VarChar).Value =
GridView1.HeaderRow.Cells[i].Text.ToString();
        limDel.CommandType =
System.Data.CommandType.StoredProcedure;

        //Put in new Limits
        System.Data.SqlClient.SqlCommand limAdd = new
System.Data.SqlClient.SqlCommand("limitChange", con);
        limAdd.Parameters.Add("@Info",
System.Data.SqlDbType.VarChar).Value =
GridView1.HeaderRow.Cells[i].Text.ToString();
        limAdd.Parameters.Add("@first",
System.Data.SqlDbType.Float).Value =
Math.Round(float.Parse(GridView1.Rows[20].Cells[i].Text.ToString()), 2);
        limAdd.Parameters.Add("@second",
System.Data.SqlDbType.Float).Value =
Math.Round(float.Parse(GridView1.Rows[21].Cells[i].Text.ToString()), 2);
        limAdd.Parameters.Add("@third",
System.Data.SqlDbType.Float).Value =
Math.Round(float.Parse(GridView1.Rows[22].Cells[i].Text.ToString()), 2);

```

```

        limAdd.Parameters.Add("@fourth",
System.Data.SqlDbType.Float).Value =
Math.Round(float.Parse(GridView1.Rows[23].Cells[i].Text.ToString()), 2);
        limAdd.CommandType =
System.Data.CommandType.StoredProcedure;

        //Execute all of them
con.Open();
del.ExecuteNonQuery();
cmd.ExecuteNonQuery();
limDel.ExecuteNonQuery();
limAdd.ExecuteNonQuery();
con.Close();

        //Let the user know that the update was succesful.
UpdtConfirm.Text = "Update successful.";
UpdtConfirm.Visible = true;

        //Re-bind the drop-down with the new database, so the new
variable can
        //be deleted if a mistake occurred
DropDelete.DataBind();
    }
}

        //If something typed in isn't what it should be, cut the
update process
catch (FormatException)
{
    UpdtConfirm.Text = "Update failed. A value was incorrect.";
    UpdtConfirm.Visible = true;
}

        //If there was not enough data added, this exception will be
thrown
catch (ArgumentOutOfRangeException)
{
    UpdtConfirm.Text = "Update failed. Not enough data
supplied.";
    UpdtConfirm.Visible = true;
}

        //Update the sources
for (int i = 0; i < GridView2.Rows.Count; i++)
{
    //Make sure that the row is not empty - would throw off
indexes
    if (GridView2.Rows[i].Cells[0].Text.ToString() != "&nbsp;")
    {
        //Delete Source information about variable
System.Data.SqlClient.SqlCommand srcDel = new
System.Data.SqlClient.SqlCommand("sourceDel", con);
        srcDel.Parameters.Add("@Info",
System.Data.SqlDbType.VarChar).Value =
GridView2.Rows[i].Cells[0].Text.ToString();
    }
}

```

```

        srcDel.CommandType =
System.Data.CommandType.StoredProcedure;

        //Put new Source data in
        System.Data.SqlClient.SqlCommand srcAdd = new
System.Data.SqlClient.SqlCommand("sourceChange", con);
        srcAdd.Parameters.Add("@Info",
System.Data.SqlDbType.VarChar).Value =
GridView2.Rows[i].Cells[0].Text.ToString();
        srcAdd.Parameters.Add("@Source",
System.Data.SqlDbType.VarChar).Value =
GridView2.Rows[i].Cells[1].Text.ToString();
        srcAdd.Parameters.Add("@Update",
System.Data.SqlDbType.Int).Value =
GridView2.Rows[i].Cells[2].Text.ToString();
        srcAdd.CommandType =
System.Data.CommandType.StoredProcedure;

        //Let the user know that the update was succesful.
UpdtConfirm.Text = "Update successful.";
UpdtConfirm.Visible = true;

        //Execute statements
con.Open();
srcDel.ExecuteNonQuery();
srcAdd.ExecuteNonQuery();
con.Close();
    }
}
}
//If they aren't updated a statistic, they're updating a source
else
{
    try
    {
        //Go through the rows this time. Sources pertain to statistics.
Statistics pertain to wards
        for (int i = 0; i < GridView1.Rows.Count; i++)
        {
            //Delete Source information about variable
            System.Data.SqlClient.SqlCommand srcDel = new
System.Data.SqlClient.SqlCommand("sourceDel", con);
            srcDel.Parameters.Add("@Info",
System.Data.SqlDbType.VarChar).Value =
GridView1.Rows[i].Cells[0].Text.ToString();
            srcDel.CommandType =
System.Data.CommandType.StoredProcedure;

            //Put new Source data in
            System.Data.SqlClient.SqlCommand srcAdd = new
System.Data.SqlClient.SqlCommand("sourceChange", con);
            srcAdd.Parameters.Add("@Info",
System.Data.SqlDbType.VarChar).Value =
GridView1.Rows[i].Cells[0].Text.ToString();

```

```

        srcAdd.Parameters.Add("@Source",
System.Data.SqlDbType.VarChar).Value =
GridView1.Rows[i].Cells[1].Text.ToString();
        srcAdd.Parameters.Add("@Update",
System.Data.SqlDbType.Int).Value =
GridView1.Rows[i].Cells[2].Text.ToString();
        srcAdd.CommandType =
System.Data.CommandType.StoredProcedure;

        //Let the user know that the update was succesful.
UpdtConfirm.Text = "Update successful.";
UpdtConfirm.Visible = true;

        //Execute statements
con.Open();
srcDel.ExecuteNonQuery();
srcAdd.ExecuteNonQuery();
con.Close();
    }
}
//Make sure everything was typed in right - if not, let the
user know
catch (FormatException)
{
    UpdtConfirm.Text = "Update failed. A value was incorrect.";
    UpdtConfirm.Visible = true;
}
}

}

protected void UpdtBtn_Click(object sender, EventArgs e)
{
}

//User has picked a statistic to delete
protected void DeleteBtn_Click(object sender, EventArgs e)
{
    Label6.Visible = false;
    DeleteBtn.Visible = false;
    Button2.Visible = false;
    //Dont delete if selectedindex = 0. If possible, this would mean
//a.) 'Please Select One...' is being deleted
//b.) People can hit enter by mistake and delete something
if (DropDelete.SelectedIndex != 0)
    {
        //Database connection
        System.Data.SqlClient.SqlConnection con = new
System.Data.SqlClient.SqlConnection("Data
Source=.\SQLEXPRESS;AttachDbFilename=|DataDirectory|\\Wards.mdf;Integrated
Security=True;User Instance=True");
    }
}

```

```

        //Create a command to run the stored delete command on the source
table with the correct parameter
        System.Data.SqlClient.SqlCommand srcDel = new
System.Data.SqlClient.SqlCommand("sourceDel", con);
        srcDel.Parameters.Add("@Info",
System.Data.SqlDbType.VarChar).Value = DropDelete.SelectedValue.ToString();
        srcDel.CommandType = System.Data.CommandType.StoredProcedure;

        //Command that would run stored delete on limits table
        System.Data.SqlClient.SqlCommand limDel = new
System.Data.SqlClient.SqlCommand("limitDelete", con);
        limDel.Parameters.Add("@Info",
System.Data.SqlDbType.VarChar).Value = DropDelete.SelectedValue.ToString();
        limDel.CommandType = System.Data.CommandType.StoredProcedure;

        //Command to run stored delete on WardsInformation (the one with
the statistics)
        System.Data.SqlClient.SqlCommand del = new
System.Data.SqlClient.SqlCommand("wardDelete", con);
        del.Parameters.Add("@Info", System.Data.SqlDbType.VarChar).Value
= DropDelete.SelectedValue.ToString();
        del.CommandType = System.Data.CommandType.StoredProcedure;

        //Execute commands (queries)
        con.Open();
        srcDel.ExecuteNonQuery();
        limDel.ExecuteNonQuery();
        del.ExecuteNonQuery();
        con.Close();

        //Let user know that the statistic was deleted
        DelConfirm.Text = DropDelete.SelectedValue.ToString() + " has
been deleted.";
        DelConfirm.Visible = true;

        //Rebind the dropdown so that the user doesnt try to delete the
same thing twice
        DropDelete.DataBind();
    }
    //If the index is on 0...
    else
    {
        //Politely remind them that they need to pick something to
delete.
        DelConfirm.Text = "Please select a statistic to delete.";
        DelConfirm.Visible = true;
    }
}

//User has specified location of statistics and needs to retrieve it.
protected void GetStatisticsBtn_Click(object sender, EventArgs e)
{
    //Display the button to confirm the data transaction
    //Display the label that goes with it

```

```

//Show table with data so that the user can check

Checker.Visible = true;
Confirm.Visible = true;
GridView1.Visible = true;

//Keep track of the fact that statistics are being updated
checkLbl.Text = "Stat";

//Connection to the database
System.Data.OleDb.OleDbConnection oledbConn;

//Database connection
//Firefox doesn't catch the C:/ in the path name, so add it if it's
not there
if (StatPath.PostedFile.FileName.ToString().StartsWith("C:"))
{
    oledbConn = new
System.Data.OleDb.OleDbConnection("Provider=Microsoft.Jet.OLEDB.4.0;Data
Source=" + StatPath.PostedFile.FileName.ToString() + ";Extended
Properties='Excel 8.0;IMEX=1'");
}
else
{
    oledbConn = new
System.Data.OleDb.OleDbConnection("Provider=Microsoft.Jet.OLEDB.4.0;Data
Source=C:/" + StatPath.PostedFile.FileName.ToString() + ";Extended
Properties='Excel 8.0;IMEX=1'");
}
try
{
    //Open Connection
oledbConn.Open();

    //Command to get information from oledbConn connection
System.Data.OleDb.OleDbCommand cmd = new
System.Data.OleDb.OleDbCommand("SELECT * FROM [Sheet1$]", oledbConn);
//Get all the data from the Excel sheet that is accessed by the
connection oledbConn
System.Data.OleDb.OleDbCommand cmd2 = new
System.Data.OleDb.OleDbCommand("SELECT * FROM [Sheet2$]", oledbConn);

    //Declare new data Adapter
System.Data.OleDb.OleDbDataAdapter oleda2 = new
System.Data.OleDb.OleDbDataAdapter();

    //set the data adapter command to the command specified earlier
oleda2.SelectCommand = cmd2;

    //Need a dataset to hold all this data for a second
System.Data.DataSet ds2 = new System.Data.DataSet();

    //Fill brand new dataset
oleda2.Fill(ds2);

```

```

        //GridView generate
        GridView2.DataSource = ds2;
        GridView2.DataBind();

        //DataAdapter declaration
        System.Data.OleDb.OleDbDataAdapter oleda = new
System.Data.OleDb.OleDbDataAdapter();

        //Sync. the data adapter and command
        oleda.SelectCommand = cmd;

        //Empty dataset
        System.Data.DataSet ds = new System.Data.DataSet();

        //Full dataset, with information from Excel sheet
        oleda.Fill(ds);

        //Bind gridview so that user can see.
        GridView1.DataSource = ds;
        GridView1.DataBind();
    }
    catch
    {
    }
    finally
    {
        //Close connection
        oledbConn.Close();
    }
}

//Pre-Delete button
protected void Button1_Click(object sender, EventArgs e)
{
    //Ask the user if the delete was on purpose
    Label6.Visible = true;
    Label6.Text = "Are you sure you want to delete " +
DropDelete.Selected.Value.ToString() + "?";

    //Allow the user to confirm or deny the delete (two buttons)
    DeleteBtn.Visible = true;
    Button2.Visible = true;
}

//Deny button
protected void Button2_Click(object sender, EventArgs e)
{
    //Simply clear away everything to start from scratch
    Label6.Visible = false;
    DeleteBtn.Visible = false;
}

```



```

    "<path id='ColliersWoods' d ='M 320 141 L 325 151 L 315 155 L 313 153
L 308 155 L 312 163 L 315 178 C 300 178 300 200 300 205 L 320 190 L 320 195 L
335 190 L 335 180 L 345 175 L 345 182 L 370 175 L 370 160 C 350 160 350 141
320 141 Z'",
    "<path id='CricketGreen' d ='M 348 300 L 354 293 L 350 288 L 357 275
L 338 270 L 292 230 L 300 215 L 320 215 L 320 218 L 335 218 L 340 228 L 352
220 L 367 226 L 365 238 L 367 245 L 375 244 L 375 240 L 380 236 L 390 240 L
383 265 L 385 270 L 450 300 L 430 310 L 420 303 L 387 314 L 380 320 L 365 320
L 360 312 Z'",
    "<path id='Dundonald' d ='M 160 205 L 157 198 L 174 187 C 185 193
210 165 230 155 C 230 160 235 160 250 167 L 257 187 C 253 192 225 194 205 197
L 208 205 C 200 208 180 210 160 205 Z'",
    "<path id='FiggesMarsh' d='M 397 216 L 394 213 L 397 210 L 394 207 L
397 204 L 388 194 L 393 189 L 384 178 L 382 182 L 375 171 L 370 167 L 370 175
L 375 203 L 370 208 L 370 211 L 360 219 L 357 215 L 352 220 L 367 226 L 365
238 L 367 245 L 375 244 L 375 240 L 380 236 L 390 240 L 400 245 L 402 243 L
405 245 L 407 238 L 405 230 L 415 225 L 410 220 L 403 224 L 398 217 L 393 222
L 397 216 Z'",
    "<path id='Graveney' d='M 420 180 L 397 216 L 394 213 L 397 210 L 394
207 L 397 204 L 388 194 L 393 189 L 384 178 L 382 182 L 375 171 L 370 167 L
370 160 L 375 150 L 397 160 L 402 158 L 428 174 L 420 180 Z '",
    "<path id='Hillside' d ='M 174 187 L 170 182 L 176 174 L 166 157 L
200 142 L 205 147 L 212 137 L 219 132 L 224 132 L 235 122 L 238 125 L 250 122
C 250 132 245 145 230 155 C 210 165 185 193 174 187 Z'",
    "<path id='LavenderFields' d='M 352 220 L 340 228 L 335 218 L 320 218
L 320 215 L 300 215 L 300 205 L 320 190 L 320 195 L 335 190 L 335 180 L 345
175 L 345 182 L 370 175 L 375 203 L 370 208 L 370 211 L 360 219 L 357 215
Z'",
    "<path id='Longthornton' d ='M 453 233 L 442 237 L 440 235 L 420 250
L 407 238 L 405 230 L 415 225 L 410 220 L 403 224 L 398 217 L 393 222 L 420
180 L 421 181 L 419 184 L 423 186 L 413 200 L 420 200 L 425 196 L 450 205 L
450 198 L 452 198 L 454 200 L 470 200 L 465 210 C 455 205 450 225 453 233 Z'
",
    "<path id='LowerMorden' d ='M 145 293 L 145 286 L 153 283 L 150 280 L
166 281 L 163 277 L 185 270 L 198 288 L 215 300 L 230 297 L 223 310 L 195
350 C 185 340 160 323 150 308 L 145 293 Z'",
    "<path id='MertonPark' d ='M 292 230 L 257 205 C 252 203 242 197
242 191 L 205 197 L 208 205 C 220 220 250 260 248 270 L 250 280 L 257 274 L
264 250 L 272 246 L 276 246 L 285 242 Z'",
    "<path id='PollardsHill' d ='M 450 300 L 385 270 L 383 265 L 390 240
L 400 245 L 402 243 L 405 245 L 407 238 L 420 250 L 440 235 L 442 237 L
453 233 L 455 235 L 480 260 L 482 260 L 465 280 L 467 282 L 450 300 Z' ",
    "<path id='Ravensbury' d ='M 280 318 C 282 298 295 250 276 266 L 265
253 L 267 250 L 264 250 L 272 246 L 276 246 L 285 242 L 292 230 L 338 270 L
357 275 L 350 288 L 354 293 L 348 300 L 332 292 L 290 325 Z'",
    "<path id='RaynesPark' d= 'M 110 175 L 125 182 L 125 175 L 155 167 L
155 162 L 166 157 L 176 174 L 170 182 L 174 187 L 157 198 L 160 205 C 155 205
142 210 145 225 L 125 215 L 120 215 Z'",
    "<path id='StHelier' d ='M 280 318 L 263 310 L 245 320 L 237 313 L
233 317 L 223 310 L 230 297 L 215 300 L 198 288 L 205 280 L 210 284 C 245 280
250 285 248 270 L 250 280 L 257 274 L 264 250 L 267 250 L 265 253 L 276 266 C
295 250 282 298 280 318 Z'",

```

```

    "<path id='Trinity' d='M 230 155 C 235 145 283 130 307 138 L 320
141 L 325 151 L 315 155 L 313 153 L 308 155 L 312 163 L 275 169 L 270 166 C
255 170 260 167 250 167 C 235 160 230 160 230 155 Z'",
    "<path id='Village' d='M 221 114 L 224 115 L 224 121 L 232 115 L 235
122 L 224 132 L 219 132 L 212 137 L 205 147 L 200 142 L 155 162 L 155 167 L
125 175 L 125 182 L 110 175 C 100 130 80 120 100 100 C 145 90 150 70 190 80 L
220 80 L 225 85 L 220 85 C 230 95 225 100 218 105 L 218 120 Z'",
    "<path id='WestBarnes' d='M 120 215 L 125 215 L 145 225 C 142 210
155 205 160 205 L 163 220 L 163 225 L 167 225 L 174 240 L 170 245 L 177 255 L
172 260 L 185 270 L 163 277 L 166 281 L 150 280 L 153 283 L 145 286 L 145 293
L 130 300 L 129 288 L 125 280 L 118 275 L 108 260 L 112 250 L 112 235 L 118
225 Z'",
    "<path id='WimbledonPark' d='M 230 155 C 245 145 250 132 250 122 L
238 125 L 235 122 L 232 115 L 224 121 L 224 115 L 221 114 L 218 120 L 218 105
C 225 100 230 95 220 85 L 225 85 L 220 80 L 275 60 C 290 90 275 115 290 95 C
310 90 295 132 307 138 C 283 130 235 145 230 155 Z'",
    "<circle id='AbbeyCircle' cx = '280' cy = '192' stroke = 'black'
stroke-width = '1'",
    "<circle id = 'CannonHillCircle' cx = '210' cy = '245' stroke =
'black' stroke-width = '1'",
    "<circle id = 'ColliersWoodCircle' cx = '333' cy = '160' stroke =
'black' stroke-width = '1'",
    "<circle id = 'CricketGreenCircle' cx = '372' cy = '275' stroke =
'black' stroke-width = '1'",
    "<circle id = 'DundonaldCircle' cx = '230' cy = '175' stroke =
'black' stroke-width = '1'",
    "<circle id = 'FiggesMarshCircle' cx = '380' cy = '225' stroke =
'black' stroke-width = '1'",
    "<circle id = 'GraveneyCircle' cx = '400' cy = '175' stroke = 'black'
stroke-width = '1'",
    "<circle id = 'HillsideCircle' cx = '190' cy = '165' stroke = 'black'
stroke-width = '1'",
    "<circle id = 'LavenderFieldsCircle' cx = '340' cy = '208' stroke =
'black' stroke-width = '1'",
    "<circle id = 'LongthorntonCircle' cx = '430' cy = '215' stroke =
'black' stroke-width = '1'",
    "<circle id = 'LowerMordenCircle' cx = '185' cy = '310' stroke =
'black' stroke-width = '1'",
    "<circle id = 'MertonParkCircle' cx = '245' cy = '225' stroke =
'black' stroke-width = '1'",
    "<circle id = 'PollardsHillCircle' cx = '440' cy = '265' stroke =
'black' stroke-width = '1'",
    "<circle id = 'RavensburyCircle' cx = '312' cy = '280' stroke =
'black' stroke-width = '1'",
    "<circle id = 'RaynesParkCircle' cx = '150' cy = '193' stroke =
'black' stroke-width = '1'",
    "<circle id = 'StHelierCircle' cx = '255' cy = '296' stroke = 'black'
stroke-width = '1'",
    "<circle id = 'TrinityCircle' cx = '280' cy = '153' stroke = 'black'
stroke-width = '1'",
    "<circle id = 'VillageCircle' cx = '150' cy = '125' stroke = 'black'
stroke-width = '1'",
    "<circle id = 'WestBarnesCircle' cx = '145' cy = '245' stroke =
'black' stroke-width = '1'",

```

```

        "<circle id = 'WimbledonCircle' cx = '260' cy = '95' stroke = 'black'
stroke-width = '1'"
        };

//Set up SVG settings
    Response.AddHeader("Content-Type", "image/svg+xml");
    Response.Write("<svg width='100%' height = '100%'
xmlns='http://www.w3.org/2000/svg'>");
    //Declare all the different classes (with different colors) to be used.
    Response.Write("<style type = 'text/css'> <![CDATA[path{stroke-
width:1; stroke:white; fill:lightgrey} path.grey{stroke-width:1;
stroke:white; fill:grey} path.white{stroke-width:1; stroke:white; fill:snow}
path.whitesmoke{stroke-width:1; stroke:white; fill:whitesmoke}
path.lightgrey{stroke-width:1; stroke:white; fill:lightgrey}
path.darkslategrey{stroke-width:1; stroke:white; fill:darkslategrey}");
    Response.Write("path.palegreen{stroke-width:1; stroke:black;
fill:palegreen} path.lime{stroke-width:1; stroke:black; fill:lime}
path.limegreen{stroke-width:1; stroke:black; fill:limegreen}
path.forestgreen{stroke-width:1; stroke:black; fill:forestgreen}
path.darkgreen{stroke-width:1; stroke:black; fill:darkgreen}");
    Response.Write("path.lightpink{stroke-width:1; stroke:black;
fill:lightpink} path.hotpink{stroke-width:1; stroke:black; fill:deeppink}
path.magenta{stroke-width:1; stroke:black; fill:magenta}
path.mediumorchid{stroke-width:1; stroke:black; fill:mediumorchid}
path.purple{stroke-width:1; stroke:black; fill:purple}");
    Response.Write("path.lightgrey{stroke-width:1; stroke:black;
fill:lightgrey} path.dimgrey{stroke-width:1; stroke:black; fill:dimgrey}
path.grey{stroke-width:1; stroke:black; fill:grey} path.darkgrey{stroke-
width:1; stroke:black; fill:darkgrey} path.darkslategrey{stroke-width:1;
stroke:black; fill:darkslategrey}");
    Response.Write("path.palevioletred{stroke-width:1; stroke:black;
fill:palevioletred} path.crimson{stroke-width:1; stroke:black; fill:crimson}
path.peachpuff{stroke-width:1; stroke:black; fill:peachpuff}
path.firebrick{stroke-width:1; stroke:black; fill:firebrick}
path.darkred{stroke-width:1; stroke:black; fill:darkred}");
    Response.Write("path.lavender{stroke-width:1; stroke:black;
fill:lavender} path.lightsteelblue{stroke-width:1; stroke:black;
fill:lightsteelblue} path.mediumslateblue{stroke-width:1; stroke:black;
fill:mediumslateblue} path.mediumblue{stroke-width:1; stroke:black;
fill:mediumblue} path.midnightblue{stroke-width:1; stroke:black;
fill:midnightblue}]]> </style>");

//Retrieve all of the statistical data from the inquiry passed from
WardProfile.aspx
    float[] stats = { float.Parse(Request["Abbey"]),
float.Parse(Request["CannonHill"]),
                    float.Parse(Request["ColliersWood"]),
float.Parse(Request["CricketGreen"]),
                    float.Parse(Request["Dundonald"]),
float.Parse(Request["FiggesMarsh"]),
                    float.Parse(Request["Graveney"]),
float.Parse(Request["Hillside"]),

```

```

        float.Parse(Request["LavenderFields"]),
float.Parse(Request["Longthornton"]),
        float.Parse(Request["LowerMorden"]),
float.Parse(Request["MertonPark"]),
        float.Parse(Request["PollardsHill"]),
float.Parse(Request["Ravensbury"]),
        float.Parse(Request["RaynesPark"]),
float.Parse(Request["StHelier"]),
        float.Parse(Request["Trinity"]),
float.Parse(Request["Village"]),
        float.Parse(Request["WestBarnes"]),
float.Parse(Request["Wimbledon"]));

    //The limits from WardProfile.aspx
    float[] limits = {float.Parse(Request["first"]),
float.Parse(Request["second"]),
        float.Parse(Request["third"]),
float.Parse(Request["fourth"])};

    //For each ward, compare their data to each limit, and color the ward
depending
    //on where the statistics lay.
    //The if statements correspond the the color drop down menu so that
the user can pick their
    //Own color scheme

if (Request["V1Color"] == "green")
{
    for (int i = 0; i <= 19; i++)
    {
        if (stats[i] < limits[0])
        {
            wardPaths[i] += " class = 'palegreen'/>";
        }
        else if (stats[i] < limits[1])
        {
            wardPaths[i] += " class = 'lime'/>";
        }
        else if (stats[i] < limits[2])
        {
            wardPaths[i] += " class = 'limegreen'/>";
        }
        else if (stats[i] < limits[3])
        {
            wardPaths[i] += " class = 'forestgreen'/>";
        }
        else if (stats[i] >= limits[3])
        {
            wardPaths[i] += " class = 'darkgreen'/>";
        }
    }
}

//Legend should also be the same color

```

```

        Response.Write("<path id = 'Green5' d = 'M 565 40 L 580 40 L 580
55 L 565 55 Z' class = 'darkgreen'/>");
        Response.Write("<path id = 'Green4' d = 'M 565 65 L 580 65 L 580
80 L 565 80 Z' class = 'forestgreen'/>");
        Response.Write("<path id = 'Green3' d = 'M 565 90 L 580 90 L 580
105 L 565 105 Z' class = 'limegreen'/>");
        Response.Write("<path id = 'Green2' d = 'M 565 115 L 580 115 L 580
130 L 565 130 Z' class = 'lime'/>");
        Response.Write("<path id = 'Green1' d = 'M 565 140 L 580 140 L 580
155 L 565 155 Z' class = 'palegreen'/>");
    }

//Same procedure for every color ...
    if (Request["V1Color"] == "purple")
    {
        for (int i = 0; i <= 19; i++)
        {
            if (stats[i] < limits[0])
            {
                wardPaths[i] += " class = 'lightpink'/>";
            }
            else if (stats[i] < limits[1])
            {
                wardPaths[i] += " class = 'magenta'/>";
            }
            else if (stats[i] < limits[2])
            {
                wardPaths[i] += " class = 'hotpink'/>";
            }
            else if (stats[i] < limits[3])
            {
                wardPaths[i] += " class = 'mediumorchid'/>";
            }
            else if (stats[i] >= limits[3])
            {
                wardPaths[i] += " class = 'purple'/>";
            }
        }
        Response.Write("<path id = 'Purple5' d = 'M 565 40 L 580 40 L 580
55 L 565 55 Z' class = 'purple'/>");
        Response.Write("<path id = 'Purple4' d = 'M 565 65 L 580 65 L 580
80 L 565 80 Z' class = 'mediumorchid'/>");
        Response.Write("<path id = 'Purple3' d = 'M 565 90 L 580 90 L 580
105 L 565 105 Z' class = 'hotpink'/>");
        Response.Write("<path id = 'Purple2' d = 'M 565 115 L 580 115 L
580 130 L 565 130 Z' class = 'magenta'/>");
        Response.Write("<path id = 'Purple1' d = 'M 565 140 L 580 140 L
580 155 L 565 155 Z' class = 'lightpink'/>");
    }

    if (Request["V1Color"] == "grey")
    {
        for (int i = 0; i <= 19; i++)
        {

```

```

        if (stats[i] < limits[0])
        {
            wardPaths[i] += " class = 'lightgrey' />";
        }
        else if (stats[i] < limits[1])
        {
            wardPaths[i] += " class = 'darkgrey' />";
        }
        else if (stats[i] < limits[2])
        {
            wardPaths[i] += " class = 'grey' />";
        }
        else if (stats[i] < limits[3])
        {
            wardPaths[i] += " class = 'dimgrey' />";
        }
        else if (stats[i] >= limits[3])
        {
            wardPaths[i] += " class = 'darkslategrey' />";
        }
    }

    Response.Write("<path id = 'Grey5' d = 'M 565 40 L 580 40 L 580
55 L 565 55 Z' class = 'darkslategrey' />");
    Response.Write("<path id = 'Grey4' d = 'M 565 65 L 580 65 L 580
80 L 565 80 Z' class = 'dimgrey' />");
    Response.Write("<path id = 'Grey3' d = 'M 565 90 L 580 90 L 580
105 L 565 105 Z' class = 'grey' />");
    Response.Write("<path id = 'Grey2' d = 'M 565 115 L 580 115 L 580
130 L 565 130 Z' class = 'darkgrey' />");
    Response.Write("<path id = 'Grey1' d = 'M 565 140 L 580 140 L 580
155 L 565 155 Z' class = 'lightgrey' />");

    }

    if (Request["V1Color"] == "blue")
    {
        for (int i = 0; i <= 19; i++)
        {
            if (stats[i] < limits[0])
            {
                wardPaths[i] += " class = 'lavender' />";
            }
            else if (stats[i] < limits[1])
            {
                wardPaths[i] += " class = 'lightsteelblue' />";
            }
            else if (stats[i] < limits[2])
            {
                wardPaths[i] += " class = 'mediumslateblue' />";
            }
            else if (stats[i] < limits[3])
            {
                wardPaths[i] += " class = 'mediumblue' />";
            }
        }
    }

```

```

    }
    else if (stats[i] >= limits[3])
    {
        wardPaths[i] += " class = 'midnightblue'/>";
    }
}

Response.Write("<path id = 'Blue5' d = 'M 565 40 L 580 40 L 580
55 L 565 55 Z' class = 'midnightblue'/>");
Response.Write("<path id = 'Blue4' d = 'M 565 65 L 580 65 L 580
80 L 565 80 Z' class = 'mediumblue'/>");
Response.Write("<path id = 'Blue3' d = 'M 565 90 L 580 90 L 580
105 L 565 105 Z' class = 'mediumslateblue'/>");
Response.Write("<path id = 'Blue2' d = 'M 565 115 L 580 115 L 580
130 L 565 130 Z' class = 'lightsteelblue'/>");
Response.Write("<path id = 'Blue1' d = 'M 565 140 L 580 140 L 580
155 L 565 155 Z' class = 'lavender'/>");
}

if (Request["V1Color"] == "red")
{
    for (int i = 0; i <= 19; i++)
    {
        if (stats[i] < limits[0])
        {
            wardPaths[i] += " class = 'peachpuff'/>";
        }
        else if (stats[i] < limits[1])
        {
            wardPaths[i] += " class = 'palevioletred'/>";
        }
        else if (stats[i] < limits[2])
        {
            wardPaths[i] += " class = 'crimson'/>";
        }
        else if (stats[i] < limits[3])
        {
            wardPaths[i] += " class = 'firebrick'/>";
        }
        else if (stats[i] >= limits[3])
        {
            wardPaths[i] += " class = 'darkred'/>";
        }
    }
    Response.Write("<path id = 'Red5' d = 'M 565 40 L 580 40 L 580 55
L 565 55 Z' class = 'darkred'/>");
    Response.Write("<path id = 'Red4' d = 'M 565 65 L 580 65 L 580 80
L 565 80 Z' class = 'firebrick'/>");
    Response.Write("<path id = 'Red3' d = 'M 565 90 L 580 90 L 580
105 L 565 105 Z' class = 'crimson'/>");
    Response.Write("<path id = 'Red2' d = 'M 565 115 L 580 115 L 580
130 L 565 130 Z' class = 'palevioletred'/>");
    Response.Write("<path id = 'Red1' d = 'M 565 140 L 580 140 L 580
155 L 565 155 Z' class = 'peachpuff'/>");
}

```

```

    }

    //Draw the newly colored wards
    for (int i = 0; i < 20; i++)
    {
        Response.Write(wardPaths[i].ToString());
    }

    //Draw watermark
    //Response.Write("<linearGradient id = 'Transparent' gradientUnits =
'userSpaceOnUse' x1 = '0' y1='0' x2='250' y2='100'/>");
    //Response.Write("<mask maskContentUnits = 'userSpaceOnUse' id =
'Mask'>");
    Response.Write("<g opacity = '.05'>");
    Response.Write("<image x='0' y = '0' width = '200' height = '125'
xlink:href='merton_logo_2_black.jpg' />");
    Response.Write("<image x='200' y = '0' width = '200' height = '125'
xlink:href='merton_logo_2_black.jpg' />");
    Response.Write("<image x='400' y = '0' width = '200' height = '125'
xlink:href='merton_logo_2_black.jpg' />");
    Response.Write("<image x='0' y = '125' width = '200' height = '125'
xlink:href='merton_logo_2_black.jpg' />");
    Response.Write("<image x='200' y = '125' width = '200' height = '125'
xlink:href='merton_logo_2_black.jpg' />");
    Response.Write("<image x='400' y = '125' width = '200' height = '125'
xlink:href='merton_logo_2_black.jpg' />");
    Response.Write("<image x='0' y = '250' width = '200' height = '125'
xlink:href='merton_logo_2_black.jpg' />");
    Response.Write("<image x='200' y = '250' width = '200' height = '125'
xlink:href='merton_logo_2_black.jpg' />");
    Response.Write("<image x='400' y = '250' width = '200' height = '125'
xlink:href='merton_logo_2_black.jpg' />");
    Response.Write("</g>");

    //Check to see if there is data available for the second variable, as
determined
    // by the pseudo-boolean 'problem' that was passed by WardProfile.
    if (float.Parse(Request["problem"]) != 1)
    {
        //Retrieve statistical data to be transferred into circle color
        float[] circleStat = { float.Parse(Request["AbbeyCircle"]),
float.Parse(Request["CannonHillCircle"]),
float.Parse(Request["ColliersWoodCircle"]),
float.Parse(Request["CricketGreenCircle"]),
float.Parse(Request["DundonaldCircle"]),
float.Parse(Request["FiggesMarshCircle"]),
float.Parse(Request["GraveneyCircle"]),
float.Parse(Request["HillsideCircle"]),
float.Parse(Request["LavenderFieldsCircle"]),
float.Parse(Request["LongthorntonCircle"]),
float.Parse(Request["LowerMordenCircle"]),
float.Parse(Request["MertonParkCircle"]),
float.Parse(Request["PollardsHillCircle"]),
float.Parse(Request["RavensburyCircle"]),

```

```

        float.Parse(Request["RaynesParkCircle"]),
float.Parse(Request["StHelierCircle"]),
        float.Parse(Request["TrinityCircle"]),
float.Parse(Request["VillageCircle"]),
        float.Parse(Request["WestBarnesCircle"]),
float.Parse(Request["WimbledonCircle"])};

//Get specified ranges for the circle color statistics there.
float[] limits2 = {float.Parse(Request["circle1"]),
float.Parse(Request["circle2"]),
        float.Parse(Request["circle3"]),
float.Parse(Request["circle4"])};

//Go through each ward and compare the statistic to the ranges.
//Add a fill to the circle corresponding to the ward depending on
what range the statistic falls in.
//Fill color is based off of variable based off of drop down on
WardProfile.aspx
if (Request["V2Color"] == "purple")
{
    for (int j = 0; j < 20; j++)
    {
        if (circleStat[j] < limits2[0])
        {
            wardPaths[j + 20] += " fill = 'lightpink'";
        }
        else if (circleStat[j] < limits2[1])
        {
            wardPaths[j + 20] += " fill = 'magenta'";
        }
        else if (circleStat[j] < limits2[2])
        {
            wardPaths[j + 20] += " fill = 'deeppink'";
        }
        else if (circleStat[j] < limits2[3])
        {
            wardPaths[j + 20] += " fill = 'mediumorchid'";
        }
        else if (circleStat[j] >= limits2[3])
        {
            wardPaths[j + 20] += " fill = 'purple'";
        }
    }
    //Legend should correspond
    Response.Write("<path id = 'cShade1' d = 'M 565 195 L 580 195 L
580 210 L 565 210 Z' class = 'purple'/>");
    Response.Write("<path id = 'cShade2' d = 'M 565 220 L 580 220 L
580 235 L 565 235 Z' class = 'mediumorchid'/>");
    Response.Write("<path id = 'cShade3' d = 'M 565 245 L 580 245 L
580 260 L 565 260 Z' class = 'deeppink'/>");
    Response.Write("<path id = 'cShade4' d = 'M 565 270 L 580 270 L
580 285 L 565 285 Z' class = 'magenta'/>");
    Response.Write("<path id = 'cShade5' d = 'M 565 295 L 580 295 L
580 310 L 565 310 Z' class = 'lightpink'/>");

```

```

}

//Same for the rest of the colors...
if (Request["V2Color"] == "grey")
{
    for (int j = 0; j < 20; j++)
    {
        if (circleStat[j] < limits2[0])
        {
            wardPaths[j + 20] += " fill = 'lightgrey'";
        }
        else if (circleStat[j] < limits2[1])
        {
            wardPaths[j + 20] += " fill = 'darkgrey'";
        }
        else if (circleStat[j] < limits2[2])
        {
            wardPaths[j + 20] += " fill = 'grey'";
        }
        else if (circleStat[j] < limits2[3])
        {
            wardPaths[j + 20] += " fill = 'dimgrey'";
        }
        else if (circleStat[j] >= limits2[3])
        {
            wardPaths[j + 20] += " fill = 'darkslategrey'";
        }
    }
    Response.Write("<path id = 'cShade1' d = 'M 565 195 L 580 195 L
580 210 L 565 210 Z' class = 'darkslategrey'/>");
    Response.Write("<path id = 'cShade2' d = 'M 565 220 L 580 220 L
580 235 L 565 235 Z' class = 'dimgrey'/>");
    Response.Write("<path id = 'cShade3' d = 'M 565 245 L 580 245 L
580 260 L 565 260 Z' class = 'grey'/>");
    Response.Write("<path id = 'cShade4' d = 'M 565 270 L 580 270 L
580 285 L 565 285 Z' class = 'darkgrey'/>");
    Response.Write("<path id = 'cShade5' d = 'M 565 295 L 580 295 L
580 310 L 565 310 Z' class = 'lightgrey'/>");
}

if (Request["V2Color"] == "green")
{
    for (int j = 0; j < 20; j++)
    {
        if (circleStat[j] < limits2[0])
        {
            wardPaths[j + 20] += " fill = 'palegreen'";
        }
        else if (circleStat[j] < limits2[1])
        {
            wardPaths[j + 20] += " fill = 'lime'";
        }
        else if (circleStat[j] < limits2[2])
        {

```

```

        wardPaths[j + 20] += " fill = 'limegreen'";
    }
    else if (circleStat[j] < limits2[3])
    {
        wardPaths[j + 20] += " fill = 'forestgreen'";
    }
    else if (circleStat[j] >= limits2[3])
    {
        wardPaths[j + 20] += " fill = 'darkgreen'";
    }
}
Response.Write("<path id = 'cShade1' d = 'M 565 195 L 580 195 L
580 210 L 565 210 Z' class = 'darkgreen'/>");
Response.Write("<path id = 'cShade2' d = 'M 565 220 L 580 220 L
580 235 L 565 235 Z' class = 'forestgreen'/>");
Response.Write("<path id = 'cShade3' d = 'M 565 245 L 580 245 L
580 260 L 565 260 Z' class = 'limegreen'/>");
Response.Write("<path id = 'cShade4' d = 'M 565 270 L 580 270 L
580 285 L 565 285 Z' class = 'lime'/>");
Response.Write("<path id = 'cShade5' d = 'M 565 295 L 580 295 L
580 310 L 565 310 Z' class = 'palegreen'/>");
}

if (Request["V2Color"] == "red")
{
    for (int j = 0; j < 20; j++)
    {
        if (circleStat[j] < limits2[0])
        {
            wardPaths[j + 20] += " fill = 'peachpuff'";
        }
        else if (circleStat[j] < limits2[1])
        {
            wardPaths[j + 20] += " fill = 'palevioletred'";
        }
        else if (circleStat[j] < limits2[2])
        {
            wardPaths[j + 20] += " fill = 'crimson'";
        }
        else if (circleStat[j] < limits2[3])
        {
            wardPaths[j + 20] += " fill = 'firebrick'";
        }
        else if (circleStat[j] >= limits2[3])
        {
            wardPaths[j + 20] += " fill = 'darkred'";
        }
    }
    Response.Write("<path id = 'cShade1' d = 'M 565 195 L 580 195 L
580 210 L 565 210 Z' class = 'darkred'/>");
    Response.Write("<path id = 'cShade2' d = 'M 565 220 L 580 220 L
580 235 L 565 235 Z' class = 'firebrick'/>");
    Response.Write("<path id = 'cShade3' d = 'M 565 245 L 580 245 L
580 260 L 565 260 Z' class = 'crimson'/>");
}

```

```

        Response.Write("<path id = 'cShade4' d = 'M 565 270 L 580 270 L
580 285 L 565 285 Z' class = 'palevioletred'/>");
        Response.Write("<path id = 'cShade5' d = 'M 565 295 L 580 295 L
580 310 L 565 310 Z' class = 'peachpuff'/>");
    }

```

```

if (Request["V2Color"] == "blue")
{
    for (int j = 0; j < 20; j++)
    {
        if (circleStat[j] < limits2[0])
        {
            wardPaths[j + 20] += " fill = 'lavender'";
        }
        else if (circleStat[j] < limits2[1])
        {
            wardPaths[j + 20] += " fill = 'lightsteelblue'";
        }
        else if (circleStat[j] < limits2[2])
        {
            wardPaths[j + 20] += " fill = 'mediumslateblue'";
        }
        else if (circleStat[j] < limits2[3])
        {
            wardPaths[j + 20] += " fill = 'mediumblue'";
        }
        else if (circleStat[j] >= limits2[3])
        {
            wardPaths[j + 20] += " fill = 'midnightblue'";
        }
    }
    Response.Write("<path id = 'cShade1' d = 'M 565 195 L 580 195 L
580 210 L 565 210 Z' class = 'midnightblue'/>");
    Response.Write("<path id = 'cShade2' d = 'M 565 220 L 580 220 L
580 235 L 565 235 Z' class = 'mediumblue'/>");
    Response.Write("<path id = 'cShade3' d = 'M 565 245 L 580 245 L
580 260 L 565 260 Z' class = 'mediumslateblue'/>");
    Response.Write("<path id = 'cShade4' d = 'M 565 270 L 580 270 L
580 285 L 565 285 Z' class = 'lightsteelblue'/>");
    Response.Write("<path id = 'cShade5' d = 'M 565 295 L 580 295 L
580 310 L 565 310 Z' class = 'lavender'/>");
}

```

```

//If there is no data for circle size, set all the circles to the
same size
// and draw them.
if (float.Parse(Request["radBad"]) == 1)
{
    for (int count = 0; count < 20; count++)
    {
        wardPaths[count + 20] += " r = '10'/>";
        Response.Write(wardPaths[count + 20].ToString());
    }
}

```

```

    }
    //If there is data for a third variable though...
    else
    {
        //retrieve statistics to be translated to circle size
        float[] rad = {float.Parse(Request["AbbeyRad"]),
float.Parse(Request["CannonHillRad"]),
        float.Parse(Request["ColliersWoodRad"]),
float.Parse(Request["CricketGreenRad"]),
        float.Parse(Request["DundonaldRad"]),
float.Parse(Request["FiggesMarshRad"]),
        float.Parse(Request["GraveneyRad"]),
float.Parse(Request["HillsideRad"]),
        float.Parse(Request["LavenderFieldsRad"]),
float.Parse(Request["LongthorntonRad"]),
        float.Parse(Request["LowerMordenRad"]),
float.Parse(Request["MertonParkRad"]),
        float.Parse(Request["PollardsHillRad"]),
float.Parse(Request["RavensburyRad"]),
        float.Parse(Request["RaynesParkRad"]),
float.Parse(Request["StHelierRad"]),
        float.Parse(Request["TrinityRad"]),
float.Parse(Request["VillageRad"]),
        float.Parse(Request["WestBarnesRad"]),
float.Parse(Request["WimbledonRad"])
        };

        //gather those ranges.
        float[] radLimit = {float.Parse(Request["firstRad"]),
float.Parse(Request["secondRad"]),
        float.Parse(Request["thirdRad"]),
float.Parse(Request["fourthRad"])};

        //Compare each ward to the ranges, and specify the radius
        //of the corresponding circle accordingly, if you will.
        for (int k = 0; k < 20; k++)
        {
            if (rad[k] < radLimit[0])
            {
                wardPaths[k + 20] += " r = '5' />";
            }
            else if (rad[k] < radLimit[1])
            {
                wardPaths[k + 20] += " r = '7.5' />";
            }
            else if (rad[k] < radLimit[2])
            {
                wardPaths[k + 20] += " r = '10' />";
            }
            else if (rad[k] < radLimit[3])
            {
                wardPaths[k + 20] += " r = '12.5' />";
            }
            else if (rad[k] >= radLimit[3])

```

```

        {
            wardPaths[k + 20] += " r = '15' />";
        }

        //draw circles
        Response.Write(wardPaths[k + 20].ToString());
    }

}

//Draw the legends if they are needed
//First Legend is always needed
Response.Write("<text x = '565' y = '25' font-size = '20' font-family =
'Arial'> Ward Shade: </text>");
Response.Write("<text x = '725' y = '25' font-size = '17' font-family =
'Arial'>" + Request["Var1Select"].ToString() + "</text>");
Response.Write("<text x = '725' y = '50' font-size = '14' font-family =
'Arial'>Last updated: " + Request["Var1LastUpdate"].ToString() + "</text>");
Response.Write("<text x = '725' y = '75' font-size = '14' font-family =
'Arial'>Source: " + Request["Var1Source"].ToString() + "</text>");

Response.Write("<text x = '590' y = '150' font-family = 'Arial'> Less
than " + Math.Round(double.Parse(Request["first"]), 1).ToString() + "
</text>");
Response.Write("<text x = '590' y = '125' font-family = 'Arial'> Less
than " + Math.Round(double.Parse(Request["second"]), 1).ToString() + "
</text>");
Response.Write("<text x = '590' y = '100' font-family = 'Arial'> Less
than " + Math.Round(double.Parse(Request["third"]), 1).ToString() + "
</text>");
Response.Write("<text x = '590' y = '75' font-family = 'Arial'> Less than
" + Math.Round(double.Parse(Request["fourth"]), 1).ToString() + " </text>");
Response.Write("<text x = '590' y = '50' font-family = 'Arial'> Greater
than " + Math.Round(double.Parse(Request["fourth"]), 1).ToString() + "
</text>");

if (float.Parse(Request["problem"]) != 1)
{
    //Second Legend - Only if there's data there
    Response.Write("<text x = '565' y = '180' font-size = '20' font-family
= 'Arial'> Circle Shade: </text>");
    Response.Write("<text x = '725' y = '180' font-size = '17' font-
family = 'Arial'>" + Request["Var2Select"].ToString() + "</text>");
    Response.Write("<text x = '725' y = '205' font-size = '14' font-
family = 'Arial'>Last updated: " + Request["Var2LastUpdate"].ToString() +
"</text>");
    Response.Write("<text x = '725' y = '230' font-size = '14' font-
family = 'Arial'>Source: " + Request["Var2Source"].ToString() + "</text>");

    Response.Write("<text x = '590' y = '305' font-family = 'Arial'> Less
than " + Math.Round(double.Parse(Request["circle1"]), 1).ToString() + "
</text>");
}

```

```

        Response.Write("<text x = '590' y = '280' font-family = 'Arial'> Less
than " + Math.Round(double.Parse(Request["circle2"]), 1).ToString() + "
</text>");
        Response.Write("<text x = '590' y = '255' font-family = 'Arial'> Less
than " + Math.Round(double.Parse(Request["circle3"]), 1).ToString() + "
</text>");
        Response.Write("<text x = '590' y = '230' font-family = 'Arial'> Less
than " + Math.Round(double.Parse(Request["circle4"]), 1).ToString() + "
</text>");
        Response.Write("<text x = '590' y = '205' font-family = 'Arial'>
Greater than " + Math.Round(double.Parse(Request["circle4"]), 1).ToString() +
" </text>");
    }

    //Third Legend - Again, only if needed (get it?)
    if (float.Parse(Request["radBad"]) != 1)
    {
        Response.Write("<text x = '565' y = '335' font-size = '20' font-family
= 'Arial'> Circle Size: </text>");
        Response.Write("<text x = '725' y = '335' font-size = '17' font-
family = 'Arial'>" + Request["Var3Select"].ToString() + "</text>");
        Response.Write("<text x = '725' y = '360' font-size = '14' font-
family = 'Arial'>Last updated: " + Request["Var1LastUpdate"].ToString() +
"</text>");
        Response.Write("<text x = '725' y = '385' font-size = '14' font-
family = 'Arial'>Source: " + Request["Var1Source"].ToString() + "</text>");

        Response.Write("<circle id = 'rad5' cx = '575' cy = '365' stroke =
'black' stroke-width = '1' fill = 'white' r = '15' />");
        Response.Write("<circle id = 'rad4' cx = '575' cy = '397' stroke =
'black' stroke-width = '1' fill = 'white' r = '12.5' />");
        Response.Write("<circle id = 'rad3' cx = '575' cy = '425' stroke =
'black' stroke-width = '1' fill = 'white' r = '10' />");
        Response.Write("<circle id = 'rad2' cx = '575' cy = '447' stroke =
'black' stroke-width = '1' fill = 'white' r = '7.5' />");
        Response.Write("<circle id = 'rad1' cx = '575' cy = '465' stroke =
'black' stroke-width = '1' fill = 'white' r = '5' />");

        Response.Write("<text x = '610' y = '465' font-family = 'Arial'> Less
than " + Math.Round(double.Parse(Request["firstRad"]), 1).ToString() + "
</text>");
        Response.Write("<text x = '610' y = '447' font-family = 'Arial'> Less
than " + Math.Round(double.Parse(Request["secondRad"]), 1).ToString() + "
</text>");
        Response.Write("<text x = '610' y = '425' font-family = 'Arial'> Less
than " + Math.Round(double.Parse(Request["thirdRad"]), 1).ToString() + "
</text>");
        Response.Write("<text x = '610' y = '397' font-family = 'Arial'> Less
than " + Math.Round(double.Parse(Request["fourthRad"]), 1).ToString() + "
</text>");
        Response.Write("<text x = '610' y = '365' font-family = 'Arial'>
Greater than " + Math.Round(double.Parse(Request["fourthRad"]), 1).ToString()
+ " </text>");
    }

```

```

//If the ward toggle has been set to on (wards == 1) then draw the test
for each ward
  if (float.Parse(Request["wards"]) == 1)
  {
    Response.Write("<text x='120' y='110' font-family='Arial'>Village
</text>");
    Response.Write("<text x='240' y='115' font-family='Arial'>Wimbledon
Park</text>");
    Response.Write("<text x='97' y='192' font-family='Arial'>Raynes
Park</text>");
    Response.Write("<text x='180' y='165' font-
family='Arial'>Hillside</text>");
    Response.Write("<text x='250' y='160' font-
family='Arial'>Trinity</text>");
    Response.Write("<text x='190' y='190' font-
family='Arial'>Dundonald</text>");
    Response.Write("<text x='260' y='195' font-
family='Arial'>Abbey</text>");
    Response.Write("<text x='100' y='250' font-family='Arial'>West
Barnes</text>");
    Response.Write("<text x='220' y='230' font-family='Arial'>Merton
Park</text>");
    Response.Write("<text x='180' y='265' font-family='Arial'>Cannon
Hill</text>");
    Response.Write("<text x='150' y='320' font-family='Arial'>Lower
Morden</text>");
    Response.Write("<text x='230' y='300' font-family='Arial'>St
Helier</text>");
    Response.Write("<text x='287' y='285' font-
family='Arial'>Ravensbury</text>");
    Response.Write("<text x='407' y='270' font-family='Arial'>Pollards
Hill</text>");
    Response.Write("<text x='360' y='300' font-family='Arial'>Cricket
Green</text>");
    Response.Write("<text x='415' y='220' font-
family='Arial'>Longthornton</text>");
    Response.Write("<text x='400' y='180' font-
family='Arial'>Graveney</text>");
    Response.Write("<text x='350' y='235' font-family='Arial'>Figge's
Marsh</text>");
    Response.Write("<text x='315' y='165' font-family='Arial'>Collier's
Wood</text>");
    Response.Write("<text x='305' y='210' font-family='Arial'>Lavender
Fields</text>");
  }

//Throw in the copyright text - don't want to get sued. And close up svg.
Response.Write("<text x='100' y='410' font-family='Arial'> Crown
Copyright. All rights reserved. London Borough of Merton. 10019259.
2008</text>");
Response.Write("</svg>");
%>

```

GraphDraw.aspx

```
<%@Page Language ="C#"%>

<%

    //Set up SVG settings
    Response.AddHeader("Content-Type", "image/svg+xml");
    Response.Write("<svg width='100%' height ='100%'
xmlns='http://www.w3.org/2000/svg'>");
    //Declare all the different classes (with different colors) to be used.
    Response.Write("<style type ='text/css'> <![CDATA[path{stroke-width:1;
stroke:white; fill:lightgrey} path.grey{stroke-width:1; stroke:white;
fill:grey} path.white{stroke-width:1; stroke:white; fill:snow}
path.whitesmoke{stroke-width:1; stroke:white; fill:whitesmoke}
path.lightgrey{stroke-width:1; stroke:white; fill:lightgrey}
path.darkslategrey{stroke-width:1; stroke:white; fill:darkslategrey}");
    Response.Write("rect.blue{fill:blue; stroke-width:1; stroke:black}
rect.green{fill:green; stroke-width:1; stroke:black} rect.grey{fill:grey;
stroke-width:1; stroke:black} rect.purple{fill:purple; stroke-width:1;
stroke:black} rect.red{fill:red; stroke-width:1; stroke:black}]]> </style>");

    string[] WardBars =
    {
        "<rect id = 'AbbeyRect' height = '15' y = '20'",
        "<rect id = 'CannonHillRect' height = '15' y = '45'",
        "<rect id = 'ColliersWoodRect' height = '15' y = '70'",
        "<rect id = 'CricketGreenRect' height = '15' y = '95'",
        "<rect id = 'DundonaldRect' height = '15' y = '120'",
        "<rect id = 'FiggesMarshRect' height = '15' y = '145'",
        "<rect id = 'GraveneyRect' height = '15' y = '170'",
        "<rect id = 'HillsideRect' height = '15' y = '195'",
        "<rect id = 'LavenderFieldsRect' height = '15' y = '220'",
        "<rect id = 'LongthorntonRect' height = '15' y = '245'",
        "<rect id = 'LowerMordenRect' height = '15' y = '270'",
        "<rect id = 'MertonParkRect' height = '15' y = '295'",
        "<rect id = 'PollardsHillRect' height = '15' y = '320'",
        "<rect id = 'RavensburyRect' height = '15' y = '345'",
        "<rect id = 'RaynesParkRect' height = '15' y = '370'",
        "<rect id = 'StHelierRect' height = '15' y = '395'",
        "<rect id = 'TrinityRect' height = '15' y = '420'",
        "<rect id = 'VillageRect' height = '15' y = '445'",
        "<rect id = 'WestBarnesRect' height = '15' y = '470'",
        "<rect id = 'WimbledonRect' height = '15' y = '495'"
    };

    string[] WardLabels =
    {
        "<text id = 'AbbeyLbl' x = '0' y = '31' font-family = 'Arial'
font-size = '14'>Abbey</text>",
        "<text id = 'CannonHillLbl' x = '0' y = '56' font-family =
'Arial' font-size = '14'>Cannon Hill</text>",
        "<text id = 'ColliersWoodLbl' x = '0' y = '81' font-family =
'Arial' font-size = '14'>Colliers Wood</text>",
```

```

        "<text id = 'CricketGreenLbl' x = '0' y = '106' font-family =
'Arial' font-size = '14'>Cricket Green</text>",
        "<text id = 'DundonaldLbl' x = '0' y = '131' font-family =
'Arial' font-size = '14'>Dundonald</text>",
        "<text id = 'FiggesMarshLbl' x = '0' y = '156' font-family =
'Arial' font-size = '14'>Figges Marsh</text>",
        "<text id = 'GraveneyLbl' x = '0' y = '181' font-family = 'Arial'
font-size = '14'>Graveney</text>",
        "<text id = 'HillsideLbl' x = '0' y = '206' font-family = 'Arial'
font-size = '14'>Hillside</text>",
        "<text id = 'LavenderFieldsLbl' x = '0' y = '231' font-family =
'Arial' font-size = '14'>Lavender Fields</text>",
        "<text id = 'LongthorntonLbl' x = '0' y = '256' font-family =
'Arial' font-size = '14'>Longthornton</text>",
        "<text id = 'LowerMordenLbl' x = '0' y = '281' font-family =
'Arial' font-size = '14'>Lower Morden</text>",
        "<text id = 'MertonParkLbl' x = '0' y = '306' font-family =
'Arial' font-size = '14'>Merton Park</text>",
        "<text id = 'PollardsHillLbl' x = '0' y = '331' font-family =
'Arial' font-size = '14'>Pollards Hill</text>",
        "<text id = 'RavensburyLbl' x = '0' y = '356' font-family =
'Arial' font-size = '14'>Ravensbury</text>",
        "<text id = 'RaynesParkLbl' x = '0' y = '381' font-family =
'Arial' font-size = '14'>Raynes Park</text>",
        "<text id = 'StHelierLbl' x = '0' y = '406' font-family = 'Arial'
font-size = '14'>St Helier</text>",
        "<text id = 'TrinityLbl' x = '0' y = '431' font-family = 'Arial'
font-size = '14'>Trinity</text>",
        "<text id = 'VillageLbl' x = '0' y = '456' font-family = 'Arial'
font-size = '14'>Village</text>",
        "<text id = 'WestBarnesLbl' x = '0' y = '481' font-family =
'Arial' font-size = '14'>West Barnes</text>",
        "<text id = 'WimbledonLbl' x = '0' y = '506' font-family =
'Arial' font-size = '14'>Wimbledon</text>"
};

```

```

string[] statDraw =
{
    "<text y = '33' font-family = 'Arial' font-size = '14'",
    "<text y = '58' font-family = 'Arial' font-size = '14'",
    "<text y = '83' font-family = 'Arial' font-size = '14'",
    "<text y = '108' font-family = 'Arial' font-size = '14'",
    "<text y = '133' font-family = 'Arial' font-size = '14'",
    "<text y = '158' font-family = 'Arial' font-size = '14'",
    "<text y = '183' font-family = 'Arial' font-size = '14'",
    "<text y = '208' font-family = 'Arial' font-size = '14'",
    "<text y = '233' font-family = 'Arial' font-size = '14'",
    "<text y = '258' font-family = 'Arial' font-size = '14'",
    "<text y = '283' font-family = 'Arial' font-size = '14'",
    "<text y = '308' font-family = 'Arial' font-size = '14'",
    "<text y = '333' font-family = 'Arial' font-size = '14'",
    "<text y = '358' font-family = 'Arial' font-size = '14'",
    "<text y = '383' font-family = 'Arial' font-size = '14'",
    "<text y = '408' font-family = 'Arial' font-size = '14'",

```

```

        "<text y = '433' font-family = 'Arial' font-size = '14'",
        "<text y = '458' font-family = 'Arial' font-size = '14'",
        "<text y = '483' font-family = 'Arial' font-size = '14'",
        "<text y = '508' font-family = 'Arial' font-size = '14'"
    };

    float[] width =
    {
        200 *
        (float.Parse(Request["Abbey"])/float.Parse(Request["max"])),
        200 *
        (float.Parse(Request["CannonHill"])/float.Parse(Request["max"])),
        200 *
        (float.Parse(Request["ColliersWood"])/float.Parse(Request["max"])),
        200 *
        (float.Parse(Request["CricketGreen"])/float.Parse(Request["max"])),
        200 *
        (float.Parse(Request["Dundonald"])/float.Parse(Request["max"])),
        200 *
        (float.Parse(Request["FiggesMarsh"])/float.Parse(Request["max"])),
        200 *
        (float.Parse(Request["Graveney"])/float.Parse(Request["max"])),
        200 *
        (float.Parse(Request["Hillside"])/float.Parse(Request["max"])),
        200 *
        (float.Parse(Request["LavenderFields"])/float.Parse(Request["max"])),
        200 *
        (float.Parse(Request["Longthornton"])/float.Parse(Request["max"])),
        200 *
        (float.Parse(Request["LowerMorden"])/float.Parse(Request["max"])),
        200 *
        (float.Parse(Request["MertonPark"])/float.Parse(Request["max"])),
        200 *
        (float.Parse(Request["PollardsHill"])/float.Parse(Request["max"])),
        200 *
        (float.Parse(Request["Ravensbury"])/float.Parse(Request["max"])),
        200 *
        (float.Parse(Request["RaynesPark"])/float.Parse(Request["max"])),
        200 *
        (float.Parse(Request["StHelier"])/float.Parse(Request["max"])),
        200 *
        (float.Parse(Request["Trinity"])/float.Parse(Request["max"])),
        200 *
        (float.Parse(Request["Village"])/float.Parse(Request["max"])),
        200 *
        (float.Parse(Request["WestBarnes"])/float.Parse(Request["max"])),
        200 *
        (float.Parse(Request["Wimbledon"])/float.Parse(Request["max"]))
    };

    for (int i = 0; i < 20; i++)
    {
        if (width[i] < 0)
        {

```

```

        WardBars[i] += " width='0'";
    }
    else
    {
        WardBars[i] += " width='" + width[i].ToString() + "'";
    }
}

if (Request["V1Color"] == "green")
{
    for (int i = 0; i < 20; i++)
    {
        WardBars[i] += " class ='green' ";
    }
}

if (Request["V1Color"] == "blue")
{
    for (int i = 0; i < 20; i++)
    {
        WardBars[i] += " class ='blue' ";
    }
}

if (Request["V1Color"] == "grey")
{
    for (int i = 0; i < 20; i++)
    {
        WardBars[i] += " class ='grey' ";
    }
}

if (Request["V1Color"] == "purple")
{
    for (int i = 0; i < 20; i++)
    {
        WardBars[i] += " class ='purple' ";
    }
}

if (Request["V1Color"] == "red")
{
    for (int i = 0; i < 20; i++)
    {
        WardBars[i] += " class ='red' ";
    }
}

for (int i = 0; i < 20; i++)
{
    if (width[i] < 0)
    {
        statDraw[i] += " x = '" + (125) + "'>" + ((width[i] / (float)200)
* float.Parse(Request["max"])) + "</text>";
    }
}

```

```

    }
    else
    {
        statDraw[i] += " x = '" + (125 + width[i]) + "'>" + ((width[i] /
(float)200) * float.Parse(Request["max"])) + "</text>";
    }
    Response.Write(statDraw[i].ToString());
    WardBars[i] += "x = '120' />";
    Response.Write(WardBars[i].ToString());
    Response.Write(WardLabels[i].ToString());
}
Response.Write("<text x = '0' y = '14' font-family = 'Arial' font-
size = '14'>" + Request["Var1Select"].ToString() + "</text>");

if (float.Parse(Request["max2"]) != 0)
{
    string[] WardBars2 =
    {
        "<rect id = 'AbbeyRect' height = '15' y = '20'",
        "<rect id = 'CannonHillRect' height = '15' y = '45'",
        "<rect id = 'ColliersWoodRect' height = '15' y = '70'",
        "<rect id = 'CricketGreenRect' height = '15' y = '95'",
        "<rect id = 'DundonaldRect' height = '15' y = '120'",
        "<rect id = 'FiggesMarshRect' height = '15' y = '145'",
        "<rect id = 'GraveneyRect' height = '15' y = '170'",
        "<rect id = 'HillsideRect' height = '15' y = '195'",
        "<rect id = 'LavenderFieldsRect' height = '15' y = '220'",
        "<rect id = 'LongthorntonRect' height = '15' y = '245'",
        "<rect id = 'LowerMordenRect' height = '15' y = '270'",
        "<rect id = 'MertonParkRect' height = '15' y = '295'",
        "<rect id = 'PollardsHillRect' height = '15' y = '320'",
        "<rect id = 'RavensburyRect' height = '15' y = '345'",
        "<rect id = 'RaynesParkRect' height = '15' y = '370'",
        "<rect id = 'StHelierRect' height = '15' y = '395'",
        "<rect id = 'TrinityRect' height = '15' y = '420'",
        "<rect id = 'VillageRect' height = '15' y = '445'",
        "<rect id = 'WestBarnesRect' height = '15' y = '470'",
        "<rect id = 'WimbledonRect' height = '15' y = '495'"
    };

    string[] WardLabels2 =
    {
        "<text id = 'AbbeyLbl' x = '400' y = '31' font-family = 'Arial'
font-size = '14'>Abbey</text>",
        "<text id = 'CannonHillLbl' x = '400' y = '56' font-family =
'Arial' font-size = '14'>Cannon Hill</text>",
        "<text id = 'ColliersWoodLbl' x = '400' y = '81' font-family =
'Arial' font-size = '14'>Colliers Wood</text>",
        "<text id = 'CricketGreenLbl' x = '400' y = '106' font-family =
'Arial' font-size = '14'>Cricket Green</text>",
        "<text id = 'DundonaldLbl' x = '400' y = '131' font-family =
'Arial' font-size = '14'>Dundonald</text>",

```

```

        "<text id = 'FiggesMarshLbl' x = '400' y = '156' font-family =
'Arial' font-size = '14'>Figges Marsh</text>",
        "<text id = 'GraveneyLbl' x = '400' y = '181' font-family =
'Arial' font-size = '14'>Graveney</text>",
        "<text id = 'HillsideLbl' x = '400' y = '206' font-family =
'Arial' font-size = '14'>Hillside</text>",
        "<text id = 'LavenderFieldsLbl' x = '400' y = '231' font-family =
'Arial' font-size = '14'>Lavender Fields</text>",
        "<text id = 'LongthorntonLbl' x = '400' y = '256' font-family =
'Arial' font-size = '14'>Longthornton</text>",
        "<text id = 'LowerMordenLbl' x = '400' y = '281' font-family =
'Arial' font-size = '14'>Lower Morden</text>",
        "<text id = 'MertonParkLbl' x = '400' y = '306' font-family =
'Arial' font-size = '14'>Merton Park</text>",
        "<text id = 'PollardsHillLbl' x = '400' y = '331' font-family =
'Arial' font-size = '14'>Pollards Hill</text>",
        "<text id = 'RavensburyLbl' x = '400' y = '356' font-family =
'Arial' font-size = '14'>Ravensbury</text>",
        "<text id = 'RaynesParkLbl' x = '400' y = '381' font-family =
'Arial' font-size = '14'>Raynes Park</text>",
        "<text id = 'StHelierLbl' x = '400' y = '406' font-family =
'Arial' font-size = '14'>St Helier</text>",
        "<text id = 'TrinityLbl' x = '400' y = '431' font-family =
'Arial' font-size = '14'>Trinity</text>",
        "<text id = 'VillageLbl' x = '400' y = '456' font-family =
'Arial' font-size = '14'>Village</text>",
        "<text id = 'WestBarnesLbl' x = '400' y = '481' font-family =
'Arial' font-size = '14'>West Barnes</text>",
        "<text id = 'WimbledonLbl' x = '400' y = '506' font-family =
'Arial' font-size = '14'>Wimbledon</text>"
};

```

```

float[] width2 =
{
    200 *
(float.Parse(Request["AbbeyCircle"])/float.Parse(Request["max2"])),
    200 *
(float.Parse(Request["CannonHillCircle"])/float.Parse(Request["max2"])),
    200 *
(float.Parse(Request["ColliersWoodCircle"])/float.Parse(Request["max2"])),
    200 *
(float.Parse(Request["CricketGreenCircle"])/float.Parse(Request["max2"])),
    200 *
(float.Parse(Request["DundonaldCircle"])/float.Parse(Request["max2"])),
    200 *
(float.Parse(Request["FiggesMarshCircle"])/float.Parse(Request["max2"])),
    200 *
(float.Parse(Request["GraveneyCircle"])/float.Parse(Request["max2"])),
    200 *
(float.Parse(Request["HillsideCircle"])/float.Parse(Request["max2"])),
    200 *
(float.Parse(Request["LavenderFieldsCircle"])/float.Parse(Request["max2"])),
    200 *
(float.Parse(Request["LongthorntonCircle"])/float.Parse(Request["max2"])),

```

```

    200 *
(float.Parse(Request["LowerMordenCircle"])/float.Parse(Request["max2"])),
    200 *
(float.Parse(Request["MertonParkCircle"])/float.Parse(Request["max2"])),
    200 *
(float.Parse(Request["PollardsHillCircle"])/float.Parse(Request["max2"])),
    200 *
(float.Parse(Request["RavensburyCircle"])/float.Parse(Request["max2"])),
    200 *
(float.Parse(Request["RaynesParkCircle"])/float.Parse(Request["max2"])),
    200 *
(float.Parse(Request["StHelierCircle"])/float.Parse(Request["max2"])),
    200 *
(float.Parse(Request["TrinityCircle"])/float.Parse(Request["max2"])),
    200 *
(float.Parse(Request["VillageCircle"])/float.Parse(Request["max2"])),
    200 *
(float.Parse(Request["WestBarnesCircle"])/float.Parse(Request["max2"])),
    200 *
(float.Parse(Request["WimbledonCircle"])/float.Parse(Request["max2"]))
};

```

```

string[] statDraw2 =
{
    "<text y = '33' font-family = 'Arial' font-size = '14'",
    "<text y = '58' font-family = 'Arial' font-size = '14'",
    "<text y = '83' font-family = 'Arial' font-size = '14'",
    "<text y = '108' font-family = 'Arial' font-size = '14'",
    "<text y = '133' font-family = 'Arial' font-size = '14'",
    "<text y = '158' font-family = 'Arial' font-size = '14'",
    "<text y = '183' font-family = 'Arial' font-size = '14'",
    "<text y = '208' font-family = 'Arial' font-size = '14'",
    "<text y = '233' font-family = 'Arial' font-size = '14'",
    "<text y = '258' font-family = 'Arial' font-size = '14'",
    "<text y = '283' font-family = 'Arial' font-size = '14'",
    "<text y = '308' font-family = 'Arial' font-size = '14'",
    "<text y = '333' font-family = 'Arial' font-size = '14'",
    "<text y = '358' font-family = 'Arial' font-size = '14'",
    "<text y = '383' font-family = 'Arial' font-size = '14'",
    "<text y = '408' font-family = 'Arial' font-size = '14'",
    "<text y = '433' font-family = 'Arial' font-size = '14'",
    "<text y = '458' font-family = 'Arial' font-size = '14'",
    "<text y = '483' font-family = 'Arial' font-size = '14'",
    "<text y = '508' font-family = 'Arial' font-size = '14'"
};

```

```

for (int i = 0; i < 20; i++)
{
    if (width2[i] < 0)
    {
        WardBars2[i] += " width='0'";
    }
    else
    {

```

```

        WardBars2[i] += " width='" + width2[i].ToString() + "'";
    }
}

if (Request["V2Color"] == "green")
{
    for (int i = 0; i < 20; i++)
    {
        WardBars2[i] += " class ='green' ";
    }
}

if (Request["V2Color"] == "blue")
{
    for (int i = 0; i < 20; i++)
    {
        WardBars2[i] += " class ='blue' ";
    }
}

if (Request["V2Color"] == "grey")
{
    for (int i = 0; i < 20; i++)
    {
        WardBars2[i] += " class ='grey' ";
    }
}

if (Request["V2Color"] == "purple")
{
    for (int i = 0; i < 20; i++)
    {
        WardBars2[i] += " class ='purple' ";
    }
}

if (Request["V2Color"] == "red")
{
    for (int i = 0; i < 20; i++)
    {
        WardBars2[i] += " class ='red' ";
    }
}

for (int i = 0; i < 20; i++)
{
    if (width2[i] < 0)
    {
        statDraw2[i] += " x = '" + (525) + "'>" + ((width2[i] /
(float)200) * float.Parse(Request["max2"])) + "</text>";
    }
    else
    {

```

```

        statDraw2[i] += " x = '" + (525 + width2[i]) + "'>" +
((width2[i] / (float)200) * float.Parse(Request["max2"])) + "</text>";
    }
    Response.Write(statDraw2[i]);
    WardBars2[i] += "x = '520'/>";
    Response.Write(WardBars2[i].ToString());
    Response.Write(WardLabels2[i].ToString());
}

Response.Write("<text x = '400' y = '14' font-family = 'Arial' font-
size = '14'>" + Request["Var2Select"].ToString() + "</text>");

}

if (float.Parse(Request["max3"]) != 0)
{
    string[] WardBars3 =
    {
        "<rect id = 'AbbeyRect' height = '15' y = '20'",
        "<rect id = 'CannonHillRect' height = '15' y = '45'",
        "<rect id = 'ColliersWoodRect' height = '15' y = '70'",
        "<rect id = 'CricketGreenRect' height = '15' y = '95'",
        "<rect id = 'DundonaldRect' height = '15' y = '120'",
        "<rect id = 'FiggesMarshRect' height = '15' y = '145'",
        "<rect id = 'GraveneyRect' height = '15' y = '170'",
        "<rect id = 'HillsideRect' height = '15' y = '195'",
        "<rect id = 'LavenderFieldsRect' height = '15' y = '220'",
        "<rect id = 'LongthorntonRect' height = '15' y = '245'",
        "<rect id = 'LowerMordenRect' height = '15' y = '270'",
        "<rect id = 'MertonParkRect' height = '15' y = '295'",
        "<rect id = 'PollardsHillRect' height = '15' y = '320'",
        "<rect id = 'RavensburyRect' height = '15' y = '345'",
        "<rect id = 'RaynesParkRect' height = '15' y = '370'",
        "<rect id = 'StHelierRect' height = '15' y = '395'",
        "<rect id = 'TrinityRect' height = '15' y = '420'",
        "<rect id = 'VillageRect' height = '15' y = '445'",
        "<rect id = 'WestBarnesRect' height = '15' y = '470'",
        "<rect id = 'WimbledonRect' height = '15' y = '495'"
    };

    string[] WardLabels3 =
    {
        "<text id = 'AbbeyLbl' x = '800' y = '31' font-family = 'Arial'
font-size = '14'>Abbey</text>",
        "<text id = 'CannonHillLbl' x = '800' y = '56' font-family =
'Arial' font-size = '14'>Cannon Hill</text>",
        "<text id = 'ColliersWoodLbl' x = '800' y = '81' font-family =
'Arial' font-size = '14'>Colliers Wood</text>",
        "<text id = 'CricketGreenLbl' x = '800' y = '106' font-family =
'Arial' font-size = '14'>Cricket Green</text>",
    }
}

```

```

        "<text id = 'DundonaldLbl' x = '800' y = '131' font-family =
'Arial' font-size = '14'>Dundonald</text>",
        "<text id = 'FiggesMarshLbl' x = '800' y = '156' font-family =
'Arial' font-size = '14'>Figges Marsh</text>",
        "<text id = 'GraveneyLbl' x = '800' y = '181' font-family =
'Arial' font-size = '14'>Graveney</text>",
        "<text id = 'HillsideLbl' x = '800' y = '206' font-family =
'Arial' font-size = '14'>Hillside</text>",
        "<text id = 'LavenderFieldsLbl' x = '800' y = '231' font-family =
'Arial' font-size = '14'>Lavender Fields</text>",
        "<text id = 'LongthorntonLbl' x = '800' y = '256' font-family =
'Arial' font-size = '14'>Longthornton</text>",
        "<text id = 'LowerMordenLbl' x = '800' y = '281' font-family =
'Arial' font-size = '14'>Lower Morden</text>",
        "<text id = 'MertonParkLbl' x = '800' y = '306' font-family =
'Arial' font-size = '14'>Merton Park</text>",
        "<text id = 'PollardsHillLbl' x = '800' y = '331' font-family =
'Arial' font-size = '14'>Pollards Hill</text>",
        "<text id = 'RavensburyLbl' x = '800' y = '356' font-family =
'Arial' font-size = '14'>Ravensbury</text>",
        "<text id = 'RaynesParkLbl' x = '800' y = '381' font-family =
'Arial' font-size = '14'>Raynes Park</text>",
        "<text id = 'StHelierLbl' x = '800' y = '406' font-family =
'Arial' font-size = '14'>St Helier</text>",
        "<text id = 'TrinityLbl' x = '800' y = '431' font-family =
'Arial' font-size = '14'>Trinity</text>",
        "<text id = 'VillageLbl' x = '800' y = '456' font-family =
'Arial' font-size = '14'>Village</text>",
        "<text id = 'WestBarnesLbl' x = '800' y = '481' font-family =
'Arial' font-size = '14'>West Barnes</text>",
        "<text id = 'WimbledonLbl' x = '800' y = '506' font-family =
'Arial' font-size = '14'>Wimbledon</text>"
};

```

```

string[] statDraw3 =
{
    "<text y = '33' font-family = 'Arial' font-size = '14'",
    "<text y = '58' font-family = 'Arial' font-size = '14'",
    "<text y = '83' font-family = 'Arial' font-size = '14'",
    "<text y = '108' font-family = 'Arial' font-size = '14'",
    "<text y = '133' font-family = 'Arial' font-size = '14'",
    "<text y = '158' font-family = 'Arial' font-size = '14'",
    "<text y = '183' font-family = 'Arial' font-size = '14'",
    "<text y = '208' font-family = 'Arial' font-size = '14'",
    "<text y = '233' font-family = 'Arial' font-size = '14'",
    "<text y = '258' font-family = 'Arial' font-size = '14'",
    "<text y = '283' font-family = 'Arial' font-size = '14'",
    "<text y = '308' font-family = 'Arial' font-size = '14'",
    "<text y = '333' font-family = 'Arial' font-size = '14'",
    "<text y = '358' font-family = 'Arial' font-size = '14'",
    "<text y = '383' font-family = 'Arial' font-size = '14'",
    "<text y = '408' font-family = 'Arial' font-size = '14'",
    "<text y = '433' font-family = 'Arial' font-size = '14'",
    "<text y = '458' font-family = 'Arial' font-size = '14'",

```

```

    "<text y = '483' font-family = 'Arial' font-size = '14'",
    "<text y = '508' font-family = 'Arial' font-size = '14'"
};

float[] width3 =
{
    200 *
(float.Parse(Request["AbbeyRad"])/float.Parse(Request["max3"])),
    200 *
(float.Parse(Request["CannonHillRad"])/float.Parse(Request["max3"])),
    200 *
(float.Parse(Request["ColliersWoodRad"])/float.Parse(Request["max3"])),
    200 *
(float.Parse(Request["CricketGreenRad"])/float.Parse(Request["max3"])),
    200 *
(float.Parse(Request["DundonaldRad"])/float.Parse(Request["max3"])),
    200 *
(float.Parse(Request["FiggesMarshRad"])/float.Parse(Request["max3"])),
    200 *
(float.Parse(Request["GraveneyRad"])/float.Parse(Request["max3"])),
    200 *
(float.Parse(Request["HillsideRad"])/float.Parse(Request["max3"])),
    200 *
(float.Parse(Request["LavenderFieldsRad"])/float.Parse(Request["max3"])),
    200 *
(float.Parse(Request["LongthorntonRad"])/float.Parse(Request["max3"])),
    200 *
(float.Parse(Request["LowerMordenRad"])/float.Parse(Request["max3"])),
    200 *
(float.Parse(Request["MertonParkRad"])/float.Parse(Request["max3"])),
    200 *
(float.Parse(Request["PollardsHillRad"])/float.Parse(Request["max3"])),
    200 *
(float.Parse(Request["RavensburyRad"])/float.Parse(Request["max3"])),
    200 *
(float.Parse(Request["RaynesParkRad"])/float.Parse(Request["max3"])),
    200 *
(float.Parse(Request["StHelierRad"])/float.Parse(Request["max3"])),
    200 *
(float.Parse(Request["TrinityRad"])/float.Parse(Request["max3"])),
    200 *
(float.Parse(Request["VillageRad"])/float.Parse(Request["max3"])),
    200 *
(float.Parse(Request["WestBarnesRad"])/float.Parse(Request["max3"])),
    200 *
(float.Parse(Request["WimbledonRad"])/float.Parse(Request["max3"]))
};

for (int i = 0; i < 20; i++)
{
    if (width3[i] < 0)
    {
        WardBars3[i] += " width='0' class = 'green' ";
    }
}

```

```

        else
        {
            WardBars3[i] += " width='" + width3[i].ToString() + "' class
= 'green' ";
        }
    }

    for (int i = 0; i < 20; i++)
    {
        if (width3[i] < 0)
        {
            statDraw3[i] += " x = '" + (925) + "'>" + ((width3[i] /
(float)200) * float.Parse(Request["max3"])) + "</text>";
        }
        else
        {
            statDraw3[i] += " x = '" + (925 + width3[i]) + "'>" +
((width3[i] / (float)200) * float.Parse(Request["max3"])) + "</text>";
        }
        Response.Write(statDraw3[i]);
        WardBars3[i] += "x = '920' />";
        Response.Write(WardBars3[i].ToString());
        Response.Write(WardLabels3[i].ToString());
    }
    Response.Write("<text x = '800' y = '14' font-family = 'Arial' font-
size = '14'>" + Request["Var3Select"].ToString() + "</text>");
}

%>

```