



URBAN SCIENCE™

Project Plan

Infographic Generator

Louis Bodnar
Peter Chen
Lok Cheung
Kevin Shreve

February 18, 2012

Contents

1	Executive Summary	3
2	Functional Specifications	3
3	Design Specifications	4
3.1	Overview	4
3.2	System Architecture	4
3.3	User Interface	5
4	Technical Specifications	9
5	Schedule	12
5.1	Week 1 (Jan 9 - Jan 15)	12
5.2	Week 2 (Jan 16 - Jan 22)	12
5.3	Week 3 (Jan 23 - Jan 29)	12
5.4	Week 4 (Jan 30 - Feb 5)	12
5.5	Week 4 (Feb 6 - Feb 12)	13
6	Risks	13
6.1	Languages	13
6.2	Ambiguous Requirements	13
6.3	Scheduling	13
6.4	Art	13

List of Figures

1	Use Case Diagram	4
2	Main Menu Mockup	5
3	Infographic Display Mockup	6
4	Infographic Display Mockup	7
5	Infographic Display Mockup	8
6	Sequence Diagram	10
7	Class Diagram	11

1 Executive Summary

Over three decades ago, a professor at Wayne State University took on a challenge because Cadillac called it unsolvable. That man was Jim Anderson. His computer-generated dot maps gave Cadillac's marketing department a competitive advantage by allowing them to easily visualize dealership locations across the nation. Jim began using the power of computers to analyze these networks of dealerships and started a company that specialized in the planning and management of these networks. Thus, Urban Science was born.

Urban Science is now an international company, headquartered in Detroit, Michigan. They assist nearly every original equipment manufacturer (OEM) in over 60 countries. Our job is to design a webapp to show OEMs monthly performance data on vehicle sales, lead management, and service for their primary market area. OEMs use this performance data to adjust spending to maximize their market potential. The appeal of our webapp is its ease of use, and visual appeal.

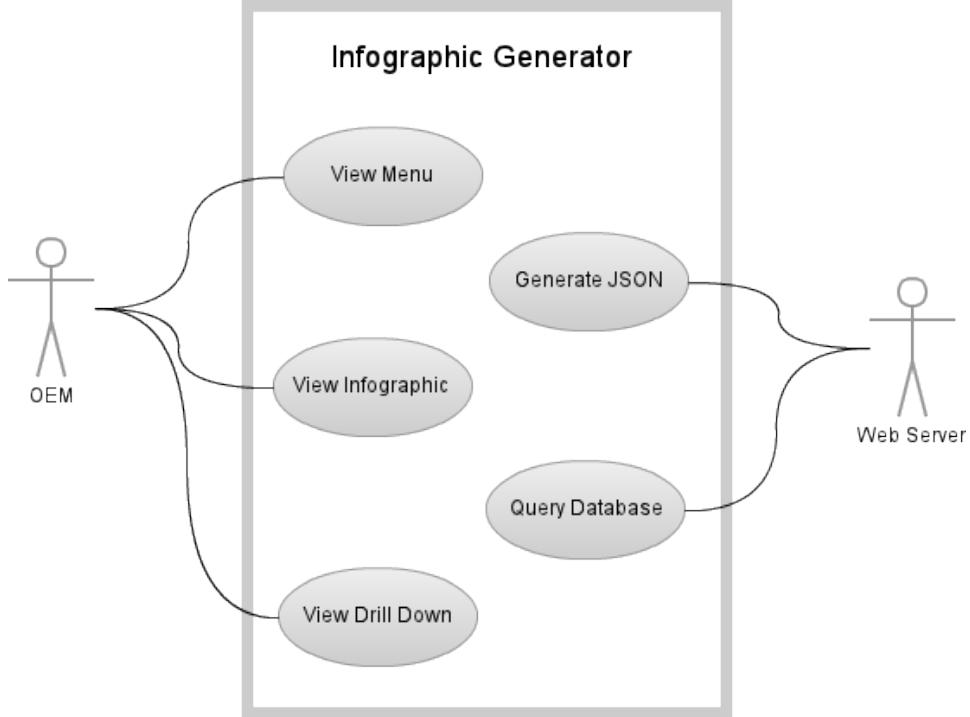
In today's world, the display of information is an evolving art. Yesterday's solution was clipart and spreadsheets, but that is no longer enough. We require a more impactful approach to delivering a point. The modern solution is an infographic or infographic. An infographic is a graphical display that quickly conveys data that would otherwise require a lengthy explanation. Modern infographics use clever design schemes and cartoon characters to keep the viewer interested and to show relationships in the data.

Our webapp uses a brand new type of infographic that is designed to update dynamically. It uses information directly from Urban Science to generate graphics that reflect the most up-to-date monthly data. The webapp also provides the ability to view previous months to allow OEMs to reference historical data.

2 Functional Specifications

The webapp focus of project is to create a new and creative experience for the KPI data. The user interface will be very simple, allowing for on the fly use by those giving presentations or demonstrating the product to clients. When a client sees this unique display of information they will assuredly choose Urban Science to help them with their strategy.

Figure 1: Use Case Diagram



3 Design Specifications

3.1 Overview

The general interface is based on a Lazy Susan like menu. A user can swipe the screen to rotate the icons, and just touch on anyone of them to select a category. The program will trigger the infographic generator depending on the category that was selected. The infographic generator will then grab the preloaded data from the xml file and display with a default graph. Then user can select different style of infographic to display the information as they need.

3.2 System Architecture

The infographic application is comprised of two major components: a web application/user interface and a backend database. Users are assumed to use safari browser on iPad. The web host then displays the menu screen. Data will be pulled until specific category is selected. When a category is selected, the program will pull up all the required data from the server. Depending on the style of infographic, data will be displayed in different form.

Figure 2: Main Menu Mockup



3.3 User Interface

Figure 2 shows the screen mockup for our prototype. The main page will be displayed in lazy Suzan style that allows the user to scroll through the icons to the right or left. The infographic for the current category will be shown when the icon is clicked on.

Figure 3, Figure 4, and Figure 5 show a sample infographic depiction. If you want to view data for another month, you can just swipe left and right to change the time frame. If you want to go back to the main menu page, just click on the top right corners menu button.

Figure 3: Infographic Display Mockup



Figure 4: Infographic Display Mockup



Figure 5: Infographic Display Mockup



4 Technical Specifications

The infographic generator will produce infographics suitable for display in a browser with support for HTML5 and JavaScript.

Each infographic will be based off of KPI data returned from a Microsoft SQL database. The server will use an asp.net script to query the database and return the information to the client device. The client device will integrate this data with predefined infographic scripts written in a combination of HTML5 and JavaScript to generate the infographic.

An infographic framework consisting of the server side database query script and a client side JavaScript library will be used to assist in the creation of infographics. The JavaScript library will contain functions to create reusable animations and artwork based off values obtained from the database query script.

The interface for viewing infographics will be written in HTML5 and JavaScript. It will use jQuery to support swipe events on the iPad for scrolling left and right in the menus. The menus will be populated with items read from an XML file. Only the front most item (the item with the greatest cascading style sheet z-index) in the menu will be selectable. The front most item will change to the item on its immediate left if swiped right and vice-versa if swiped left.

The main menu will be used to select a category. Once the category has been selected, the contents screen will change to show the infographic for the selected category. The user will have the ability to return to the main menu while viewing an infographic. This functionality will be provided by a button located on each infographic display page but not as part of the main menu.

Figure 6: Sequence Diagram

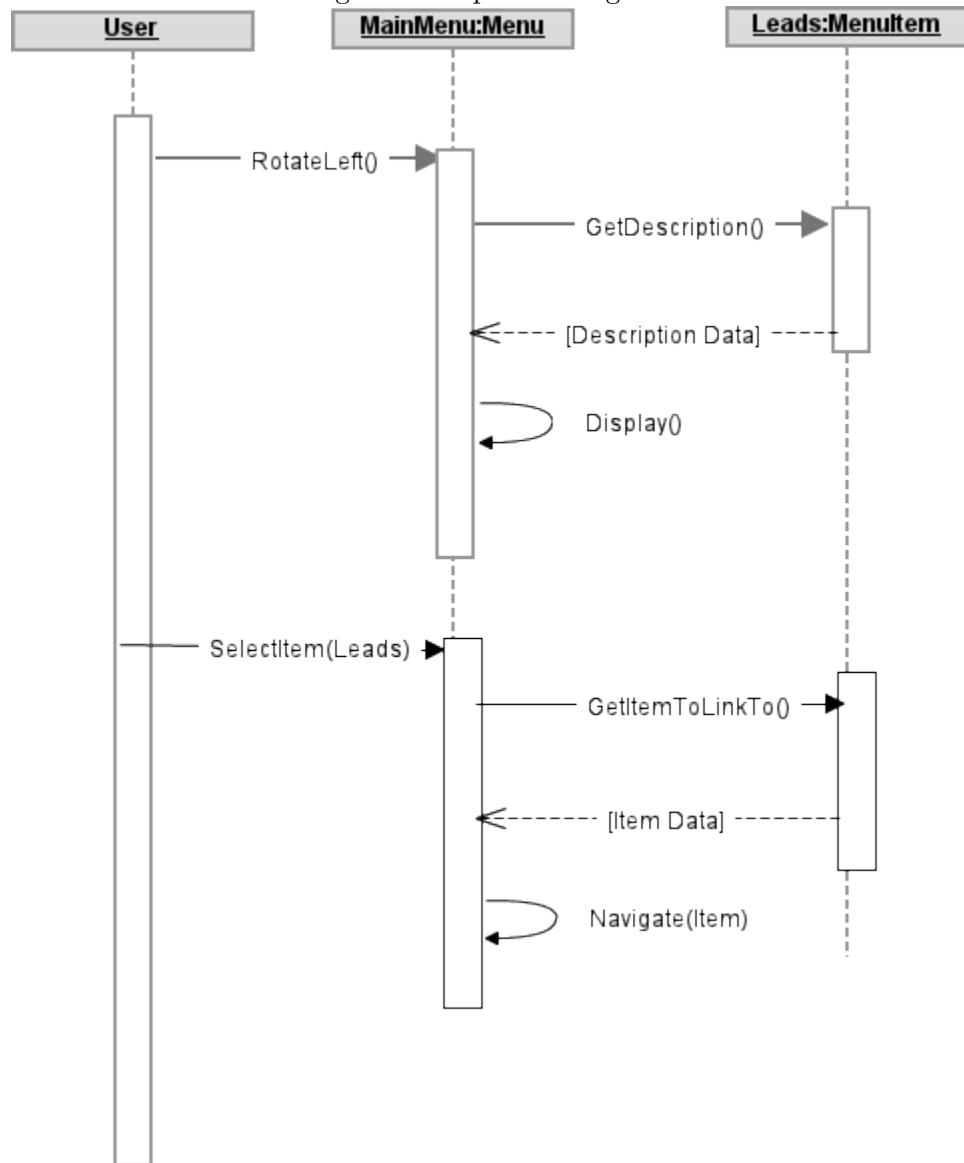
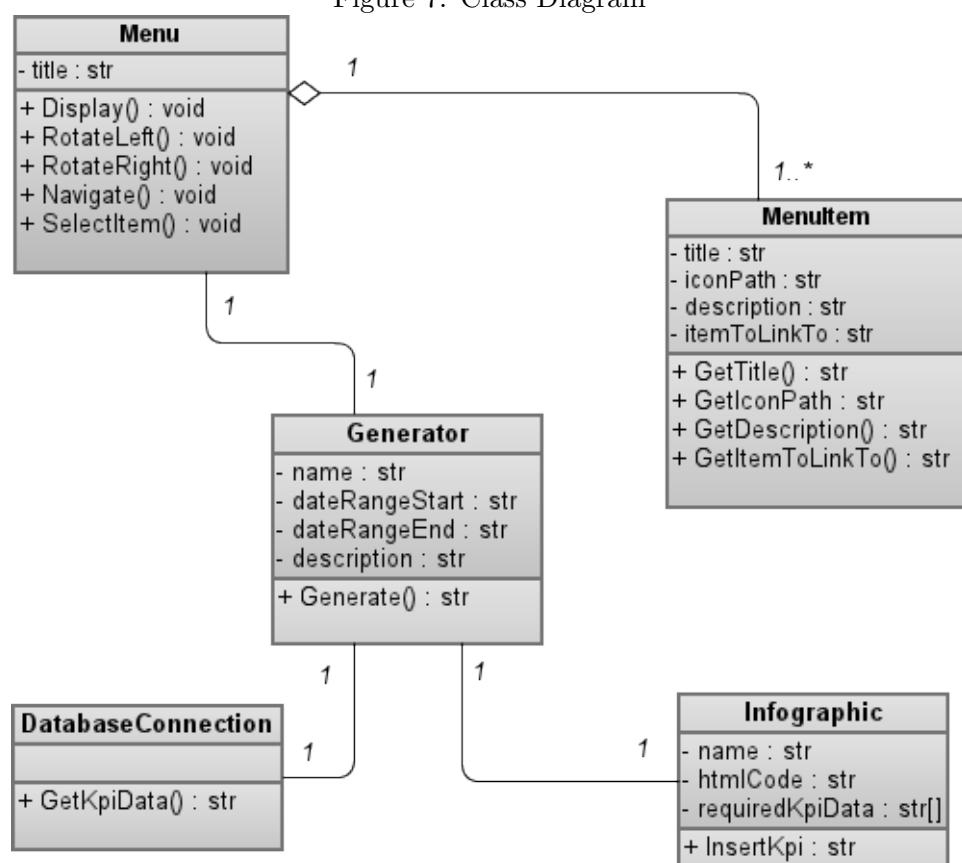


Figure 7: Class Diagram



5 Schedule

5.1 Week 1 (Jan 9 - Jan 15)

- Had our first conference call with our customer
- Set up weekly conference calls with our customer
- Set up regular team meetings to meet twice a week
- Installed virtual machines

5.2 Week 2 (Jan 16 - Jan 22)

- Started UML diagrams
- Installed Windows Server 2008 R2
- Installed IIS 7
- Installed ASP.NET
- Installed Microsoft SQL Server
- Talked with our customer about interface mockups

5.3 Week 3 (Jan 23 - Jan 29)

- Completed more UML diagrams
- First draft of project-plan
- Installed Visual Studio 2010
- Created screen mockups
- Created sample infographic elements

5.4 Week 4 (Jan 30 - Feb 5)

- Created infographic elements using actual KPI data
- Wrote website to showcase infographics
- Decided to work on sales infographic first

5.5 Week 4 (Feb 6 - Feb 12)

- Presented project plan to class
- Changed from using XML to JSON for pulling database information
- Have roundabout working with several images for infographic selector buttons
- Successfully pulled data from a SQL database using ASP.NET

6 Risks

6.1 Languages

- Risk: Unfamiliar with ASP.NET and HTML5
- Mitigation: Read online tutorials, books, and documentation

6.2 Ambiguous Requirements

- Risk: Customer gave broad requirements
- Mitigation: Help customer find focus by using mockups and eliciting information

6.3 Scheduling

- Risk: Not much free time overlap between our schedules
- Mitigation: Keep everybody up-to-date with emails and plan ahead

6.4 Art

- Risk: Little artistic experience in our group
- Mitigation: Brainstorm and find inspiration from the artwork used in popular applications