



URBAN SCIENCE™

## Infographics Generator

### Project Plan

Louis Bodnar  
Peter Chen  
Lok Cheung  
Kevin Shreve

April 10, 2012

## Contents

<b>1</b>	<b>Executive Summary</b>	<b>2</b>
<b>2</b>	<b>Functional Specifications</b>	<b>2</b>
2.1	Web Application . . . . .	2
2.1.1	Main Menu . . . . .	2
2.1.2	Infographic Display Page . . . . .	3
2.2	Web Backend . . . . .	3
<b>3</b>	<b>Design Specifications</b>	<b>3</b>
3.1	Overview . . . . .	3
3.2	System Architecture . . . . .	3
3.3	User Interface . . . . .	3
<b>4</b>	<b>Technical Specifications</b>	<b>7</b>
<b>5</b>	<b>Schedule</b>	<b>11</b>
5.1	Week 1 (January 9, 2012) . . . . .	11
5.2	Week 2 (January 16, 2012) . . . . .	11
5.3	Week 3 (January 23, 2012) . . . . .	11
5.4	Milestone: Status Reports . . . . .	11
5.5	Week 4 (January 30, 2012) . . . . .	11
5.6	Milestone: Project Plan Presentation . . . . .	12
5.7	Week 5 (February 6, 2012) . . . . .	12
5.8	Week 6 (February 13, 2012) . . . . .	12
5.9	Week 7 (February 20, 2012) . . . . .	12
5.10	Milestone: Alpha Demonstrations . . . . .	12
5.11	Week 8 (February 27, 2012) . . . . .	12
5.12	Week 9 (March 12, 2012) . . . . .	12
5.13	Week 10 (March 19, 2012) . . . . .	13
5.14	Week 11 (March 26, 2012) . . . . .	13
5.15	Milestone: Beta Demonstrations . . . . .	13
5.16	Week 12 (April 2, 2012) . . . . .	13
5.17	Week 13 (April 9, 2012) . . . . .	13
5.18	Week 14 (April 16, 2012) . . . . .	13
5.19	Week 15 (April 23, 2012) . . . . .	13
5.20	Milestone: Projct Video . . . . .	14
5.21	Milestone: All Deliverables . . . . .	14
5.22	Milestone: Design Day . . . . .	14
<b>6</b>	<b>Remaining tasks</b>	<b>14</b>

## 1 Executive Summary

Over three decades ago, a professor at Wayne State University took on a challenge that Cadillac deemed unsolvable. That professor 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 web application 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 web application 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 engaging approach to delivering a point. The modern solution is an information graphic 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 while showing relationships in the data.

Our web application uses a brand new flavor 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 web application also provides the ability to view previous months to allow OEMs to reference historical data.

## 2 Functional Specifications

### 2.1 Web Application

- uses local storage

#### 2.1.1 Main Menu

- has lazy susan style interface -

### 2.1.2 Infographic Display Page

- has navigation buttons - has infographic

## 2.2 Web Backend

- has database - outputs json
- features: - infographics - months - menu - dynamicness

The web application is 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.

## 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 infographics generator depending on the category that was selected. The infographics 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.

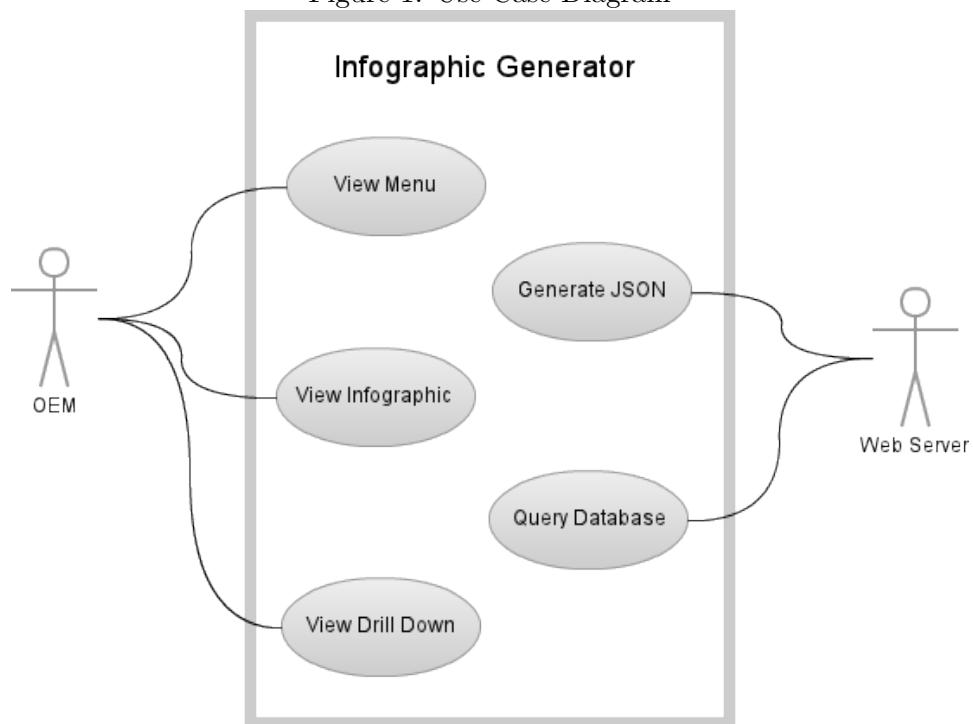
### 3.2 System Architecture

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.

### 3.3 User Interface

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 1: Use Case Diagram



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.







## 4 Technical Specifications

The infographics generator produces infographics suitable for display on the iPad 2 using the Safari browser.

The Index.html page in our webapp contains the infographic selector menu. The infographic selector menu is used to browse all available infographics. The infographics available to choose from include: Sales, Service, and Lead Management. The selector interface has three icons, each representing an infographic, arranged in a horizontally elongated ellipse. A swipe in the left direction causes the icons to rotate clockwise following the elliptical path. Likewise, the icons rotate counter clockwise when swiped in the right direction. This creates the illusion that the icons are rotating on an invisible lazy susan. By swiping, the user feels that they are causing the invisible lazy susan to rotate. Upon pressing the front most icon, Safari navigates to the corresponding infographic's page.

The infographic display page contains a floating image called the header. The header spans the width of the page and remains at the top of the page when the user scrolls. The header has a button on the left hand side which,

if pressed, navigates back to the infographic selector menu. The right side of the header has text displaying the month and year of the data being displayed. The infographic is located immediately beneath the header and spans the entire width of the page. If the infographic is swiped left, the next month's data is displayed. If swiped right, the previous month's data is displayed. If no data exists for the month attempting to be displayed, the current date is not changed.

Each infographic displays multiple key performance indicators (KPI) data. The data is obtained from a Microsoft SQL database. The server hosting the infographics generator uses an ASP.NET script (KPIHandler.ashx) to query the database for all available KPI data, serializes the data, and returns the data to the client device in JavaScript Object Notation (JSON). The client device then parses this data using a JavaScript function (found in Scripts/KPILocalStorage.js) and adds the data to local storage.

An infographic element is a visual representation of KPI data, responsible for displaying one or more KPI data. An infographic is comprised of multiple infographic elements. Each infographic element is drawn on an HTML5 canvas using JavaScript functions found in one of the three infographic drawing files (draw-sales.js, draw-service.js, draw-lead.js) or the common element file (elements.js). The common element file is used to centralize the generic JavaScript functions that are used to draw the elements in the infographics. The infographic specific files call functions from the common element file and have their own JavaScript functions to draw the infographic. They also calls functions to retrieve specific KPI values from local storage.

A drill down display is shown in the center of the screen if an infographic element is tapped. The drill down display gives an brief description of the KPI that was tapped and shows a trend chart of the KPI over the last 6 months (ending in the current month) or the maximum number of months available, whichever number is smaller.

Figure 2: Sequence Diagram

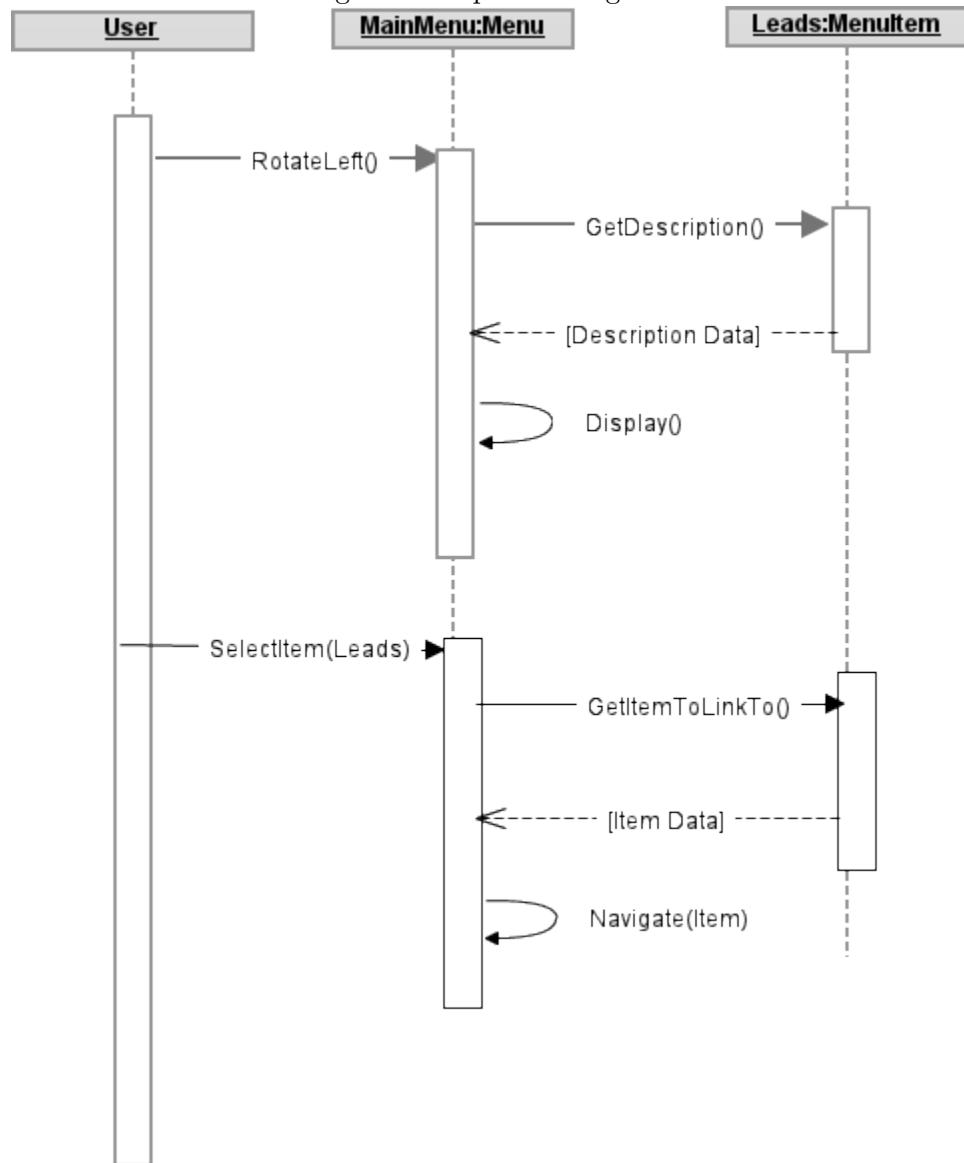
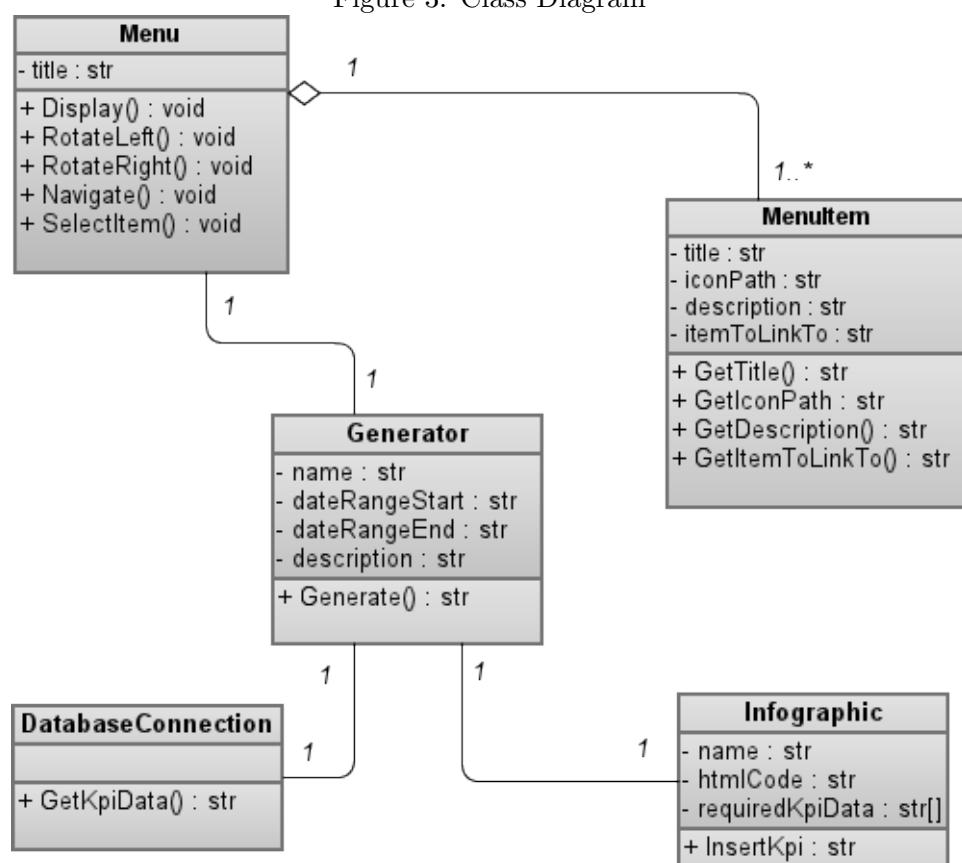


Figure 3: Class Diagram



## 5 Schedule

### 5.1 Week 1 (January 9, 2012)

- 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 (January 16, 2012)

- 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 (January 23, 2012)

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

### 5.4 Milestone: Status Reports

- Gave report to class

### 5.5 Week 4 (January 30, 2012)

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

**5.6 Milestone: Project Plan Presentation**

- Gave presentation in class

**5.7 Week 5 (February 6, 2012)**

- 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

**5.8 Week 6 (February 13, 2012)**

- Sales infographic exists in basic form
- Added background image to main menu

**5.9 Week 7 (February 20, 2012)**

- Infographics now use values from database
- Rewrote pump in sales infographic
- Practiced Alpha Demonstration

**5.10 Milestone: Alpha Demonstrations**

- Demonstrated software to Urban Science for first time
- Received useful feedback from Urban Science

**5.11 Week 8 (February 27, 2012)**

- Planning for betas
- Created mockups of service and lead management infographics

**5.12 Week 9 (March 12, 2012)**

- Added service infographic to website
- Fixed bugs with swiping
- Local storage now refreshes with website

**5.13 Week 10 (March 19, 2012)**

- Added MSU and Urban Science branding to website
- Visited Urban Science headquarters in Detroit, MI

**5.14 Week 11 (March 26, 2012)**

- Added remaining KPI to service and lead management infographics
- Implemented drill down display
- Began writing script for video
- Practiced for Beta presentation

**5.15 Milestone: Beta Demonstrations**

- Presentation was successful
- Project was feature complete

**5.16 Week 12 (April 2, 2012)**

- Changed text formatting upon customer request
- Fixed bugs with swiping

**5.17 Week 13 (April 9, 2012)**

- Updated project plan to reflect most recent version of webapp
- Discovered origin of grey background in jQuery css file

**5.18 Week 14 (April 16, 2012)**

- Has not occurred

**5.19 Week 15 (April 23, 2012)**

- TODO: clean up code
- TODO: write documentation for functions
- TODO: project video
- TODO: rename /webapp/images/avg\$.png because of potential problems with having a dollar sign in the filename

**5.20 Milestone: Project Video**

- Has not occurred

**5.21 Milestone: All Deliverables**

- Has not occurred

**5.22 Milestone: Design Day**

- Has not occurred

**6 Remaining tasks**

- clean up code
- write documentation for functions
- project video
- rename /webapp/images/Avg\$.png because of potential problems with having a dollar sign in the filename
- make close rate a percentage