URBAN SCIENCE™

Project Plan

Infographic Generator

Louis Bodnar
Peter Chen
Lok Cheung
Kevin Shreve

January 25, 2012

Contents

1   Executive Summary

The Infographics generator will create a new an intuitive way to present the Key Performance Indicators (KPI) that Urban Science has been collecting. Our HTML5 mobile application will be used to show Corporate OEMs or Ford plant managers, the exact kind of data to improve performance.

The main objective of this system is to increase usage of the KPI data, and make it easier to read and understand. Users will be able to drill-down through the categories in which they have interest. Once a category is selected, the following screen will bring up a breakdown of that category and any other related data will also be displayed. This functionality will allow for easy navigation throughout the entire application. Urban Science is in a great market position, where in bad economic times companies use their data to save money, and in good economic times companies use Urban Science to help them make money. Our application will deliver a robust solution to break down the data, so that clients can know exactly what relevant data in order to be successful.

Our software will also build the framework to allow Urban Science to continue to use our product, by being able to easily add in more infographics. This allows for future extensibility for Urban Science to keep the similar look and feel that the users will become accustomed to seeing. This product will be a key selling point for Urban Science to offer its clientele as the easy to understand interface will guide their customers towards new goals.

2   Functional Specifications

The main 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.

The system will be a standalone web app, but may have the possibility of being pulled into another website at their discretion. It will be displayed via Safari on an iPad, with the android platform or access via a desktop computer being the secondary access point.

Figure 1: Use Case Diagram
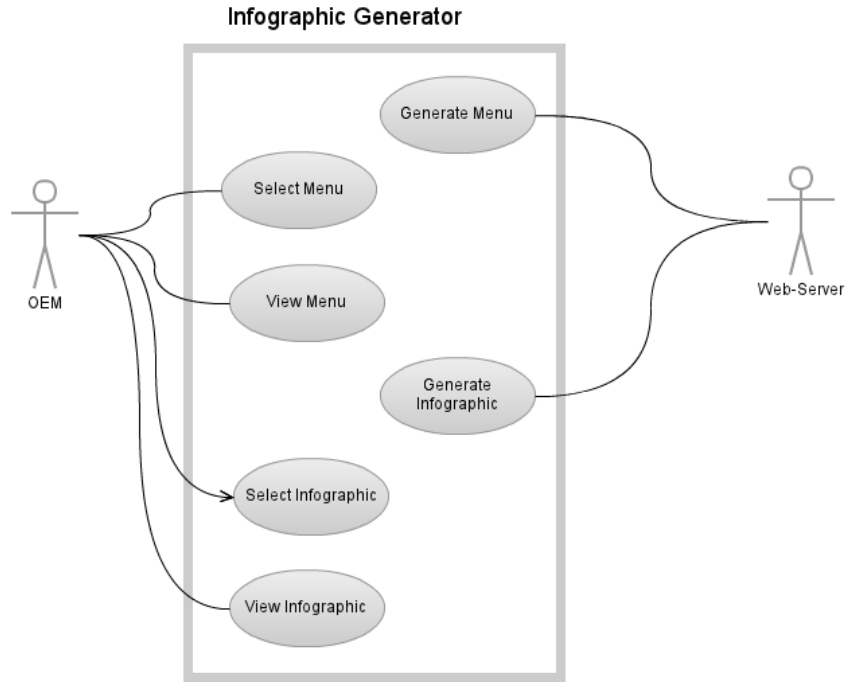
3 Design Specifications

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 display a sub-menu or trigger an infographic generator depending on the category that was selected. The infographic generator will then grab the required data from the database 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 not 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

Figure 2 shows the screen mockup for our prototype. The main page will be displayed in lazy Suzan style that you can scroll the icon to the right or left. And you click on the icon to enter the next category or the infographic for the specificity item.

Figure 3 shows the infographic depiction and information. On the top left corner is the general image for the KPI data, and on the top right we got the general description. On the bottom, we have the numerical data value and the image generated by infographic generator. If you click on the monthly report, there is be a drop with list of report types. And when you click on the Pie Chart, you can pick other kind of infographic style. If you want to view other years (months) report, you can just scroll the 2011 or 2013 icon to the right or left to change to the year (month) you want. If you want to go back to the last page you viewed before, just click on the top right corners bottom to go back.

Figure 2: Main Menu Mockup

Figure 3: 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 of the on-screen menu will change to show the infographics in the selected category. The user will have the ability to return to the main menu after selecting a category. This functionality will be provided by a button located on each menu page but not as part of the menu.

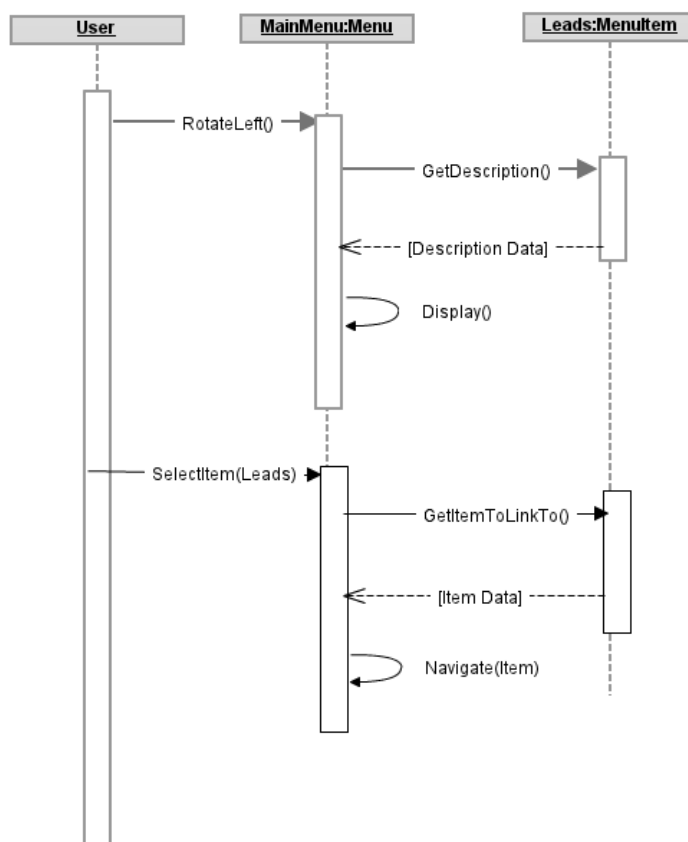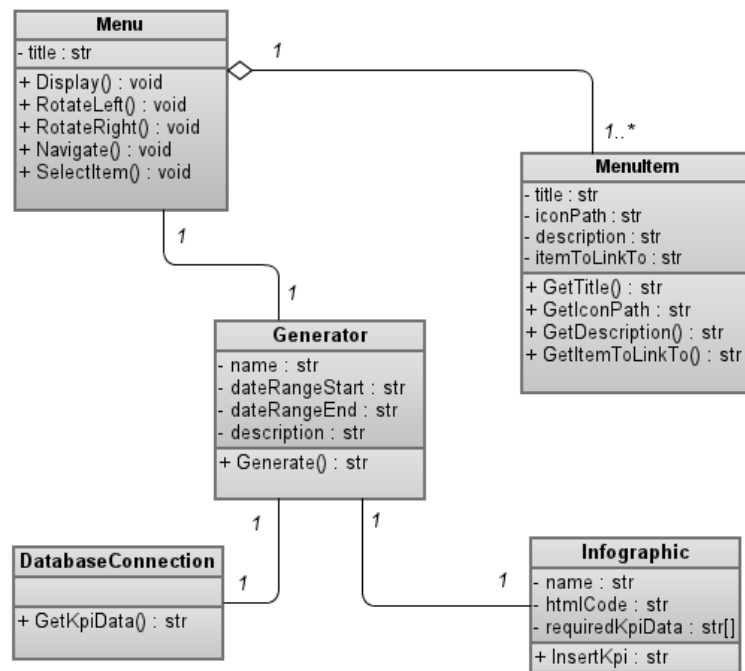Figure 4: Sequence Diagram

Sequence Diagram

User | MainMenu:Menu | Leads:MenuItem

RotateLeft()

GetDescription()

[Description Data]

Display()

SelectItem(Leads)

GetItemToLinkTo()

[Item Data]

Navigate(Item)

Figure 5: Class Diagram

Class Diagram

Menu
- title : str

+ Display() : void
+ RotateLeft() : void
+ RotateRight() : void
+ Navigate() : void
+ SelectItem() : void

1

1..*

MenuItem
- title : str
- iconPath : str
- description : str
- itemToLinkTo : str

+ GetTitle() : str
+ GetIconPath : str
+ GetDescription() : str
+ GetItemToLinkTo() : str

1

1

Generator
- name : str
- dateRangeStart : str
- dateRangeEnd : str
- description : str

+ Generate() : str

1

1

DatabaseConnection

+ GetKpiData() : str

1

1

Infographic
- name : str
- htmlCode : str
- requiredKpiData : str[]

+ InsertKpi : str

9

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 30)

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

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