

SOFTWARE DEVELOPMENT PLAN

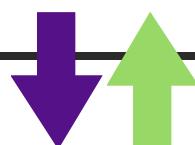
8/23/2013

Graduate Capstone



Table of Contents

1	Introduction.....	3
1.1	Intended Audience.....	3
1.2	References.....	3
1.3	Revision History	3
2	Defining Project Artifacts and Deliverables.....	4
2.1	Vision and Scope Document.....	5
2.2	Software Requirement Specification.....	5
2.3	Architectural and Detailed Design Documents.....	5
2.4	Source and Executable Code.....	5
2.5	Test Plan.....	5
2.5.1	Web Services.....	6
2.5.2	Integration	6
2.6	Acceptance Plan.....	6
2.7	Periodic Reporting	7
2.8	Deployment Plan	8
2.8.1	Stage One	8
2.8.2	Listing on App Marketplaces.....	8
2.8.3	Marketing	9
2.8.4	Future Releases.....	9
2.9	User and Operational Manuals	9
2.10	Risk Management Plan.....	9
2.11	Work Breakdown Structure.....	9
2.12	Test Plan.....	10
3	Software Development Process Model	10
3.1	Software Process Model.....	10
3.1.1	Unified Process Model.....	11
3.1.2	Waterfall Model	11
3.1.3	Iterative Model.....	11
4	Defect Prevention Process	12
4.1	Defect Methodology.....	12
4.2	Defect Reporting	12
5	Change Management Plan	13
6	Project Effort	13
7	Project Schedule	13
8	Developer Log.....	13
9	Documentation Formatting.....	14
10	Reflection Document (CSC650).....	15
11	Reflection Document (CSC651).....	16



1 Introduction

This purpose of this document is to identify all documentation and resources used in the entire project and where to each artifact can be found. It is geared to be used as a resource as well as a learning tool for those new to the project.

1.1 Intended Audience

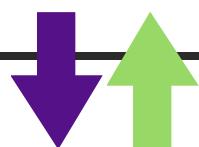
This document is intended for anyone of all technical and business backgrounds. Also this is geared for anyone of any understanding of the project.

1.2 References

- <http://www.enterim.com/docs/Example%20Software%20Development%20Plan.pdf>
- http://sce.uhcl.edu/helm/rationalunifiedprocess/process/artifact/ovu_arts.htm

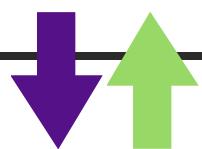
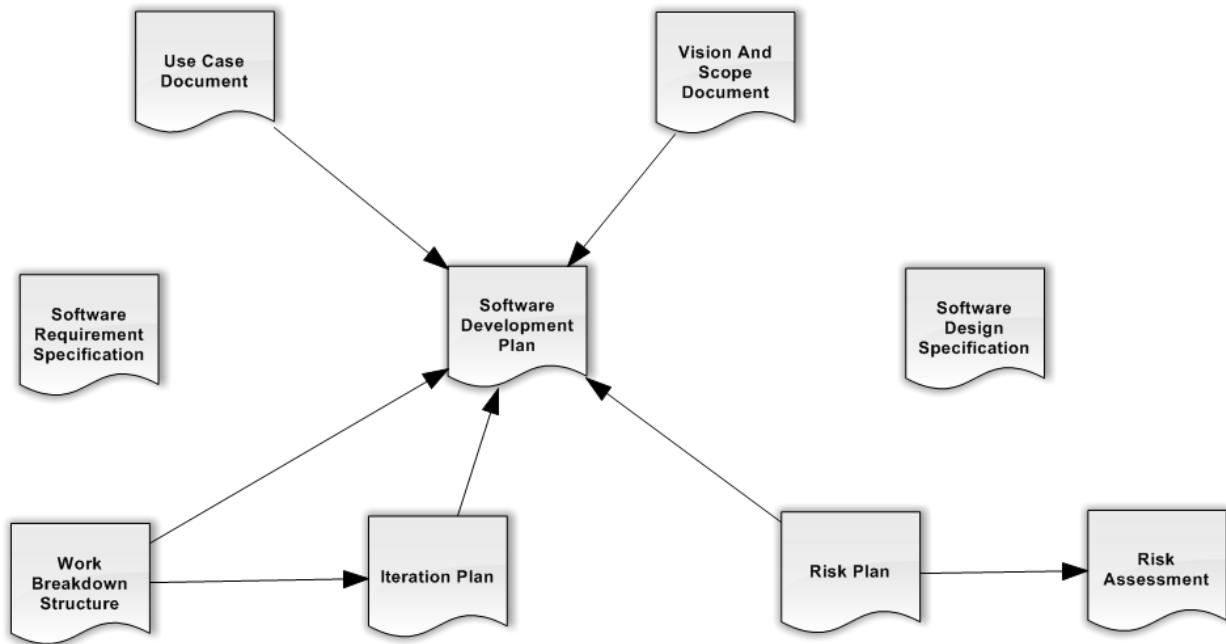
1.3 Revision History

Name	Date	Reason For Change	Version
Andy Bottom	04/29/2013	Started compiling the plan into the document	0.1
Andy Bottom	05/07/2013	Finished the finalizing of the first version. Wrote the reflection paper as well.	1.0
Andy Bottom	08/18/2013	Updated more sections and started proof reading.	1.1



2 Defining Project Artifacts and Deliverables

Below is a list of the flow of artifacts between one another.



2.1 Vision and Scope Document

Please refer to the [Vision and Scope Document](#).

2.2 Software Requirement Specification

Please refer to the [Software Requirement Specification](#).

2.3 Architectural and Detailed Design Documents

Please refer to the [Use Case Document](#) about information about the use cases of the system. Refer to the [Database Design Document](#) for further analysis of the Data and Database Design.

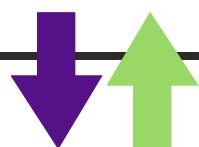
2.4 Source and Executable Code

NOTE: Will be delivered upon request in digital format.

2.5 Test Plan

Testing for the application has no separate entity to test. Technical Testing will be performed by Andy Bottom. Requirement Verification will be performed by Andy Bottom. Testing will be performed as the product is developed.

For more information please refer to the [Test Plan Document](#).



2.5.1 Web Services

The web services and the Data Access Layer reside in the web services are the most important area as it contains most of the logic in the system. Thus this will be an area where automated testing would be performed. We will use SoapUI to create tests for the data. Unfortunately load testing is outside the scope of this project.

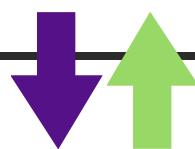
2.5.2 Integration

After large pieces of functionality are performed, integration with the other parts of the software will be done to ensure no other extraneous bugs occur.

2.6 Acceptance Plan

The acceptance testing of the phone application will be performed by Andy Bottom and also by friends of Andy Bottom. The acceptance testing for the Administrator Web Application will be performed by Andy Bottom and a couple of Andy Bottom's Friends.

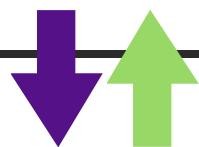
The test environment will be on several mobile phone operating systems and also through the web browsers.



2.7 Periodic Reporting

Updates to documents and reporting will be done on a reoccurring basis. Changes to any of the documentation will be logged in the revision section of the corresponding document.

Document	Action	Schedule
Software Requirement Specification	Updates and changes	Bi-Weekly
Software Development Plan	Updates and Changes	Bi-Weekly
Vision and Scope Document	Updates and Changes	Bi-Weekly
Use Cases	Additions; Updates;	Weekly
Functional Requirements	Updates or Changes	Weekly
Code	Backup	Weekly
Database	Backup	Monthly
Test Cases	Additions; Updates;	Weekly
Defect Tracking	Additions; Updates;	Daily



2.8 Deployment Plan

The following section contains the steps needed to be accomplished when the app will finally be deployed and released to the public. Please note that the Development Plan is outside the current scope of the project and will be done at a later point in time.

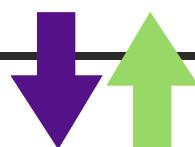
2.8.1 Stage One

- Reading of Hardware, Database and Data
- At this stage, all testing should have been completed and successful.
- Data in the Database must be cleaned up and started fresh. This is so that all the data generated during development and testing will be removed.

2.8.2 Listing on App Marketplaces

Microsoft, Android and iOS all have their own marketplaces. Each marketplace has a fee in order to list on there.

- All Licenses must be purchased to have the application listed on the stores
- Developer profile must be created
- Application must be sent to be submitted on the marketplace
- In addition, screenshots, descriptions, features and icon must be created to help market the application once it is listed.



2.8.3 Marketing

- Promotional material will be created to promote the application

2.8.4 Future Releases

Updates will be released to ensure the application remains successful

- Bug Fixes
- Feature Updates

2.9 User and Operational Manuals

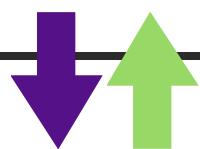
For Administrators, please see the [Admin CMS User Manual](#). For general users, please refer to the [Phone App User Manual](#).

2.10 Risk Management Plan

Please refer to the [Risk Management Plan](#) regarding information about the risks of the system.

2.11 Work Breakdown Structure

Please refer to the [Work Breakdown Structure Document](#).



2.12 Test Plan

The testing of the code will be done during the Construction and Transition phases. All code will be tested during development. After a requirement is completed coding, it will be tested to be verified that the requirement has been fulfilled by using the FIT Criteria found on the requirement. In addition, after a development iteration has been completed. An entire smoke test will be done on the entire system again to ensure no integration problems have occurred.

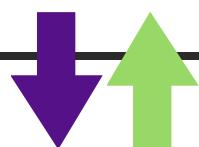
In addition to the manual testing, hopes of automated tests will be created to ease general testing of the system. Although the automated tests will be outside the scope of the project at the moment, but in the future may be a possibility.

Please refer to [the Test Plan Instruction](#) Document for more specifics about testing.

3 Software Development Process Model

3.1 Software Process Model

Analysis of which process model will be followed during the project is a very important aspect. One major disadvantage in this project is that I am the one man team in this entire project. With this, I really can only focus on one aspect of the project at once. In contrast, most processes are followed with the assumption that there are many people involved so that several areas of the project can be concurrently worked on and focused. Since I will be working on this solo, the following a Unified Process is very difficult. But nevertheless, a process must be chosen to follow.



3.1.1 Unified Process Model

I will be following a mix of models during the implementation. First off, I will be including the four major stages of the Unified Process (UP) Model. This process will contain the following categories.

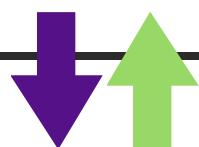
- Inception Phase
- Elaboration Phase
- Construction Phase
- Transition Phase

3.1.2 Waterfall Model

I understand that the waterfall model is very old-fashioned and non-robust to changes. However, since there is no outside client for the moment, requirement and other aspects of the project aren't anticipated to be changed. Also, the waterfall model follows a very narrow minded focus of working on one area at a time. This works well since I will be the only individual working on the project, so the single focused trait works well in this instance.

3.1.3 Iterative Model

Even though the overall workflow of the project will be single focused, I intend to break up parts of the process into iterative cycles when possible. Since there are several domains in the system, cycles to complete each domain, (requirement and development) will be done. Also, once the basic framework of the entire system is implemented and functioning, then an iterative model will be followed more thoroughly by breaking up selected features and implement them and cycle through again until all desired features of the system are implemented and tested. At this point the transition phase would be done to end the project.



4 Defect Prevention Process

4.1 Defect Methodology

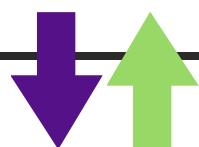
Defects will be found during development and testing. The ability to find these defects is vital to the success of the project to mitigate as many defects as possible. It is the goal to implant a defect mitigation mindset to help improve the quality of the code base and develop functionality that fulfills as closely specified by the requirements.

4.2 Defect Reporting

The success of addressing all defects is defined by the ability to track and log all defects found in the system. All automated reporting will create a log when ran to identify all failed tests. The logs would then be able to identify potential defects.

Also, a heavy presence of user testing will be needed to identify defects on all platforms and systems. A log will be created to catch any errors that occur. These logs will then be able to identify potential defects. Also, User Acceptance Testers will be able to fill out a defect card to track the bug.

Most bugs are hoped to be identified during the implementation phase. Whenever bugs are found that are unrelated to the current functionality found, a defect card will be created, identify the bug and other useful information, and then the bug will be put into the work backlog. The defects will then be fixed at the next available development time, in order of priority of the defect.



5 Change Management Plan

All changes to all documentation, diagrams, and code artifacts will be managed in the documents themselves. In each document there is a revised history table which displays all the major changes that have occurred in the document. By doing this, we can preserve and identify when the last modification occurred and result in a thorough and up-to-date project specifications.

6 Project Effort

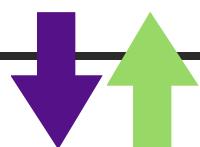
For this project, the use of effort points as a form of measurement to any given task will not be used in this project. The reason is because since there is only one person working on it, that the points will not have an impact on what gets done. Requirements will be worked on based on the system and related system it is in. Points will not determine when a requirement will get worked on. Also, due to the fact that there is a short time constraint on the development aspect of the project, and the learning curve it takes to accurately assign effort points will also pretty much render the task null and void. However, we recognize the usefulness the effort pointing provides, thus in future iterations where there would be a more team environment to accomplish tasks, the use of effort points may be reconsidered.

7 Project Schedule

Please refer to the [Iteration Plan](#) regarding schedules and the iterative process.

8 Developer Log

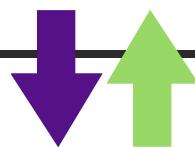
The developer log can be found at the end of the [Iteration Plan](#).



9 Documentation Formatting

The following is a list of specifications as how the documentation will be formatted. This standard will keep each document consistent with each other and improve the organization and improve readability.

- All text will be spaced with 1.5 spacing to allow for additional comments and revisions marks to be easily written on the document itself.
- Spacing in tables will be 1.5 spaced. However based on the context that the table is located, the spacing may be single spaced to allow for a cleaner transition between pages.
- Another note that the table of contents will be single spaced.
- Headings and Subheadings will follow the standard technical software documentation by beginning with numbers and decimals.
- Each document will contain very helpful links describing documentation which I felt were great resources about how to organize my documentation.
- Each page will have a cover page and table of contents.
- General text size of the body will be 11 point font.



10 Reflection Document (CSC650)

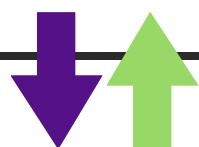
I'd like to start about reflecting back to my undergraduate Capstone which I did two years ago. It is truly remarkable of how much one can learn in just that short amount of time. Within that, I have taken many Graduate Classes where I gained a greater understanding of the entire software development life cycle. In addition to this I also gained almost two years of internship and work experience as a developer. I learned how to interact with everyone involved and so how the software development process worked in the "real world."

I truly feel that I have a very deep understanding of the software development process and can see how it is approached from different point of view including business, developer, tester, user experience developer and from the user. It is all these points of views and my experience that I brought to the table this time around when I was to work on my capstone project.

I must say that I did quite a lot of research and time on this project. The ability to run and develop my own project and product in a more professional manner was a very great opportunity. The product is on a great projection and hopefully will prove to be successful and useful to myself and others.

The hardest part of this project is easily the fact that I am essentially the only person involved with the entire product. This is evident when the usual needs that occur in software development arise as I am the one who has to do it all. This then leads to some things being put off and other things taking focus.

If I had to make a change, I would have liked to have spent the upfront time to develop my documentation a little bit earlier as it may have been easier to track changes and promote better habit of documenting, if I had a system already in place. But overall, my effort in this project has been no less than 100% and I am very proud of all that I have accomplished in it. I look forward to seeing the product come together during the summer.



11 Reflection Document (CSC651)

This past semester of the graduate capstone has certainly been filled to the brim. In this half, a majority of my time was spent focusing on the development of the phone application and the overall integration between all the pieces of the system and testing the entire flow to ensure that the base case succeeded.

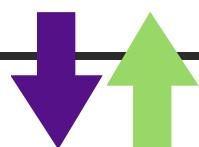
I must say that I put a lot of time and effort into the coding and the documenting of this project. I did so much that this past summer seems to have been forever. Everyday seemed eventful.

Overall I'd say that the hardest part of this portion of the project was something that I didn't expect. I put a ton of time using the application that I made to configure the Satisfaction Surveys into my application. I certainly underestimated the complexity of those surveys. However, with that fact, I am even more proud that all the designing and coding paid off because despite the new found complexity of the survey, my application sill ran successfully. It was such a reward to go through the entire process finally have my software work from start to finish.

I also, have a better understanding of the bottom-up approach. By doing this, my back-end functionality of my system was very solid, the cost was that the phone application was the last thing to develop, thus I had the least amount of time to devote to those and thus the apps weren't as finished as much as I would have preferred. But nevertheless, I feel a huge sense of accomplishment.

In addition to the coding, I feel that I certainly have learned a lot about how to do proper documentation. I actually spent a lot of time researching to find how to best create, format and organize a professional looking piece of documentation.

Overall, this Capstone Project was very ambitious and I am proud that I had the diligence to plan and document ahead and reduce the occurrence of unexpected events so that I could successfully create a great product from start to finish in a complete fashion.





VISION & SCOPE DOCUMENT

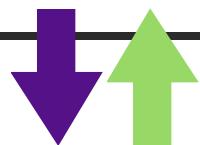
8/23/2013

Graduate Capstone



Table of Contents

1	Introduction	3
1.1	Revision History.....	3
2	Project Background.....	4
2.1	So where does this competitive edge lie?.....	4
2.2	Where Does the Data Come From?	4
2.3	Transactions.....	5
3	The Problem.....	6
3.1	So When Should I Take the Survey?	6
4	Vision Statement	7
5	Stakeholders.....	8
6	Users	8
6.1	User	8
6.2	Administrator	8
6.3	Company	8
6.4	Diagram.....	9
7	Main Features	10
7.1	Convenience.....	10
7.2	Security	11
7.3	Language.....	11
7.4	Monetization.....	12
8	Features Not Developed	13

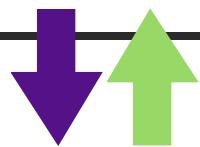


1 Introduction

The purpose of this document is to identify the vision of the software application and the business applicability it has.

1.1 Revision History

Name	Date	Reason For Change	Version
Andy Bottom	02/16/2013	Implemented most of the entire document	1.0
Andy Bottom	04/28/2013	Reformatted the document to be consistent throughout.	1.1



2 Project Background

In America, we are a fundamentally corporate world. The open marketplace where businesses compete for the business of the consumer and fight to get the edge over their competitors. Of course the goal in business is to make money. That is where the importance is on the customer. The customer equals revenue for your company and many corporations understand the fact that having a loyal customer is the best thing to have.

2.1 So where does this competitive edge lie?

Many believe that investment in knowing who your customer is and catering to their needs has enormous positive effect. That is why there has been a huge push towards spending money on analytics. By understanding the customer, the company knows exactly what the consumer wants and how they can appeal to them to reach the end goal of gaining a loyal customer and a returning source of revenue.

2.2 Where Does the Data Come From?

This is in fact the solution we are coming towards. At some point the end user needs be surveyed in order to attain the information and perform analytics which the business can thus turn into statistics and make appropriate adjustments.

If you scroll through the internet, you often can find all sorts of sites that want a user to fill some questions to then. But this doesn't necessarily target the people that get to the people who are in the stores making purchases. This is why there uses a different outlet for transactions.



2.3 Transactions

Transactions are what are involved with any purchase. A customer purchases a product from a company, and pays a certain amount of money for it. This transaction then creates a receipt for the customer to hang on to for proof of purchase. Since the transaction is the goal for the business, and the overall strive to reach the customer is growing, it has become a very common practice to use the receipt as an opportunity to be able to acquire the data directly from a consumer who is already making a purchase.

However, most customers will never willing take their valuable time to fill out a survey for a company because there isn't anything in it for them. This is why corporations will offer an incentive for the customer to fill out their survey. By filling out a survey the company will receive the valuable data to form analytics, and the customer will receive a little reward for taking the time to submit it.

The rewards can range to a variety of things, but most will fall into two categories, a contest where a customer will be entered into a raffle for a very valuable prize for a chance to win it. The other is a very small prize that they will attain. For instance, a coupon for \$2 off of something or even maybe a free item, such as a cookie from a restraint. The small prizes are 100% which is why the business needs to take into account how much the small amount prize is verses the revenue gained from the data.



3 The Problem

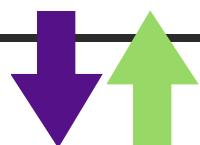
The problem arises in the actual follow through of the receipt surveys. Traditionally the way it works is that the customer makes a purchase and gets the receipt and the request to fill out the survey. The survey requires going to their website to send the information. The user arrives home and takes the receipt, goes to the computer and takes the survey and redeems the reward. They then save the receipt and the code till the next time they go to the business and can then redeem their reward.

The problem occurs because this is not how it actually works. People tend to lose their receipts because most of them aren't very valuable or worth hanging on to. Also, by the time that the get home, the time has passed of which the importance of the receipt is not worth the trouble, (aka procrastinate.) Because of this, the task become cumbersome the further away it gets that it is unlikely that it will ever get done.

3.1 So When Should I Take the Survey?

The best time to actually fill out the survey is when you have the receipt right in front of you when you finish the transaction. Thanks to the availability of the internet through the rise of smart phones and tablets, it has become very easy and almost convenient to take the survey while you are out and about. You can easily fill out the survey by going to the website. Unfortunately, most of these survey websites do not cater to the mobile phones.

This situation thus is a lose-lose for everyone. The customer has to go through all sort of obstacles to get the reward, and thus don't take the time. Because of this, the businesses lose out on very beneficial analytical data. This is the overall problem of this capstone that we will hope to solve and change to a win-win for everybody.

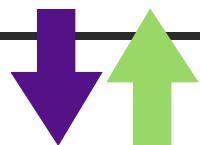


4 Vision Statement

Our solution is geared to make the receipt surveys into a win-win situation for everyone. We have researched and found that the people are most likely to fill out the survey is when they have the receipt in hand.

Since mobile smart phones have been growing dramatically in popularity and the ability to have content instantly at hand through applications is become a very large resource. This is how we intend to address the surveys.

Our goal is to create a smart phone application that will allow users to easily choose their survey and be able to have a easy to fill out interface to complete surveys and track their rewards, all in one convenient place. I have researched and found that many of the survey websites are not mobile friendly, which makes it harder to use.



5 Stakeholders

Andy Bottom is the only person working on this project thus he is the only stakeholder in this project. However in the future, this product may be used to help small businesses gather analytics, in which the small businesses would become stakeholders in the successfulness of the project.

6 Users

The basic description of how the system works is the following. The companies use the application to reach and obtain data and information from the users. The administrator ensures that the surveys and data of the application are functioning properly. The users use the application to send data to the companies in order to retrieve an award.

6.1 User

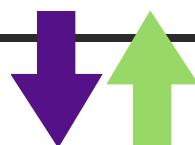
The user is the person who will be interacting with the system and features through the phone. They are who the phone application was made for to interact.

6.2 Administrator

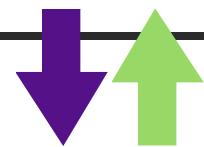
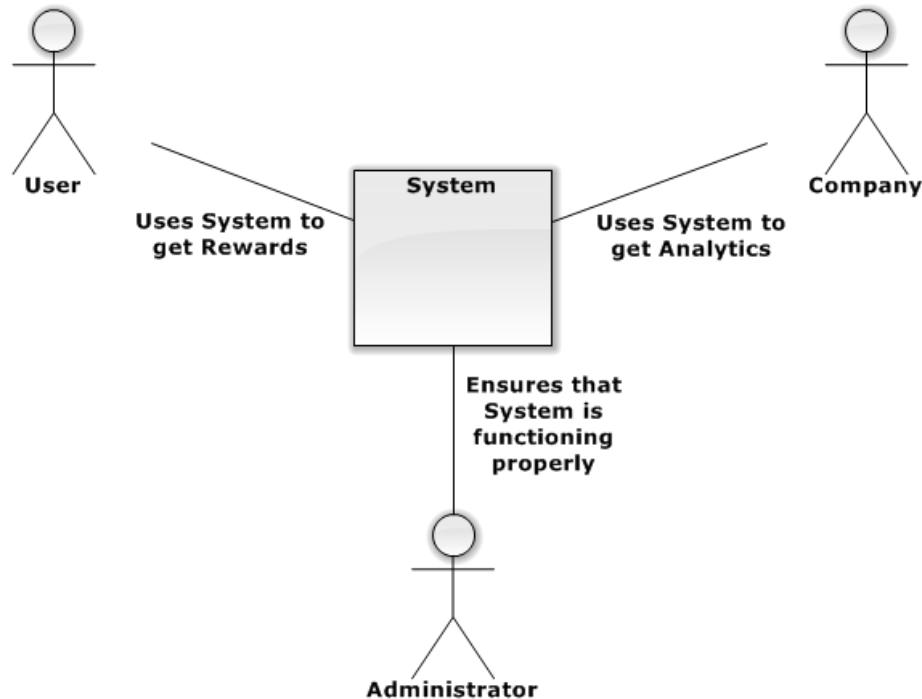
The administrator user is person who is managing the data, specifically the surveys and companies and users to keep the surveys functioning properly.

6.3 Company

The company user is who custom surveys will be created for and who gets sent the data and reports from their survey every month. The company can be both the company's survey websites, and a custom survey implemented through our product.



6.4 Diagram



7 Main Features

Below is a high level list of functionality that explains the functions of the application that would be beneficial to the project from a business point of view and are key to the success.

7.1 Convenience

This is the number one goal of the project which will make people want to use it, because it will be so much more convenient that people would have to be “dumb” to not use the app. The following ways we will do this is by.

- Easy to use interface for the mobile phones. The surveys will be outputted to the user in a nice and mobile-phone friendly view.
- When a person is logged in as a user, then we will store key information that are common answers on many surveys (such as gender) and those questions won't be displayed to the user because we already know the answer, thus providing convenience.
- Users who buy the premium feature of automatic completion will either do one of the following convenient features
 - The user will be able to run the survey and the previous answers will be submitted for that location if they say that nothing has changed.
 - The user will step through each section and determine if anything changed from that previous location
 - A user can choose to use the same answers from a different location that were similar.
- Also, the fact that they can fill out surveys for receipts to many of their favorite companies is another major feature the user will like and very convenient.



7.2 Security

Security is also another important aspect of the application

- People will not have to have a user account with our system, (they can do the very basic stuff that you can normally do through the website.) So anonymous users will be able to fill out the surveys
 - When an anonymous user is waiting for their reward, the waiting page will show the user the following message: We noticed that you are not a user yet, become a user and have access to many of these great features to make your survey next time even quicker. And it's Free"
- For surveys that send a coupon to the user's e-mail address, then we won't submit the user's e-mail, but instead have the coupon be sent to the receipt reward address and the coupon will be pulled from their and sent to the user.
- Security will be taken into account for the web services to secure any sensitive data that may be sent.

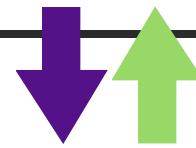
7.3 Language

- The phone will support the ability to display in many languages (probably whatever the top 10 largest languages.) This will have the competitive edge over the actual surveys that typically at most only support 1 - 3 languages. This also adds convenience.



7.4 Monetization

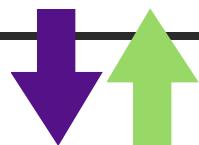
- Work with small businesses to set them up with their own custom survey in which people will use my app to fill them out. I will charge them a very low monthly fee to send them a monthly report including the analytics that they can use to improve.
- The first couple months, I will offer a free trial for the business, in exchange for having a display regarding the app for users to take their surveys. That way it will help publicize my application to get gain users.
- I will have the premium feature of automated surveys. Since this is a premium version, users will have to purchase the feature via an in-app purchase.
- Worst-case scenario, I could just have an advertising box and make a couple pennies.

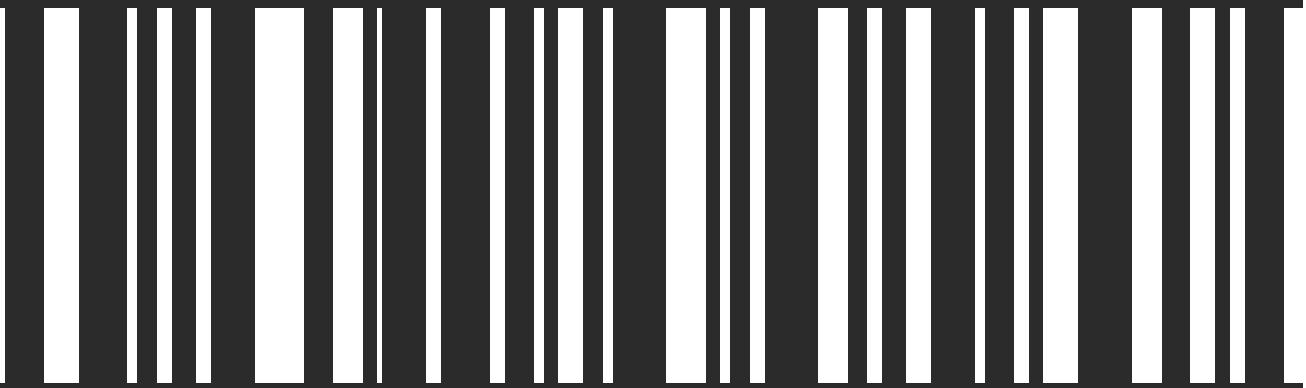


8 Features Not Developed

The following is a list of the features that will not be developed for the scope of my Capstone. They will be implemented at a later time, but due to the time constraint of this class, cannot be developed in time.

- Only the Windows Phone and basic functionality of another Phone OS will be implemented.
- All CRON jobs that would run daily checks and tests will not be implemented
- All functionality involving creating custom functionality for Small-Business will not be implemented.
- High level security precautions will not be implemented.
- User maintenance in the Admin CMS will not be implemented.
- Only a couple reward obtaining actions of the Web Driver will be implemented.
- Any monetization functionality will not be created.
- Premium Automatic Survey Submissions for the User will not be implemented
- Use of anonymous e-mail and phone submissions will not be implemented.
- User ability will not be developed for this version.
- Multiple Languages will not be developed.





ITERATION PLAN

8/23/2013

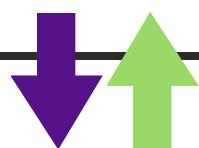
Graduate Capstone

RECEIPT

REWARDS

Table of Contents

1 Introduction.....	3
1.1 Intended Audience.....	3
1.2 References.....	3
1.3 Revision History	3
2 Plan	4
2.1 Phase Plan	4
2.2 Phase Purpose	5
2.2.1 Inception	5
2.2.2 Elaboration.....	5
2.2.3 Construction.....	5
2.2.4 Transition.....	5
3 Iteration Activities.....	6
3.1 Iteration Schedule	6
3.2 Iteration Deliverables	7
3.3 Detailed Iteration Information.....	8
4 Task Schedule.....	8
5 Resources	9
5.1 People.....	9
5.2 Hardware.....	9
5.3 Software.....	9
6 Scope	10
6.1 Basic Functionality	10
6.2 Searchable Phone Features.....	10
6.3 Users and Anonymous Users.....	10
6.4 Extra Features Scope	10
6.5 Custom Company Surveys	10
7 Iterations	11
7.1 Diagram	11
8 Developer Log.....	12



1 Introduction

The purpose of this document is to provide explanation regarding the targeted processes used in the entire scope of the project from start to finish.

1.1 Intended Audience

This document is intended for individuals with a very limited technical background and also individuals with a medium understanding of the project.

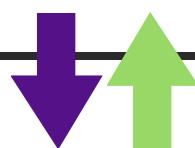
1.2 References

- http://www.isk.kth.se/~db99loer/IterationPlan1_0.doc
- http://sce.uhcl.edu/helm/rationalunifiedprocess/process/artifact/ovu_arts.htm
- http://cs.hbg.psu.edu/comp413/Iteration_Plan_Template.doc

1.3 Revision History

Managing the change history of this document will occur in this table.

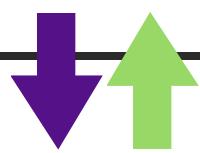
Name	Date	Reason For Change	Version
Andy Bottom	04/27/2013	Started creation of the iteration plan and transferring information into it.	0.1
Andy Bottom	05/07/2013	First finalized version of the iteration plan complete. Added the developer log into it.	1.0
Andy Bottom	08/26/2013	Finished Log and Iteration Milestone section.	2.0



2 Plan

2.1 Phase Plan

Phase	Start Date	End Date
Inception	01/26/2013	03/14/2013
Elaboration	02/16/2013	05/07/2013
Construction	02/26/2013	07/31/2013
Transition	07/16/2013	08/26/2013



2.2 Phase Purpose

2.2.1 Inception

The Inception Phase has a strong focus of the business side of the project. It determines research, background and marketplace purpose of the application. The goal for the Inception phase is to determine preliminary functionality of the system.

2.2.2 Elaboration

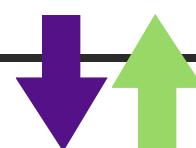
The Elaboration Phase begins more of the technical designing including analysis of specific requirements and use cases. Also, development of the system architectures and design patterns are created in this phase.

2.2.3 Construction

In the construction phase, the focus in this stage is simply the actual development of the product. It involves all the programming and testing.

2.2.4 Transition

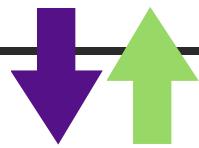
The goal for the Transition phase is to transition from the development of the project and gets it out to the users. It is the release of the product to the marketplace. Also, the future plans are determined, including updates and maintenance.



3 Iteration Activities

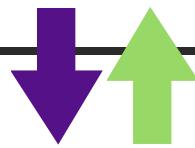
3.1 Iteration Schedule

Workflow	Phase
Research & Brainstorm	Inception;
Analysis and Design	Elaboration;
Web Services Development	Elaboration; Construction; Transition;
Web Admin Development	Elaboration; Construction; Transition;
Phone Development	Elaboration; Construction; Transition;
Testing	Construction; Transition;
Release	Transition;



3.2 Iteration Deliverables

Workflow	Deliverables
Research & Brainstorm	Vision and Scope Document;
Analysis and Design	Software Requirement Specification; Work Breakdown Structure; Use Case Documents; Risk Assessment;
Web Services Development	Web Services Source Code;
Web Admin Development	Web Admin Source Code;
Phone Development	Phone Source Code;
Testing	Test Plan;
Release	User Manual; Deployment Plan;



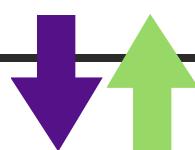
3.3 Detailed Iteration Information

The following table is the overall major development milestones that were performed of each iteration. Note that additional work, diagrams and documentation is also being done alongside the development and testing. Refer to the [Work Breakdown Structure](#) for more information on the iterations.

ID	Description
1	Brainstorm Ideas and Proposal for Project
2	Set Up Database and Tables
3	Research Web Services
4	Implement Web Services mapping. And test that they get hit.
5	Implement SQL for Web Services. Integration Test.
6	Design Structure and Implement Web Admin CMS
7	Hook up Web Admin binding to Web Services. Integration Test.
8	Create ERD of Database Objects and research Phone Structure.
9	Implement a PCL Library.
10	Implement Hello World Android App. Test with PCL. Integration Test.
11	Implement Async connection from PCL to Web Services. Integration Test.
12	Implement Windows Phone Application. Integration Test with PCL.
13	Implement Windows Phone UI Pages.
14	Implement UI Elements for Android Application
13	Implement Survey Information in PCL.
14	Create Pages for Survey. Test PCL.
15	Implement Submissions for Phone Application. Test Data into Database.
16	Performance Testing on Web Services and Phone Applications.
17	Configure Surveys. Implement Web Driver integration needed.
18	Test Complete Start to Finish of the Base Case Scenario.

4 Task Schedule

Please refer to the [Work Breakdown Structure](#) regarding task schedule.



5 Resources

5.1 People

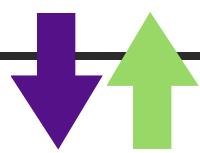
- Andy Bottom

5.2 Hardware

- Computers
- Web Server
- Database Server

5.3 Software

- Visual Studios 2012
- Office 2012
- NetBeans IDE
- SQL Server
- Windows 8
- Software Ideas Modeler 6.0



6 Scope

6.1 Basic Functionality

This is the first and largest scope aspect of the system. The basic functionality will include creating the basic ability of creating and filling out surveys.

6.2 Searchable Phone Features

Implement the ability to search for companies and their features

6.3 Users and Anonymous Users

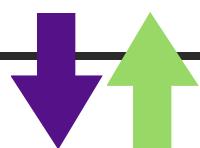
The Next scope is to implement the User aspect of the phone so they can log in and have features

6.4 Extra Features Scope

This one simply adds additional features that would enhance the experience of the app, but was missed in the other previous scopes.

6.5 Custom Company Surveys

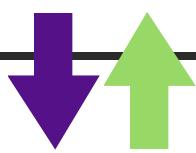
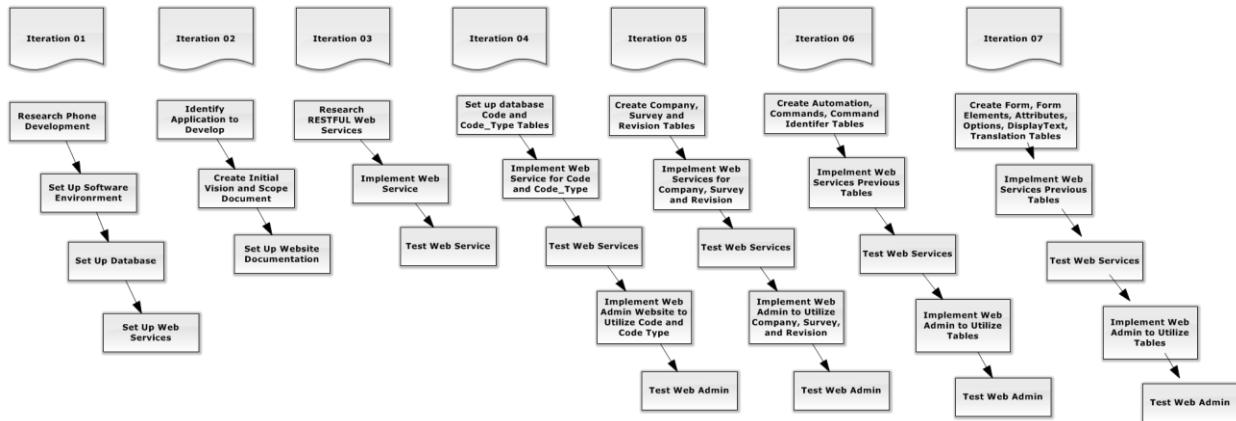
This is another large scope focusing on the ability to create custom surveys in our system designed so that small businesses can have a survey and use our system.



7 Iterations

7.1 Diagram

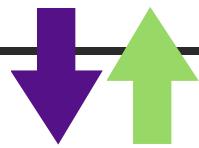
Below is a portion of a detailed work diagram for the first seven iterations.



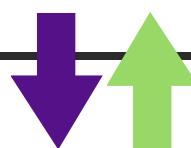
8 Developer Log

The following is the log for the project and the time tracked for it.

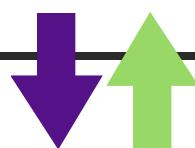
Date	Hours	Summary of Work
01/22/2013	4	First meeting with Dr. Hu. I accidentally went to the Carroll Campus instead of the Grad Center, so I was 15 minutes late. Discussed the game plan. He gave me the go ahead with my project but advice I make a small simple demo of some of the code that my project will depend on so that I can verify that it can actually be done. I'll be doing that next then.
01/24/2013	4	Created the web driver feature which will be in charge of getting kicked off and will do the automated process for the surveys. Create this little test one to ensure that it could be done, since the whole point of the app is for the automated process, so Dr. Hu suggested to test this out. I successfully made one and ensure I can continue on with the web services
01/25/2013	2.5	Created a very simple web service to ensure that it could be done. I used a SOAP web service, (which is where I had experience with before) and I set one up and kicked off the automated web driver within about 20 minutes. Super easy. Very happy with this and will continue to implement the full version of the web service.
01/30/2013	3.5	Started my attempt to create the RESTful web services. Whole time I got frustrated with them. Getting nowhere.
01/30/2013	3.5	Worked on the RESTful web services. Very frustrating, still don't understand the concept. Working with NetBeans ide. The auto-generated RESTful web services are a bunch of mumbo-jumbo. Not very helpful in making customized services.



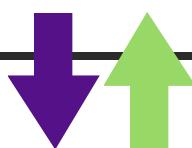
01/31/2013	4	Worked on the RESTful web services. Basically did a lot of researching and YouTubing on how to properly set up the web services. Also spent time finding examples. Kinda found a few that were helpful. No breakthrough though.
02/01/2013	4 + 3	Worked on the web services on my local machine so that it would implement the RESTful set up and also be able to hit the database. Successfully made it hit the database and accept JSON. At this point, I'm satisfied and will use this implementation of the web service.
02/05/2013	4.25	Worked on Vision and Scope document and also migrated initial web service code that was on my local box to the web server and configured it so I could hit the web service on a separate computer.
02/06/2013	3	Started on making the web admin website. Using Visual Studios 2012. I haven't programmed in .Net for about 2 years, so I'm just getting used to the IDE. Spend a lot of time at looking at the my Undergraduate Capstone project, which I may use a similar structure for the admin website. Also talked with Dr. Hu, and told him my game plan of what I've been up to. I'll have to send him the link to this website document.
02/08/2013	3	Continued to work more on the Web Admin site. I am getting used to .Net and Visual Studios again. I decided to use a similar structure to my undergrad capstone as it was a very logical approach to take and I felt it was very organized well. Will be coding the classes and everything from scratch, but the basic layout may remain the same as before. Decided on using a MVC approach, by creating my objects, and implementations. Also, I will not be using direct SQL to interact with the database, but instead be sending JSON REST requests, to perform the functionality.



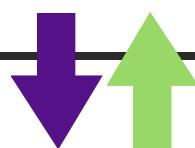
02/09/2013	3	Coded more of the admin site. I will be working on implementing one of the simpler functionalities of the system, so I am implementing the code_type and code manager pages. I managed to make the “create” and “delete” functionality to work, but didn't get update to work yet.
02/12/2013	3	Continued working on the admin web site. Was successful in getting the update functionality to work. So now I was happy that I could do each of the basic database calls. I then cleaned up some of the code.
02/14/2013	2	Decided at this point to check my database ERD diagram and determine that it logically makes sense and that I am happy with it. Made some changes to it, so I will need to update the code and code_type codes to update the fixes. I will be implementing the very basic functionality regarding the surveys.
02/16/2013	5	Worked on my documentation. Figured that the weekends will work well to work on the documentation because it's easier when I'm at home, and I'll work on the coding and implementation / initial prototype at Carroll, where I can easily boot off my hard drive and code away. Right now, I worked on the research category and posted my initial findings of the research on the companies.
02/17/2013	2+3	Worked on documenting my mobile phone research I did to the web document.
02/19/2013	4	Continued to work on the web services and implementing the database logic for the simple CRUD functionality.



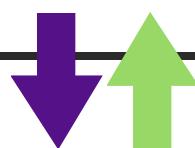
02/20/2013	4	Again I continued to work on the web services on the webserver and implementing the CRUD functionality on the Survey process table objects. As I came to creating the form_elements, I made note to rethink the interactions to possibly create a more functional relation between the elements and the element types and the attributes of each of them. Decided to make note, and will contemplate them next time.
02/22/2013	2 + 4.75	Decided to look at ERD diagram from an implementation standpoint and think through the process of how to have the most flexible database for implementation, as I have realized that some of the key form elements and attributes may be hard to identify the importance of them at this time, until I get to the implementation. I thus have removed all the original form_element_(type) tables, and will store all attributes of the form_elements in an attribute table for the form_elements. I also reworked the interaction of how a user may complete a survey and how the data will interact. After a long time of thinking of the logic, I needed a break and decided to start implementing some of the high-level objects of the web admin site (because companies and surveys aren't going to be changing all that much.) Felt good to see some results on the web admin. I implemented the company and survey manager pages on the web admin.
02/24/2013	4 + 1	Took the time on the weekend to document my database efforts that I did the previous work sessions. Decided to put the information into a Database Design Document at this time. Went through and identified each element and attribute and the purpose of the attribute and also explained the relations between tables for having concrete requirements.



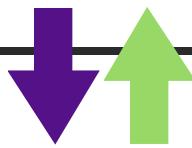
02/26/2013	4.5	Worked on the Admin Website and implemented the logic of Companies Manager, Survey Manager, and Revision Manager so that all CRUD functionality would work. The next area to tackle will be the Form and Automation Managers to work, which is a very important part of this project and pretty much the sole purpose of the admin website (which is to manage the surveys (which hinges on Forms and Automation.)
02/27/2013	2.5	Worked on the Admin Website and fixed bugs that I may have overlooked from the implementations that I made previously. All the admin website functionality is solid, so next part to code is the Automation Manager on the Web Admin.
03/04/2013	2	Took a little break to recuperate my burnout meter. Worked on documentation and preparing for Tuesday's Inception Phase Report Presentation for CSC650.
03/12/2013	4	Continued to work on the Admin website, worked on the automation manger and the commands.
03/13/2013	4	Finished implementation of the Automation Manager and the commands manager.
03/19/2013	4	Worked on the ERD diagram again to ensure that all correlations for the Form are intact. May be something to attach something so that the forms will be able to be used for reporting.
03/20/2013	4	Talked to Dr. Hu and confirmed my suspicions of needing to implement a Form Question object to attach the quantitative data of "First Name" for example that that is what is in Textbox X. Worked then on coding the Form Manger.



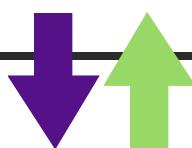
03/22/2013	2	Worked on implementing the Form Manager including the Options and Attributes fields. Came into an error with handling my JSON from my web services into parsing it into an object, because what I had been doing has just been two deep, and this was nested inside objects three deep. Couldn't figure it out so I had to stop and ponder.
03/30/2013	4	Spring break, not foreseeing me get much done this week, but I did work on some documentation. I think next presentation is April 9th, so working at documenting ideas that have been floating around my head onto paper so I can present about it.
04/02/2013	4	Completed implementation of the form manger on the admin site. I think this is the last piece that I had needed so now I can manage everything about the surveys to a basic level. I next decided to research how to go about designing my cross-platform mobile application. Looking for some sort of design pattern that is best to use.
04/05/2013	6	Basically just working on documenting lots of the information that I have already figured out, but haven't had a chance to implement. I will need to present my designs, so creating my diagrams and documentation so that I can have it written down to reference for later.
04/06/2013	4	Worked today on researching the architecture and implementation information regarding the phone application. I added my findings under the research --> phone on this web guide. Very useful information that I know have a much better understanding about the benefits and tools to use to create a successful and effective Cross-Platform Mobile Application.



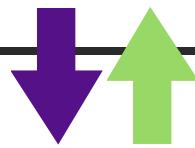
04/09/2013	4	Worked on my presentation for the elaboration presentation I had tonight. Showed off what I've done at this point. People were still skeptical that it may work, but I'm still convinced it will be fine. Otherwise, I got very positive feedback from the others.
04/11/2013	4	Worked on creating use cases and functional requirements of the system.
04/27/2013	6	Spent the day working on creating documentation to turn in for my binder. Basically I focused on creating the Use Case Document. Felt like a lot of copying and pasting because they seemed to be very closely related after a while. I also made the diagrams of the use cases.
04/29/2013	4	Spent the entire day researching and creating the documentation that I will be using in my binder. I found many great sources of which I liked how people formatted and organized the documentation.
04/30/2013	4+3	Worked again on the documentation for the binder. Basically created a lot of the technical documents. I found the hardest part was finding a nice way to organize the documentation, but even harder was determining what information to put where. Also, implemented my test web driver automation which is fully hooked up to the database.
4/31/2013	7	Spent the day finalizing for the most part all of the documentation. I feel at this point everything is relatively in good condition. I need to keep in mind that the documentation is not perfect and that I am not a professional technical writer, but I feel that it looks relatively professional.



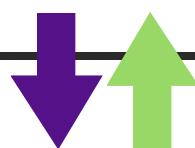
05/01/2013	7	Worked on creating my presentation for the CSC650 class. I will be using a new type of web based presentation style. It uses a jQuery file called Impress.js which is very impressive and is a great way to make presentations outside the box. However it takes a lot of skill to have an eye for everything and also I have to code it, but I think it will be worthwhile.
05/02/2013	8	Pretty much spent working on the presentation
05/03/2014	8	More coding of the presentation
05/04/2013	6	Spent the day coding the presentation
05/05/2013	9	Worked on the Presentation
05/06/2013	10	Worked on the presentation. Starting to get to crunch time, so starting to feel the pressure. The presentation is 95% done with just little tweaks that could be done.
05/07/2013	7	Crunch time, finalizing my documentation that I will turn in to my binder today. Also preparing for the presentation. Wish me luck!
05/13/2013	6	Did very intensive research on MVVM Cross framework. I think it is a very promising, but the technology I need for my dynamic surveys isn't supported yet in MVVX Cross. So I decided that Xamarin is 100% going to be the framework to develop in.
05/14/2013	3	Basically just updated and configured my IDEs so I can start working in them.
05/21/2013	3	Did some documentation.
05/23/2013	6	Worked on researching and implementing a reverse proxy for the webserver. The proxy will be a buffer between the clients and the server so that it can limit the total load on the database.



05/28/2013	6+3	<p>Started off the day had to call ITS because the Web Server was down thanks to storms this morning that cause Carroll to lose power for a couple of seconds. They got it back up though. I Completed implementation of Reverse Proxy today. And then worked on creating a Acknowledgement document where I will put credits and license info that is used in the project.</p> <p>Later, worked on attempting to get a PCL to work. Ran into problems and kept getting errors. Arg! I'll have to research the error and take another stab at it.</p>
05/29/2013	4	<p>Did research this afternoon and made a game plan as to what was causing the errors. Got the PCL to successfully load correctly. Next I coded and put logic into the PCL, (basically using the Helper Files from the Admin CMS site as a starting point for the logic. Everything seemed to work, except for the Web Service Calls. Arg! Apparently the way that I coded to use the web services, the .Net library is not available in the "portable" version of the .Net that I will need. I'll do some research on other implementations that I can try to get it to work.</p>

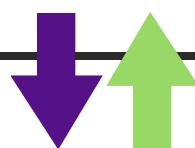


06/01/2013	6	Got out to the Graduate Center only to find that it was apparently closed. Arg! Kept my cool and headed over to Carroll University to work at the tech lab there. Worked on implementing the Portable Class Library, along with making the Web Service call work from in there, (basically made a small test.) I managed to get the Web Services to work by utilizing some the Web Service logic of the MVX platform, (pretty much looked like it was the only option at the time when using PCLs.) Then I tested the PCL by hooking it up to a very simple android application, and it successfully worked. I was very happy that the PCL works. This will be great for separating an getting a lot of reuse by using the PCL. I will next work on making the basic Windows Phone interface and try hooking that up with the PCL.
06/06/13	4	Started creating the very basic Windows Phone UI application of how I think the flow may go. Basically was learning how to work with the Windows Phone UI. Got the hang of it and how I may be organizing my windows app. And hooking up a PCL went great without problems.
06/07/13	6	Continued working on the Windows Phone App, it is going very well. I got stuck a little bit with how I will hookup the Windows Phone to the Web Services. But I figured out a way and it worked out. I actually think that I am following a form of MVVM, as I originally had wanted to. I actually deployed the application to my phone and it worked well and also hits the Web Services.

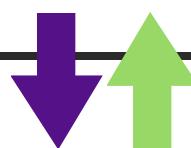




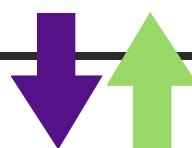
06/10/13	3	Worked a little more on the Windows Phone. Then started to work on the Android version. Basically I feel it would be a good idea to keep both versions of the app relatively close to each other to ensure that the PCL library will always be compatible. I'm a little clumsy working on the Android. Rest of the night was just spent trying to understand the ideology of the droid device, as I am not too familiar with much of the principals.
06/12/13	4	I did research and understand the main ideology and components of the Android system. I'll explain more of what I found in the UI Documentation. But basically I figured I need to get some of these navigational tools that the system will use to be built first before I start coding the individual Activities (Views). It was going ok, still stumbling through issues of how to implement drawers and the action bar.
06/13/13	5	Did research this afternoon and have some examples. I again tackled understanding the code for the drawer and the action bars. I managed to successfully implement a drawer which I am happy with, and the action bar is very good off as well. I tried to dynamically add a picture and was having problems, and implementing a viewpager wasn't showing up either. But relatively happy that I understand the android code a little bit better.
06/20/13	5	Worked more on the Android interface and functions of things. Just spent time learning and getting the drawer to work a little better. Also learning how debugging works in android.



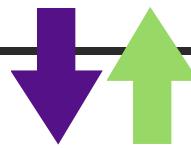
06/21/13	8	Worked more on the Android interface of things. Realized that the action bar and drawer elements that I had added is only compatible with Android versions of Ice Cream Sandwich or higher. I realize that it would be a bad decision to abandon the users of the Gingerbread release so I worked again and figuring out a different approach to implement the Action Bar and Drawer implementations so it can work on all these versions. Finally figured out that the Action bar isn't compatible. Thus I'm using a open sourced widget called LegacyBar which does work. It's a little more fiddling, but I think it will work great. Got it to work on the Gingerbread emulator now.
06/23/13	8	Worked on hooking up the Android interface with the PCL library to see that it can get data from the server. Unfortunately, the test code that I used to make the call the first time, I had accidentally written over, and so I couldn't remember how on earth I got that to work Arg!! Spent a lot of time figuring out what I did, which I eventually figured out. I also spent time learning how to use the ListView UI element of Android, which I successfully worked. A lot of time was spent looking at error messages, Googling error messages and searching what is causing the problem. But good new it works on Gingerbread and up versions. Yeah! Finally got to the point where my android UI is hooked up to the server and my PCL. I am very happy with this. I will now move back to WP and start working on the Surveys, which will be the core use case of the application. Also, I have my first summer meeting with Dr. Hu, so I'll need to put together my PowerPoint to present.
06/24/13	4	Worked on creating the surveys for the windows phone. Went very well, gonna keep on working on it.



06/25/13	4	Continued to work on the surveys to display on the Windows Phone and I got all the questions on the Culvers survey to display correctly. Yeah! This is where I wanted to be to display to Dr. Hu. I'll work on my presentation and documentation next.
06/26/13	4	Worked on creating my presentation and also updated some of the documentation. Made some realistic goals for myself to be able to mark the milestone/iterations for the rest of the project. Felt it was good. The scope of the project seems very manageable. Presented to Dr. Hu tonight and they seemed to like where I am at. I brought up the Iron Triangle which I should also present for my final presentation.
06/27/13	4	Next I worked on hooking up the surveys and save the answers from the user into a submission answer and save it. Ran into some bugs. Needs more work.
06/30/13	6	Continued to work on the ability to save the submissions answers from the user to the database. And I got it implemented and working properly and all the answers were saved to the database. Yeah. Next will be to create the have the submission kick-off the automation and then send reward to the phone. I'm getting pretty close.
07/02/13	2	Made a list of possible names for the application. Then I showed the list to my friends to get their input. Turns out that the temporary name of "Receipt Rewards" was the winner, so I will continue using that.

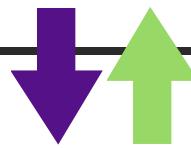


07/03/13	3	Worked on creating the logo for my application. Found several inspirations and then used my creative eye to create multiple logos. I later had my friends take a look at each logo and let me know what they thought looked the best. We came up with a winner, which I agree with. I will work on creating a color schema next, but am definitely leaning toward a purple and green for colors. (Purple feels special and green indicates money or reward.)
-----	---	Fourth of July
07/05/13	5	Wanted to implement and fix up the UI for the Windows Phone Application so I created the new images and look and feel for it using the new logo and a new color scheme. It looks pretty good I feel. I also implemented a Pivot instead of a Panorama view on all but the main page because it simply has a better feel and works well.
07/06/13	7	Worked on attempting to get a survey to fully work on the phone. So that I fill it out, and get it attached to the automation and get a response. But after a lot of putzing around I realize that most of these survey sites actually have a dynamic flow of questions (such as I mark my experience was poor, the next question will say, please explain why your experience was bad, and then continues to the main flow.) This will/is causing problem with my application because I did not take into account these variable flows, but instead made the application to handle a very static flow of things. Arg! I will have to brainstorm of how to effectively solve the problem.

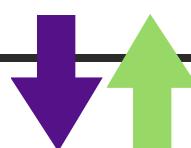




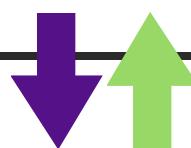
07/07/13	7	Figured out a new database design and strategy to enable the ability to use a dynamic flow for the Surveys. I will update the database document and design specification at a later day, to explain what happened. So today, I essentially had to spend refactoring the code to use the new logic. I am also using the opportunity of refactoring to clean up some things that I had wanted to change, such as adding question groupings, pass in more objects as opposed to ids to the references and also additional error checking for null values on the web services. I also wanted to preserve the work I did previously as a just in case type of fallback if this new approach ends up not working, (which I am sure it will not.) So to do that I implemented a copy of all the new tables with the ending of legacy to identify the new tables I will be hitting. That way my old code will hit the old tables. So today I implemented the database tables and the web services. Next I will update the PCL and get the Web CMS and Phone to work. This may still be a 2-3 day process.
07/08/2013	4	Worked today on getting the Web CMS to follow the improved database design. But while trying to test, I had tried to create a new Company, but it kept returning an 404 error. But I checked the database and it does create the company. So in the server logs it shows that I am trying to the URI (/company/company/create) instead of (/company/create). So I have no idea where the other company tag is being directed. Arg! I will have to Google and figure out what is happening.



07/09/2013	4.5	After Searching I found the reason of my bug from yesterday. In the new company create web service I decided to return a company object. So I would be return the object in the body, however I was missing the tag @ResponseBody, thus it was causing unexpected behavior and was forwarding to another URL instead of return the object. Made the fix and continued to work on the Web CMS, testing was going much better and things were working as they should.
07/10/2013	4.5	Worked today on the Web CMS and was able to test and created a new Culvers survey to follow the new logic. Worked then on the PCL to work with the new logic. No roadblocks; Next will work on the Windows Phone App
07/11/2013	5	Worked on the Windows Phone app today with hooking up the PCL correctly today. I changed the display of the survey to display only one question at a time. Then I had to test and it was the moment of truth. Success. The two-different work flows that I set up in the survey worked so that if I chose drive-thru it asked if the drive thru was clean, otherwise it continued on. Next week I will complete the rest of the survey and hook up the Windows Phone to correctly save the answers and then by the end of the week to be able to fill out a survey and have the automation run on submit and I can attain the reward.
07/16/2013	4	Basically wasted a complete evening because I was working on the project and I was getting a database error. I logged into the SQL Server Manager application and I could connect because of some TCP/IP error or whatever. Of course the ITS helpdesk is closed at this time. So I called the campus safety line (because the ITS message said to call Campus Safety for urgent issues.) I talked to the person and they said they will see what they could do. Took a little while



07/17/2013	4	Decided that the culvers survey may have been too much to chew to start, so I will work on the taco bell survey. So spent the evening making the Taco Bell Survey with the Admin CMS
07/18/2013	4	Realized that Taco Bell has a way to track your ip address and blocks it for 2 weeks so that you can't submit more than twice from the same computer. Worked on a way around this, but unfortunately won't be able to test until 2 weeks.
07/19/2013	8	I decided to switch to a different survey, so I switched to dunkin' donuts survey. Basically spent time creating the dunkin donuts survey in the Admin CMS.
07/22/2013	8	Spent time fixing bugs and documentation. Also worked on performance stuff a little bit and configuration of stuff.
07/24/2013	3	Found out that the Get Revision Web Service call was Timing out Past 15minutes to return from the database due to the extensive recursive attributes of it. So then I worked on creating a method to optimize the Get Revision so that would comeback faster. Tested it and it came back successfully and faster.
07/25/2013	5	After testing the new optimized revision web service, I wanted to test it with the phone. However I found out that the phone was causing a memory leak error while trying to retrieve the JSON. It turns out that the JSON file is just too big for the phone to handle. I then decided that I could solve this problem by creating alias objects with short names that would pass the actual data in. So I started with the web service to create all the alias objects and the AliasHelper Converter. I didn't complete all of it so will continue tomorrow.
07/26/2013	4	Worked on miscellaneous stuff. Fixing some bugs and documentation.
07/28/2013	3	Worked on coding the Alias code in the PCL. That was pretty much it. Didn't get a chance to test the PCL code. That will be for tomorrow.



07/29/2013 5

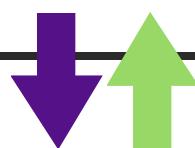
Continued to work on the aliases of testing the PCL Alias with the phone. I booted up the phone and got a connection error. I then went to the SQL Server Manager application to see what the problem was and I wasn't able to connect to it. I got the same error that I did a month ago. I called campus safety to call a ITS person to restart the server and an hour and a half later, it was reset and functioning. So after that roadblock, I tested and found out that the response is still slightly too large for the phone to make and there was a memory leak still. So then I decided to adjust the Web Server so that it won't send the Automation stuff to the phone (since that stuff isn't needed,) and it worked very well and very fast. Success! So next is to finish setting up the dunkin donuts survey and tests. Let's hope I can finally get a survey to work soon! I also fixed up a couple bugs in the Admin CMS.

@TODO

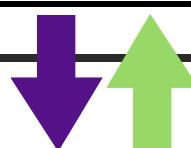
@TODO

08/05/2013 3.5

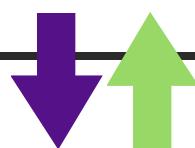
Worked on configuring the Dunkin Donuts survey question by question and performing testing on the phone side to fix any bugs that came up during the automation side of things. The goal is to get from start to finish. Getting extremely close. Hopefully tomorrow will be the day.

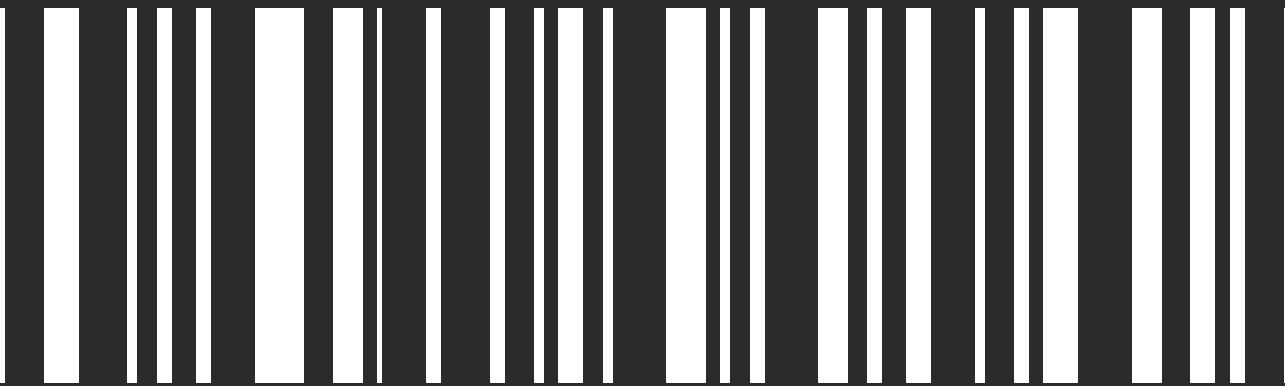


08/06/2013	6.5	<p>Finally finished up configuring the Dunkin Donuts survey so that the automation automatically went from start to finish to the rewards page which it did. Yes! Finally onto the retrieval and storage of the code. I had to update the database and web services to add the reward objects. Took some time to implement that and then performed integration tests for the rewards objects into the system.</p> <p>Worked very diligently on to complete the base success criteria of filling out a form from the phone, submit the answers and then the automation successfully iterate and retrieve and store the Reward Code.</p>
08/07/2013	2	<p>Worked on updating documents and diagrams from over that past few weeks to present to Dr. Hu.</p> <p>Had the meeting with Dr. Hu and my application worked from start to finish with perfection. Very good feeling. I felt that Dr. Hu was impressed. Considered my base success scenario to pass and now I need to focus on updating and finalizing my documentation. Dr. Hu mentioned that I should make a detailed document about all my research and findings and knowledge about the Satisfaction surveys. That was a great point, and I will get going on that.</p>
08/08/2013	4	Started working on finalization of documentation.
08/10/2013	5	Continued working on documentation. Started the survey analysis document which will contain all the business logic involved with the surveys.
08/11/2013	6	Continued documenting the surveys. Started User manual for the Web Admin CMS.
08/12/2013	3.5	Continued documenting. Updated some diagrams and fixed a couple little bugs in the Admin CMS.
08/13/2013	4	Worked more on the Survey Analysis Document on the elements of the surveys. Also started the Phone User Manual



08/15/2013	4	Moved on to the Test Plan document. Worked on getting more details on testing and forms to fill out for my integration testing.
08/19/2013	8	Documentation, working hard to get everything cleaned up and ready to go.
08/20/2013	4	Documentation, including User Manuals.
08/21/2013	8	More Documentation, mostly spend on Formatting the documentation so that everything is consistent.
08/22/2013	16	Last full day to work on the Capstone Project. Figured it be appropriate to pull an all-nighter and finish off strong.
08/23/2013	8	Finally complete. Today is the day. I am finally turning in my Capstone Project. I am truly proud of all the work I have put into it and for everything that I have learned.





SURVEY ANALYSIS DOCUMENT

8/23/2013

Graduate Capstone

RECEIPT

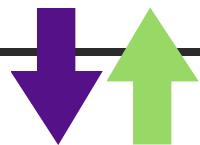
REWARDS

Table of Contents

Table of Contents	2
1 Introduction.....	4
1.1 Intended Audience.....	4
1.2 References.....	4
1.3 Revision History	4
2 Company Perspective	5
2.1 Goals as a Business.....	5
2.2 Analytics.....	6
2.2.1 Surveys	6
2.2.2 Transactions.....	7
3 Satisfaction Surveys.....	7
3.1 Websites	7
3.2 Types of Questions	7
3.2.1 Demographics	8
3.2.2 Ratings.....	8
3.2.3 Categorical.....	8
3.2.4 Locational and Time	8
3.2.5 Contact	8
3.3 Flows	8
3.3.1 Static	8
3.3.2 Dynamic	9
4 Managing the Data	10
4.1 The Goal	10
4.2 Reverse Engineering.....	10
5 Receipt Reward Surveys.....	11
5.1 Surveys.....	11
5.2 Revisions	11
5.3 Questions	13
5.4 Forms and Automations	15
5.5 Form Flows and Automation Flows	15
5.6 Form Elements	16
5.6.1 Form Element Attributes.....	16
5.6.2 Form Element Options	16
5.7 Command Elements	19
5.7.1 Identifier.....	19
5.7.2 Form Element Relation.....	20
8 Data Validation	21
8.1 Required Fields	21
8.2 Validation of Data	21
8.2.1 Numerical.....	21



8.2.2 Range	21
8.2.3 Is one of / is not one of.....	22
8.2.4 Format	22



1 Introduction

The purpose of this document is to explain and give an in depth analysis on the structure, design and make up of Company Survey Websites. By doing so, this will give a great understanding to the complexity involved with having to re-create these surveys in the system.

1.1 Intended Audience

This document is intended for persons trying to understand the fundamental business side of how survey work for this project. The goal is for those individuals to learn about the surveys and learn what is involved to recreate the surveys in the system.

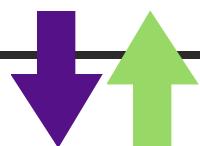
1.2 References

No references were used.

1.3 Revision History

Managing the change history of this document will occur in this table.

Name	Date	Reason For Change	Version
Andy Bottom	08/08/2013	Created the introduction and parts of the document.	0.1
Andy Bottom	08/22/2013	Finished the survey analysis.	1.0



2 Company Perspective

This section will talk about the business side of what the purposes of the surveys are.

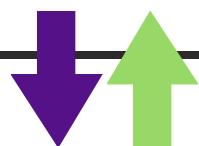
2.1 Goals as a Business

For any business, the goal is to be able to make a profit. By doing so, businesses sell something and in return are paid. Now the variety of what a company may vary greatly including, (ex. products, services, etc...) but they do sell them to their customers.

The ability of increase their sales typically depends on several factors. The first is that they have a quality product that people want and serves a need in the marketplace. However having a great product unfortunately is not all that it takes to make the sales. A bit of advertising, marketing and strategic targeting are also very important to get the word out about your company.

So how do you determine whether or not the success of the sales. This first obviously would be profitability, and ensuring that the company is actually making money. But the second is the satisfaction of the consumer.

The consumer is what you're entire business is depended on, and when it comes down to it, if you have no consumer then you can't stay in business. So the successfulness of your company is to meet the needs of your client. Their satisfaction with the product and the overall experience is what will make them come back to your business and hopefully return to and returning loyal customer and in turn, a larger customer base through word of mouth.



2.2 Analytics

The ability to target your clientele puts you at a huge advantage as a business. In fact, this is the reason many businesses deciding on investing in analytics for the purpose to do just this.

Analytics has certainly grown exponentially over the decades with newer, cheaper, accessible software has become available.

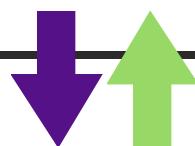
So with analytics, you perform analysis on data and recognize trends, costs, and feedback. Once this analysis, you can make changes and respond quickly to remain competitive and continue targeting your customer base.

2.2.1 Surveys

Data can be gathered from a variety of places. You can identify what products have been selling and collect the data in that way. But by doing it this way, the results will only contain data about the product. Again, going back to the original point, you could have the best product ever, but without a customer base, it won't make a sale. Reminds me of the famous quote, "If you build it, they will come," which happens to be not as true as it may seem. So, one must collect data about the customer in order to identify who the customer is and how to target them.

Typically, to collect data about the customer, you would usually create surveys to get the user to fill out. However, we have gotten very bogged down with companies trying to get our information and fill out surveys that we now feel that it is a waste of our time.

So to combat this, businesses must create an incentive for the customers to fill out surveys, thus they give out a reward for taking time to fill out the survey. This then creates the balance for the customer to fill out the survey and get the reward that they want and the user to obtain the valuable data about the customer. This is the win-win situation for all parties.



2.2.2 Transactions

During transactions, it has been commonality that with all purchases and transactions for there to be a receipt that they customer receives to show proof of purchase. Since the receipt is already given to the customer already, it has become practice to conveniently advertise their survey at the bottom of the receipts for the customer to see and then take action.

3 Satisfaction Surveys

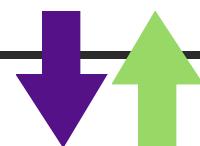
This section will break down the typical flow of the survey websites.

3.1 Websites

Originally, satisfaction and feedback forms were more of a formality that customers, if they choose, could write down feedback and submit it for the managers to be able to read and respond to if need be. There wasn't any underlying need of the business other than help the customer out and respond to their requests. But now with the analytics, businesses are proactive in asking questions in the surveys to be able to obtain analytical data from the users. In addition, with the help of technology, all the surveys and feedback forms are now online on websites to provide an easier and faster way to extract the data and turn it into meaningful reports and act accordingly.

3.2 Types of Questions

As briefly mentioned above, almost all the questions on these surveys are designed so that it can be transformed into meaningful analytical data. These are broken down into three categories: Demographics, Ratings, Categorical, Locational, Time and Contact



3.2.1 Demographics

The demographic questions involve asking about quantitative details about oneself and who they are. This way, they are able to cluster people into groups and observe any similarities or correlations in the data in this way.

Typical questions that are asked about the person may include:

- Race / Ethnicity
- Age
- Gender
- Annual Income
- Zip Code / State

3.2.2 Ratings

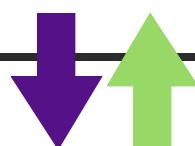
The rating questions are used to observe feelings towards something via a scale of positive or negative. The scale may also be displayed as scales including: Numerical, Stars, etc... but always represented as positive or negative.

Questions using a ratings scale typically are used to determine the point-of-view of a person about their experience and the quality of something. Typical questions may include:

- Quality of the Environment
- Quality of the Product
- Friendliness of staff
- Overall Experience

3.2.3 Categorical

The categorical questions are similar to demographic questions in that they are used to combine people into groups in hopes of seeing similarities and patterns in the responses. If there are correlations, then it can be determined that the common factor may have an impact and depending on the impact should be removed or expanded upon.



3.2.4 Locational and Time

Location questions are typically asked for of the location visited the business. Time questions involve the date of the visit and the time period that they visited the locale. Also, there may be questions on how often the person visits the business.

3.2.5 Contact

Contact information is again asking about information directly from the person, but as opposed to demographics, contact questions are a way of getting information about how to contact you and perhaps market to the person about sales, promotions or etc... However, most surveys claim that they won't use this information other than to have it.

Just to note: I've casually surveyed people and many felt uncomfortable about giving out contact information for this very reason of not wanting to get spammed by e-mails or phone calls from the company.

3.3 Flows

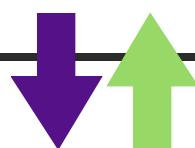
This section is used to explain how the flows of questions work.

3.3.1 Static

There are some surveys that take a very static flow of the questions. In this way, the questions are always in the same order and collected.



Figure 3.3.1-1



3.3.2 Dynamic

Dynamic flows differ from static flows in that the questions may vary depending on what responses were received from the customer.

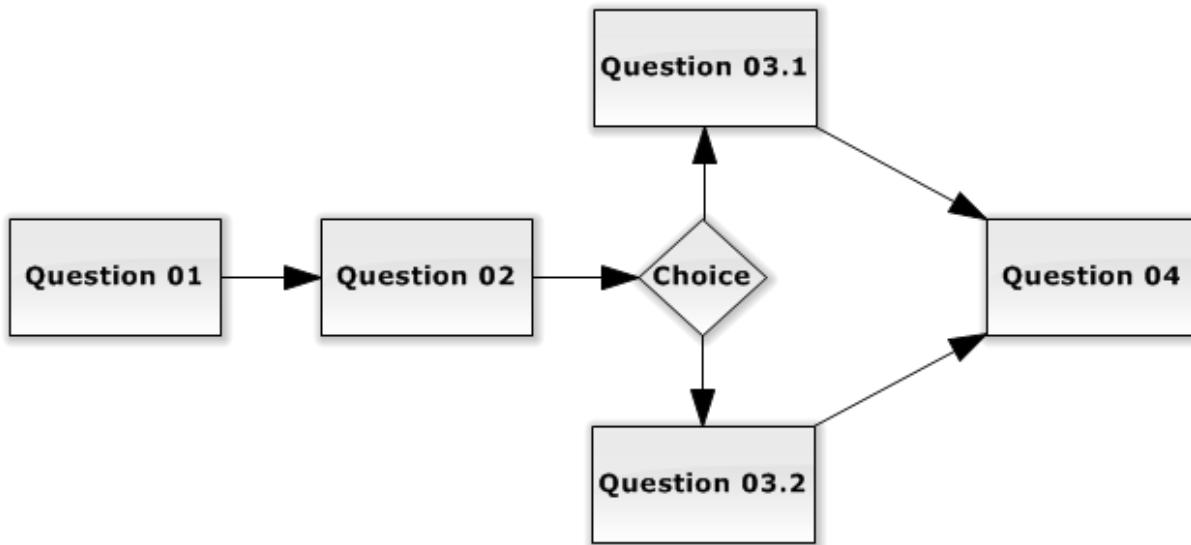


Figure 3.3.2-1

To give another example, on a restaurant survey, they may ask the question: Where did you place your order? Depending on what answer you gave, it will decide which flow you go through. As seen in *Figure 3.3.2-2*.

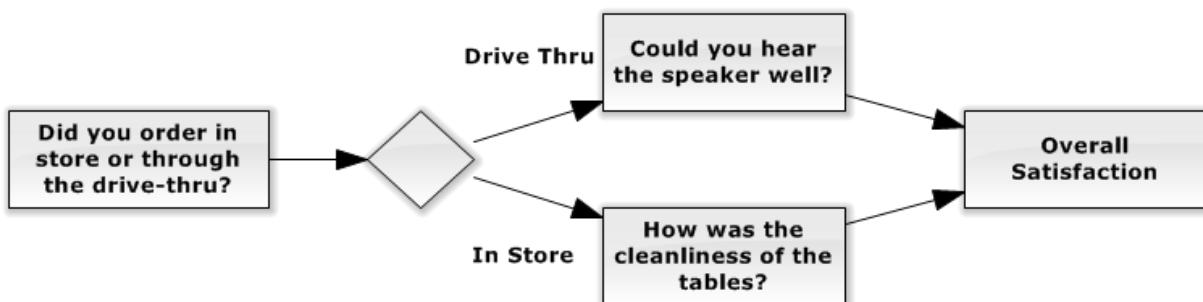
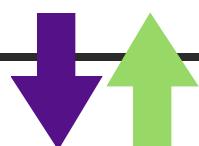


Figure 3.3.2-2



4 Managing the Data

In this section we will identify how we will get surveys from the Survey Websites into our application.

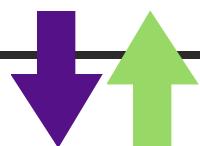
4.1 The Goal

So first, let just reiterate what the Receipt Rewards application will do. The user chooses the survey and we will display that survey to them on the phone. They fill it out and submit the answers. Then our Web Driver process will hit the Survey Website and in an automated process fill in the users answers into the form. At the end, we will obtain a reward or success message and return it back to the user. This is the base case.

4.2 Reverse Engineering

As we have stated earlier, it is clean that our entire application is completely independent from the Survey Websites. So how then do we go about adding the survey to our system?

Unfortunately there is no easy way to get the Website questions into our application, so thus a manual way needed to be created in order to create the surveys in our system. By doing this we will have a Web CMS site that an Administrator will use in order to recreate the survey.



5 Receipt Reward Surveys

In this section, we will discuss in depth the makeup of all the surveys. This section is extremely important in understanding all the relations involved with Surveys.

To view the realizations of the Objects that are being discussed, please see the [Database Diagram Document](#) and the [Software Design Document](#) for more details.

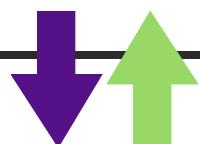
5.1 Surveys

All companies will have a survey attached to them. These surveys are the Satisfaction Surveys that our application is aimed to aid in completion for the user.

5.2 Revisions

Revisions are the next level under surveys. These represent the version of a survey. Most importantly, a revision creates the separation between the questions, forms and commands with the actual high-level Survey Object. This is very important for the following reason: If a company decides to make changes to their Satisfaction Survey Website, they may add, update or remove a question or reorder something. By doing this, all references of that previous version of the survey may have updated all the elements and references of the elements ids could have changed, thus rendering our Web Driver Automation completely useless.

So whenever a survey has changed, we must make a new revision of that survey and recreate that survey for the new version. The reason that we create a new revision and not just update the existing revision is because that revision may have already had form_submissions associated to that revision. If we made changes to the revision, then that would create orphaned data of the form_submissions, and thus becomes problematic. Thus the new revision object leads itself to better data quality in the database by keeping preexisting relations intact. See Figure 5.2-1 for details.



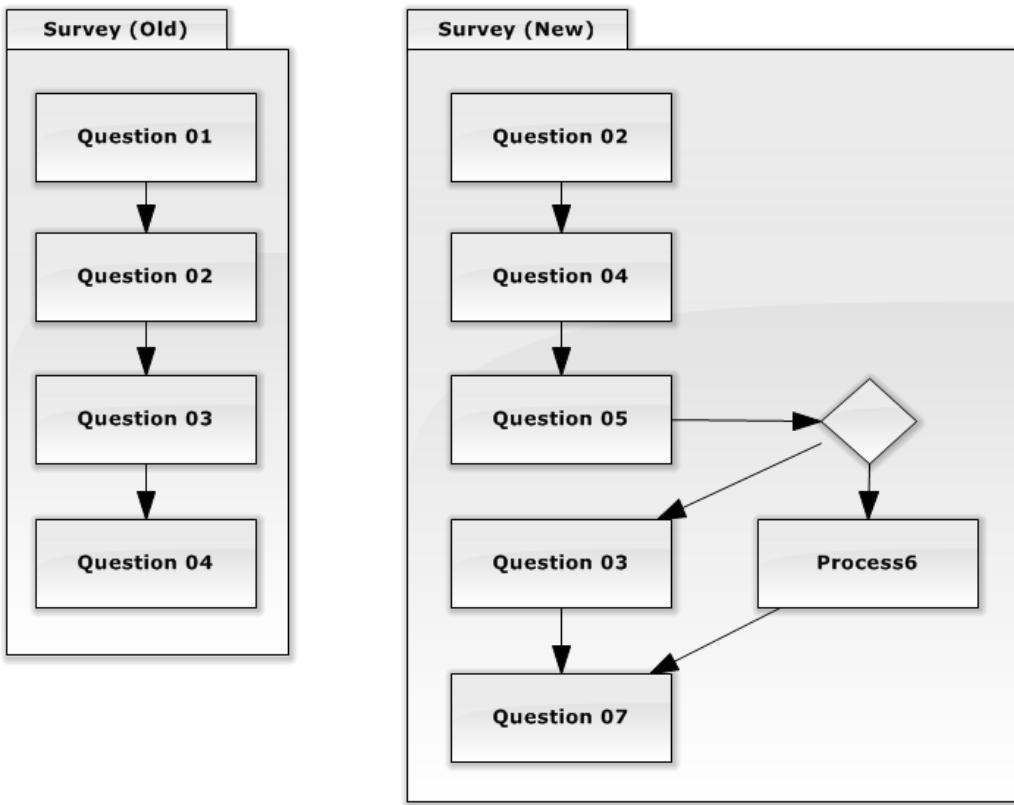
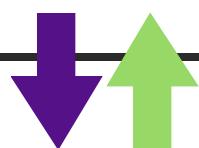


Figure 5.2-1

For example, let's say that a company changes their survey. The original version is Survey Old in the diagram above. However, the new version is Survey New. As you can see, they removed Question 01, switched the order of Question 03 and Question 04. They also added Question 07 and Question 05. On Question 05, they added a multiple flow to Question 06.

Now the revision in our system would have looked exactly like Survey Old. In addition, all the submissions that were made had Question01 -04 as well. However, if we simply edited the revision to look like Survey New, then all the Submissions in our system would be rendered useless and wouldn't be able to run since they don't contain any new questions. Nor could we look at how it was changed since the Revision has been edited and no way to look at how it was. This is why we create a new revision whenever a company changes a survey so that we can preserve the historical information about the revisions.



5.3 Questions

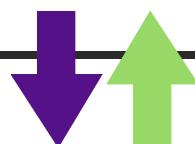
Questions act independently as their own entity. The questions purpose is simply a container which holds one distinct form and distinct automation list. Questions provide the logical connection grouping between the form and automation. This grouping is needed because it allows for an easier logical understanding between the form and automation.

On surveys, the Question is a single question on a survey. Thus the form of the Question will contain the elements of that questions and the automation will have the commands needed to fill in that question during the automation process. Thus the conceptual link is made by putting both the form and automation inside a Question.

As stated earlier, there is a need to be able to have multiple flows for the questions. Depending on what the user answers may then lead to a different set of questions being asked. Thus the standard single relation would not be flexible enough to handle these types of associations. In addition to this, questions are always made up of two distinct halves.

Questions are referenced in two different contexts, in the Form displaying context and the Automation process. This is the key in determining the flow that the question will take.

So, the idea behind how the Question associations are follows the Linked List pattern; where every Question is a node. The node knows where the possible relations are that it goes to next. These associations are identified in the form_flow and automation_flow tables. Because we stated that the flow depends on which the context the Question is in. If it is in the form context, then the questions will look at the form_flow and determine which Question will be next, and likewise for the automation. The Flows are defined below, but look at the example diagrams to get a better idea.



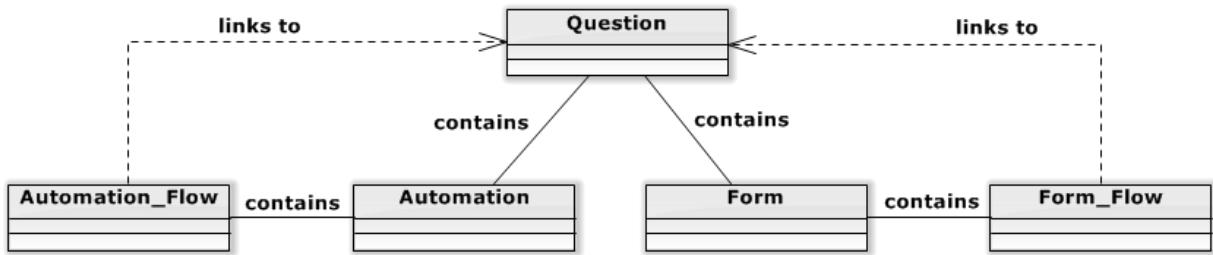


Figure 5.3-1

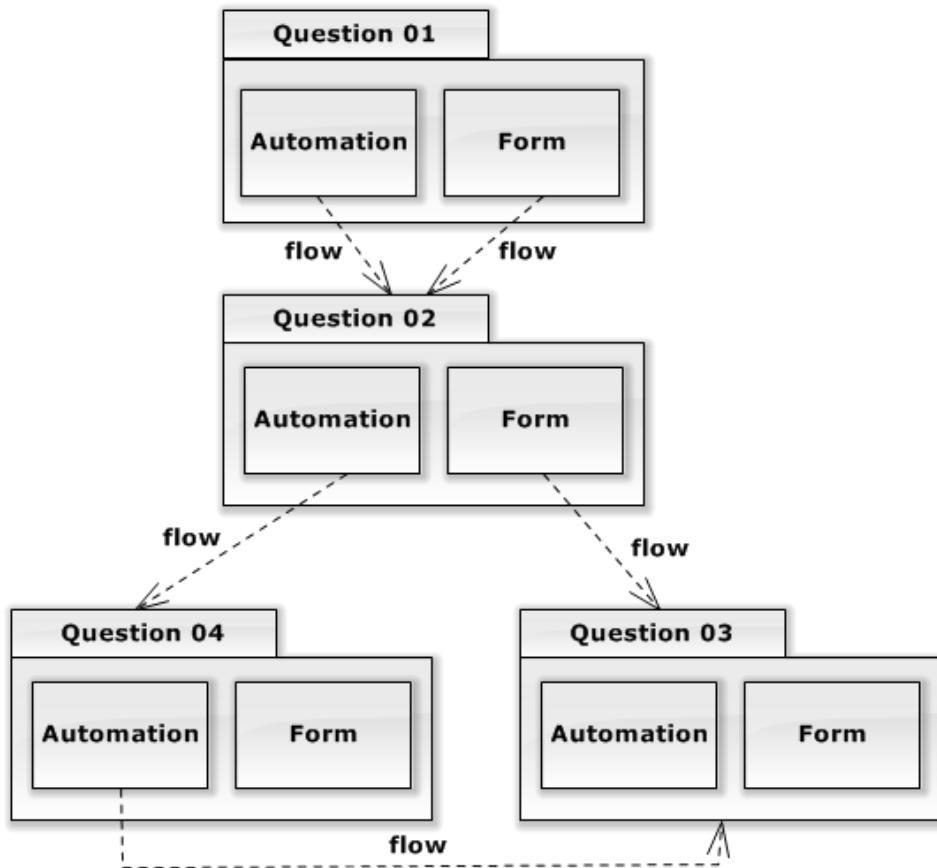
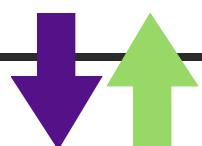


Figure 5.3-2



5.4 Forms and Automations

Forms are very simple in their concept in that they are simply a list of form_elements. Also, forms contain the form_flow associations. The form is used to display the elements out to the phone application of the user so that they can fill out the survey.

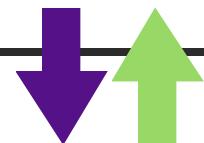
Automations are the counterpart to the form in that they contain a list of commands that will be ran during the Web Driver Automation part of the application. The Automation also contains the associations of the Automation_Flow.

5.5 Form Flows and Automation Flows

The way that the form flows and the automation flows act are identical except that they are only ran in different contexts, the form and automation contexts respectively.

The flows can contain one or more relations to the next value. Typically one of the flows will be a default and the other flows will be taking instead if the user inputs a value that would cause the event to take place based on the value inputted. This is how the form_flow works.

The automation flow takes on a different behavior. Basically, it prioritizes all the possible flows. When the action occurs where we need to know whether or not a flow should be taken, it goes by the priority. It checks to see if the question that that flow would take is present in the form_submission object. If there was an answer submitted from the submission, then that means that that question was displayed to the user on the form side and thus the automation process should be performed. If it was not fine, then that means that the question wasn't displayed, and the automation flow skips to the next prioritized flow option.



5.6 Form Elements

Form Elements are added representations of the elements that will be appearing on the Phone of the user for a particular question. The elements can scale dynamically just by adding the code and the logic on the phone application.

Currently there are several types of elements that can be added including:

- Labels
- Textareas
- Sliding Bars
- Textboxes
- Radio Buttons
- Dropdown Boxes

Form Elements contain the following properties, Form Element Attributes and Form Element Options, which will be discussed in more detail next.

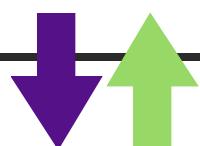
5.6.1 Form Element Attributes

Form Element attributes are essentially properties that are to be added to the element to give the element certain behaviors when it is displayed. The purpose of this is needed when it comes to needs of how it will appear on the phone, if it dynamically responds to events, and most importantly data validation.

5.6.2 Form Element Options

Form Element Options provide a very unique ability to the form elements. In fact, depending on what the element is, the way that the option is used will change. However, the common thread is that they all involve displaying a value out to the user.

Form Element Options contain two properties, the Display_Text which is the text displayed out to the user and the other is the value.



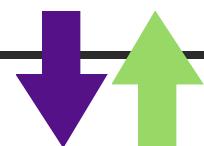
5.6.2.1 Option Value

The value property contains two very distinct possible values. The first type is a text value. Essentially, this is the property that is typed in, such as “YES”. The other is a very special case and that is a reference value. The reference value will contain the Command Element Identification Value inside the Option Value so that it indicates which value it will find.

This value should be formatted however the Identification Type is defined. For instance, if the identification type is XPATH, then the Option value must contain an XPath. If it is IDENTIFICATION, then the option value should be the Id of the Element.

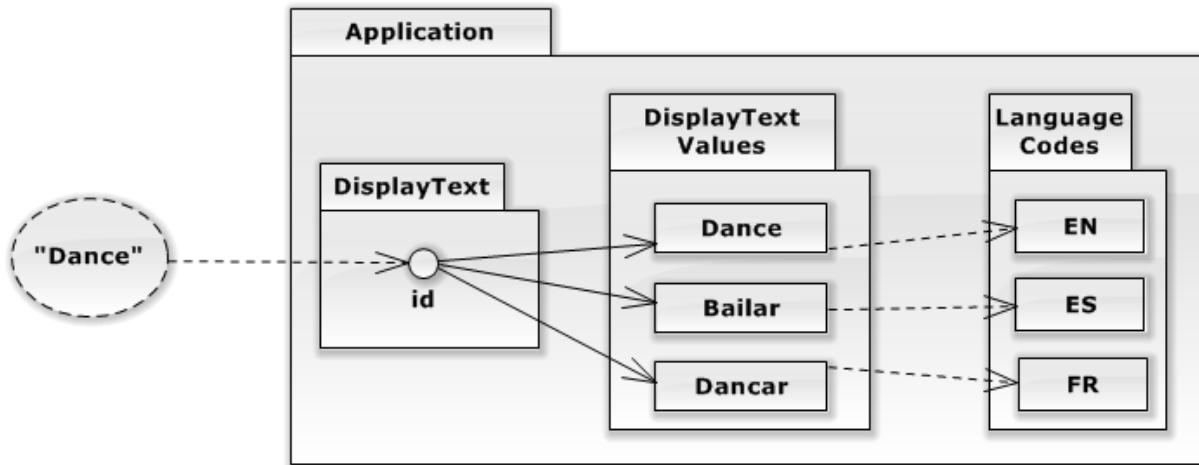
Below is a table explaining what elements follow what pattern:

Element Type	Option Value Behavior
Label	None
Textbox; Textarea;	Text Value
Radio Button; Check Box; Dropdown;	Reference Value
Slider;	

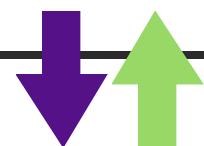


5.6.2.2 Display Text

One of the main reasons for separating out the displayed information of the forms is that it gives us the flexibility to provide the feature of using multiple languages in our forms, which as defined in the **Vision and Scope Document** is a huge advantage as it increases the audience base.



Essentially, the way that the Display Text will be stored is through references. First the text is submitted to the Administrator, and then a new Display Text entry is made. That is then translated into X number of languages. For each translation, the value gets inputted into the Display Text Translation field. Thus when a user wants to see a text in a different language, the language code will be sent in the request. The database will then look at the referenced Display Text and find the translation with the correct language code. Thus, this provides the support for dynamic multi-language support.



5.7 Command Elements

Command elements are the counter part to the form elements. The Command Elements are the commands that are performed during the automation of a question.

Command Elements also have two properties, Command Element Identifier and Command Form Element Relation.

5.7.1 Identifier

The identifier of the command element is the reference as to how the Web Driver will locate the element on the page in order to perform the action on it.

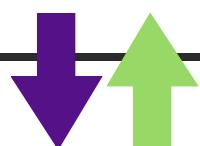
There are several ways to identify a command element:

- Identification (ID)
- Form Element Name (NAME)
- CSS Class (CLASS)
- XML Path Language (XPATH)

In addition to the identification of the element, the identifier can also contain properties that add behavior to the web driver. The following are the types of behavior

- SKIPPABLE - If the web driver does not immediately find the element, then it will skip over to the next element. The default behavior is that if it cannot find an element prior to the timeout, then it will throw an error and end the automation.
- DEPENDS_ON - is used for elements that depend on SKIPPABLE command_elements. If an element has a Command_Element_Id referenced in the value of the DEPENDS_ON method, and that Command_Element which is SKIPPABLE is skipped, then the DEPENDS_ON Command_Element will not be ran and also skipped over.

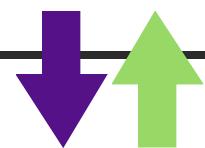
Note: That you must list the identifier FIRST and any additional properties should come after.



5.7.2 Form Element Relation

The form element relation property is relatively straight forward. This property is needed for any element that relies on the answer of the user. In this case, the Command Element will create a relation to a Form Element so that it identifies that that is the form_element to use for the value.

During the web driver process, when the command is ran, it will look through the submission answers for the same form_element_id and take the value of that and use it in accordance to that command_type.



8 Data Validation

Currently, the data validation is not implemented at this time for the element, but it is important to address what actions it will need to entail.

8.1 Required Fields

Often times, surveys require certain fields to be required, or not left empty. This means that the field is required. When a field is required, then the user cannot continue with the survey until the element contains a value.

8.2 Validation of Data

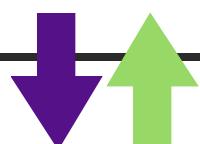
The Validation of the Data is the next aspect needed for the elements. Many times, on surveys, they contain their own validation of value, especially for input values such as textboxes. Because of this, we need to enforce our users to follow the same type of validation by adding these same rules to our application.

8.2.1 Numerical

Numerical fields are fields where the user is capable of inputting any value into a textbox, but we can only accept numerical values. This attribute would ensure that the value is a number or integer.

8.2.2 Range

This is also applied to a numerical field, but it determines that the value fits inside a specified range.



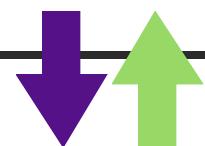
8.2.3 Is one of / is not one of

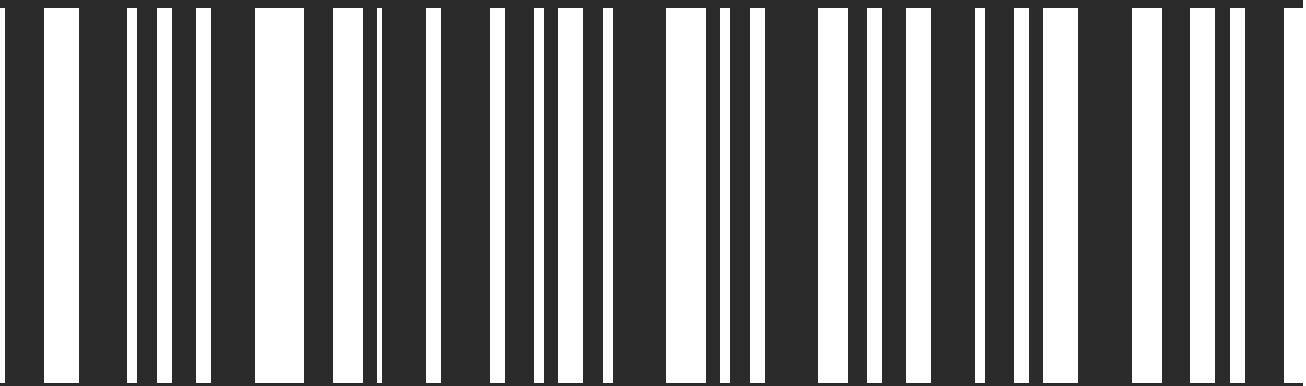
This type of validation will have the admin specify certain values which are “acceptable” or “not acceptable”. Then the field will ensure that the value that was given follows the specified lists.

8.2.4 Format

The ability to determine that a value follows a certain format is critical. It is often used in instances of validating: Dates, Phone Numbers, Social Security, Zip Codes, etc.

The format validation will except a regex pattern that the element will then use to compare that the value given is passes the formula of the regex.





SOFTWARE REQUIREMENT SPECIFICATION

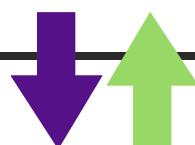
8/23/2013

Graduate Capstone

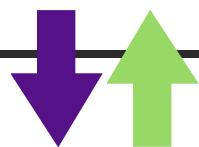


Table of Contents

Table of Contents	2
1 Introduction.....	4
1.1 Intended Audience.....	4
1.2 References.....	4
1.3 Revision History	4
2 Overall Description	5
2.1 Product Perspective	5
2.2 Product Functions	5
2.2.1 Web Services.....	5
2.2.2 Automation.....	5
2.2.3 Admin Back-Office Application	5
2.2.4 Phone Application.....	5
2.3 User Characteristics.....	6
2.4 General Constraints	6
2.2.1 Hardware Constraints	6
2.2.2 Software Constraints.....	6
2.2.2.1 Local Caching.....	6
2.2.2.2 Miscellaneous.....	6
2.2.3 Other Constraints	7
2.2.3.1 Company Survey Websites.....	7
2.5 Assumptions and Dependencies	7
2.5.1 Assumptions	7
2.5.2 Dependencies	8
2.5.2.1 Hardware.....	8
2.5.2.2 Software	8
3 Specific Requirements.....	9
3.0 Definitions	9
3.0.1 Types.....	9
3.1.2 Categories	9
3.1 External Interfaces	10
REQ-03: Anonymous Sign In Page	10
REQ-05: Sign In Link Property	10
REQ-06: Any User Can Browse	10
REQ-07: Any User Can Search.....	11
REQ-08: Main Menu Functionality	11
3.2 Functions.....	11
REQ-01: Any User Can Use Application	11
REQ-02: User Cookie Created	12
REQ-04: Users Can Sign Out	12



REQ-09: Receive Take Survey.....	12
REQ-10: Answer Question	13
3.3 Performance Requirements	13
REQ-18: Response Time of Automation Run.....	13
REQ-19: Automation Work Load.....	14
3.4 Logical Database Requirements	14
REQ-17: Active Revisions.....	14
3.5 Design Constraints	14
3.6 Software System Attributes	15
3.6.1 Reliability	15
3.6.2 Availability.....	15
FR-14: Web Service Down-Time.....	15
3.6.3 Security.....	16
FR-15: Web Service Request Security	16
FR-16: Encrypted Requests and Responses.....	16
3.6.4 Maintainability	16
FR-11: Web Admin Data.....	16
FR-12: Test Automation CRON Job	17
FR-13: Test Automation CRON Job Failure	17



1 Introduction

The purpose of this document is to provide a repository and description of all the requirements found in the system of the project. The document will be used by the development team to develop the functionality. Clients can also use this to verify the agreed upon functionality.

1.1 Intended Audience

This document is intended for individuals with a medium-high technical or business background.

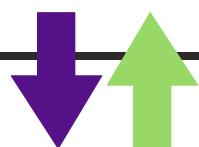
1.2 References

- http://www.env.gov.bc.ca/csd/imb/3star/sdlc/3analysis/Software_Requirements_Specification_Standards.doc

1.3 Revision History

Managing the change history of this document will occur in this table.

Name	Date	Reason For Change	Version
Andy Bottom	04/05/2013	Began compiling information into SRS document;	0.1
Andy Bottom	04/16/2013	Continued to work a tad more in the document;	0.15
Andy Bottom	04/28/2013	Continued funneling information into the document;	0.2
Andy Bottom	05/07/2013	Finished finalizing the first version.	1.0



2 Overall Description

2.1 Product Perspective

The following are the requirements of the entire system. Even though the system itself contains several whole models including: Phone, Web Services, Database, Automation and Back-Office, all requirements will be specified in this entire document.

2.2 Product Functions

The following is a list of the high-level and more functions of each system.

2.2.1 Web Services

- Used to handle all database requests
- Utilized by all external interfaces (phone and web admin.)
- Kicks off the automated process.

2.2.2 Automation

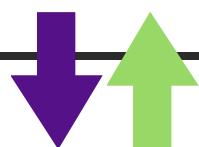
- Web Driver to submit the user submissions into the Company Survey Websites.

2.2.3 Admin Back-Office Application

- Manage all data in the system and database

2.2.4 Phone Application

- Interfaces with the user for them to utilize the functionality of the system.



2.3 User Characteristics

- Administrators must be somewhat technical understanding of the system
- Users of the Phone Application must have a working knowledge of how to use Smart Phone Applications. Also, have an understanding of the purpose of the application.

2.4 General Constraints

The anticipated operating environment will be on a mobile device. Since this will be an application, the goal is to make the application available to the companies that already have the largest marketshares. Currently, we will be aiming to having our application developed for iOS, Android, and Windows Phone.

2.2.1 Hardware Constraints

- Phone Screen Size
- GPS

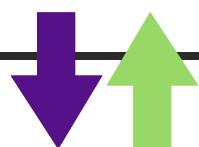
2.2.2 Software Constraints

2.2.2.1 Local Caching

Caching is going to be used on the phones and the Web Services so that data can be sent quicker and relieve the need for constantly hitting the database or the Web Services. Depending on the hardware phone OS.

2.2.2.2 Miscellaneous

- Phone OS
- Xamarin Framework



2.2.3 Other Constraints

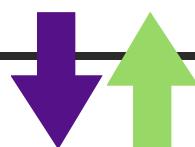
2.2.3.1 Company Survey Websites

We pretty much are at the mercy of how well we can utilize the surveys on the websites. If a company changes a survey such as questions, application will fail and need to have a new revision made for that survey so that the automated process can continue.

2.5 Assumptions and Dependencies

2.5.1 Assumptions

- We are making the assumption that the load of the application will not surpass one individual per minute. We are making this assumption because there is only one server that is handling all the work. The assumption once the application has passed testing and will be released that the load will increase significantly.
- Those users of the Admin CMS have a medium-high technical background.



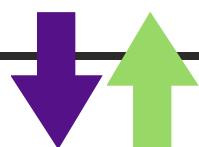
2.5.2 Dependencies

2.5.2.1 Hardware

- I will be using Windows 8 OS to perform most of the coding and testing on. I have it stored on an external hard drive that is bootable for the Carroll Computers. That way my settings and programs will be already configured each time I start up. Also, if time permits, a Macintosh computer will be needed to code and test the iOS app.
- I will be testing my phone application on my Windows 7.5 phone, and on my families Android Phones. If time permits, a iPhone will be used to test the iOS app.
- A Microsoft SQL Server Database will be used to store the data. The server is stored on the Carroll Network, provided for the CSC students.
- The Web Services will be hosted on a Java Server hosted on the Carroll Network provided for the CSC Students.

2.5.2.2 Software

- I will be programming the Windows Phone and Android Application in Visual Studio 2012. If time permits, another Mac IDE will be used to code the iOS App.
- For testing the applications, I will be using a Windows Phone, Android and iOS phone emulators.
- For coding the Web Services and Web Driver, I will be coding in NetBeans IDE.
- Tomcat will be running the Web Services.
- Microsoft SQL Server will be the database software.
- To test the Web Service, SoapUI will be used.
- Web Browsers will be used to use the Admin Website.
- Software Ideas Modeler will be the software used to make the diagrams.



3 Specific Requirements

3.0 Definitions

3.0.1 Types

F – Functional

NF – Non Functional

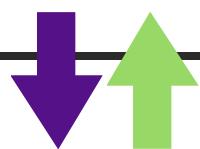
3.1.2 Categories

SECURITY – Required Security Features

INTERFACE – Required User Interface

CONFIGURATION – Required Configurable Behavior

CORE – Required Core Features of the System



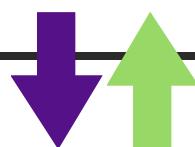
3.1 External Interfaces

This section contains requirements relating to the inputs and outputs of the system. Specifically, relating to the Web Admin and the Phone Applications.

Functional Requirement ID	REQ-03: Anonymous Sign In Page
Description	Only an anonymous user can visit the Sign In page and log in. Users that are already signed in will not be allowed to view this page.
Origin	Log In Functionality
Priority	High
FIT Criterion	Anonymous Users can log in and Logged In users cannot.
Type	FUNCTIONAL
Category	CORE

Functional Requirement ID	REQ-05: Sign In Link Property
Description	When an anonymous user is using the application, they will see the sign in link. However, when a Logged in User uses the application, they will see the logout link.
Origin	Log In Functionality
Priority	High
FIT Criterion	Anonymous Users can only see Log In; Logged In Users can only see Log Out
Type	FUNCTIONAL
Category	INTERFACE

Functional Requirement ID	REQ-06: Any User Can Browse
Description	Any User can use the application and browse for companies
Origin	Browsing Functionality
Priority	Medium
FIT Criterion	Users can visit the browse company pages.
Type	FUNCTIONAL
Category	INTERFACE
Reference	UC-04;





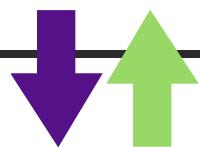
Functional Requirement ID		REQ-07: Any User Can Search
Description		Any User can user the search functionality of the application
Origin		Browsing Functionality
Priority		Medium
FIT Criterion		Users can use the search page.
Type		FUNCTIONAL
Category		INTERFACE

Functional Requirement ID		REQ-08: Main Menu Functionality
Description		A Main Menu will be used to help the user navigate between different screens of the phone application
Origin		Phone Functionality
Priority		Medium
FIT Criterion		A Main Menu is created for the user to use and navigate
Type		FUNCTIONAL
Category		INTERFACE
Reference		UC-01;

3.2 Functions

These describe the fundamental actions that take place for processing the inputs and outputs.

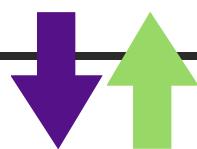
Functional Requirement ID		REQ-01: Any User Can Use Application
Description		A user can use the application without needing to log in.
Origin		Log In Functionality
Priority		High
FIT Criterion		That a user will be able to anonymously fill out a survey and receive the reward without having to log in.
Type		FUNCTIONAL
Category		CORE



Functional Requirement ID	REQ-02: User Cookie Created
Description	When a user logs in, a cookie will be created on the phone to store their logged in user information.
Origin	Log In Functionality
Priority	High
FIT Criterion	When user logs in, they will be shown as being logged in via the cookie that is stored
Type	FUNCTIONAL
Category	CORE
Reference	UC-02;

Functional Requirement ID	REQ-04: Users Can Sign Out
Description	Only signed in users can sign out. Anonymous Users cannot.
Origin	Log In Functionality
Priority	High
FIT Criterion	Logged In Users can sign out.
Type	FUNCTIONAL
Category	CORE
Reference	UC-03;

Functional Requirement ID	REQ-09: Receive Take Survey
Description	When User Takes Survey, the first question should be displayed to the user.
Origin	Browsing Functionality
Priority	Medium
FIT Criterion	Users can use the search page.
Type	FUNCTIONAL
Category	INTERFACE
Reference	UC-07;

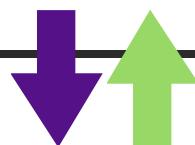


Functional Requirement ID REQ-10: Answer Question	
Description	When Upon a User Answers a Question from a Survey, the phone application shall perform any required validation on the form element that may be tied to the field. If it fails, then it won't make a request to the Web Service;
Origin Priority FIT Criterion	Take Survey High Inputting an incorrect value for a form_element that has validation associated to it shall display an error message to the user about the value; The user will not be able to continue to the next question if a required validation specific form element has an invalid field in it.
Type Category Reference	FUNCTIONAL CORE; UC-08;

3.3 Performance Requirements

Contain requirements specifying numerical requirements placed on the software system.

Functional Requirement ID REQ-18: Response Time of Automation Run	
Description	The response time of an automation process to run and return with a response shall not exceed 15 seconds.
Origin Priority FIT Criterion	Form Answer Submission; LOW; Ensure that the duration of a answer submission of a form does not exceed 15seconds from the phone interface.
Type Category	NON-FUNCTIONAL; CORE;



Functional Requirement ID		REQ-19: Automation Work Load
Description	The automation should response within 15 seconds with 10 simultaneous calls	
Origin	Web Driver	
Priority	Medium	
FIT Criterion	When there is 10 simultaneous calls, the automated process should return within 15 seconds.	
Type	FUNCTIONAL	
Category	CONFIGURATION	

3.4 Logical Database Requirements

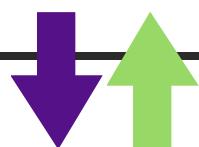
Contains logical requirements for information inserted into the database.

Functional Requirement ID		REQ-17: Active Revisions
Description	Only one revision of a survey can be active.	
Origin	Revisions;	
Priority	High	
FIT Criterion	Ensure that through the Back-Office Application, that there is no way to set two revisions as active.	
Type	FUNCTIONAL	
Category	DATABASE;	

3.5 Design Constraints

Specific requirements imposing standards or limitations on the system.

There are no requirements at this time.



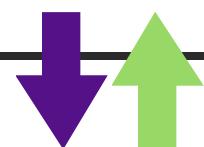
3.6 Software System Attributes

3.6.1 Reliability

There are no requirements at this time.

3.6.2 Availability

Functional Requirement ID	FR-14: Web Service Down-Time
Description	The Web Services Down-Time shall not exceed 5%
Origin	WEB-SERVICES;
Priority	Low
FIT Criterion	The down-time of the Web Services should fit within the specified percentage range
Type	NON-FUNCTIONAL;
Category	SYSTEM;



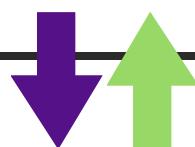
3.6.3 Security

Functional Requirement ID	FR-15: Web Service Request Security
Description	The Web Service Requests shall contain security to ensure that the security code of that is sent along with the request matches a trusted application.
Origin	WEB-SERVICES;
Priority	Low
FIT Criterion	None;
Type	NON-FUNCTIONAL;
Category	SYSTEM;

Functional Requirement ID	FR-16: Encrypted Requests and Responses
Description	All Sensitive Information Shall be Encrypted when sent to and from the web services
Origin	WEB-SERVICES;
Priority	Low
FIT Criterion	Check that the request and response strings are encrypted.
Type	NON-FUNCTIONAL;
Category	SYSTEM;

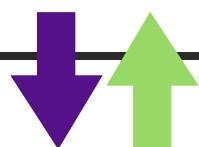
3.6.4 Maintainability

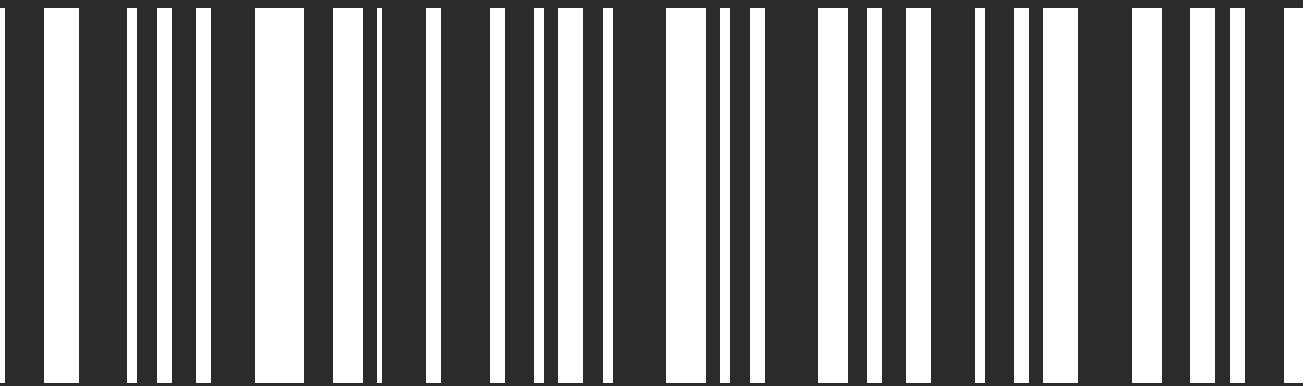
Functional Requirement ID	FR-11: Web Admin Data
Description	The use of the back office is to maintain the data in the system and database/
Origin	Back-Office;
Priority	Low
FIT Criterion	None
Type	NON-FUNCTIONAL;
Category	CORE;





Functional Requirement ID		FR-12: Test Automation CRON Job
Description	A CRON Job will be ran daily at 12:00am and will perform a test run on all automations to ensure that no updates have been performed on the Company Survey Sites.	
Origin	CRON;	
Priority	MEDIUM-HIGH;	
FIT Criterion	A CRON Job is made that accomplishes this requirement and is successfully run at 12:00am daily.	
Type	NON-FUNCTIONAL;	
Category	SYSTEM;	
Functional Requirement ID		FR-13: Test Automation CRON Job Failure
Description	If the Automation CRON Job finds a survey in which the Automation Process fails, the error will be logged and need of Administrator Assistance to look into the Failed Automation will be requested.	
Origin	CRON;	
Priority	MEDIUM-HIGH;	
FIT Criterion	A failed automation note should be added to the CRON log. And message to the Administrator shall be sent.	
Type	NON-FUNCTIONAL;	
Category	SYSTEM;	





USE CASE DOCUMENT

8/23/2013

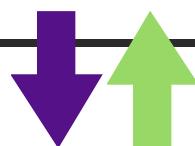
Graduate Capstone

RECEIPT

REWARDS

Table of Contents

1 Introduction.....	3
1.1 Intended Document	3
1.2 Resources	3
1.3 Revision History	3
2 Use Cases.....	4
2.1 Phone Application.....	4
UC-01: See Main Menu	4
UC-02: User Sign In	5
UC-03: User Sign Out	6
UC-04: Browse All Companies	7
UC-05: View Company Information	8
UC-06: View Survey Information.....	9
UC-07: Take Survey	10
UC-08: Answer Survey Question	11
UC-10: User Search For Companies.....	12
2.2 Web Services.....	13
UC-09: Get List of Companies	13
UC-22: Add / Edit / Delete Code Type	14
UC-23: Add / Edit / Delete Code.....	15
UC-24: Edit / Delete User.....	16
UC-25: Add / Edit / Delete Company	17
UC-26: Add / Edit / Delete Survey.....	18
UC-27: Add / Edit / Delete Revision	19
UC-28: Add / Edit / Delete Command Element.....	20
UC-29: Add / Edit / Delete Command Identifier	21
UC-30: Add / Edit / Delete Form Element	22
UC-31: Add / Edit / Delete Form Element Attribute	23
UC-32: Add / Edit / Delete Form Element Option	24
2.3 Administrator Web-Application.....	25
UC-11: Add / Edit / Delete Code Types	25
UC-12: Add / Edit / Delete Code.....	26
UC-13: Add / Edit / Delete Company	27
UC-14: Add / Edit / Delete Survey.....	28
UC-15: Add / Edit / Delete Revision	29
UC-16: Add / Edit / Delete Command Element.....	30
UC-17: Add / Edit / Delete Command Identifier	31
UC-18: Add / Edit / Delete Form Element	32
UC-19: Add / Edit / Delete Form Element Attribute	33
UC-20: Add / Edit / Delete Form Element Option	34
UC-21: Add / Edit / Delete User.....	35



1 Introduction

This document is a resource containing all the use cases that are involved with the project.

1.1 Intended Document

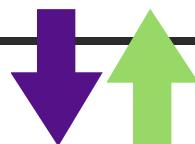
This document is intended for individuals with a medium level of technical background.

1.2 Resources

None

1.3 Revision History

Name	Date	Reason For Change	Version
Andy Bottom	04/12/2013	Started making the document; Created First Couple Use Cases	0.1
Andy Bottom	04/16/2013	Continued making document. Implemented the next several use cases and requirements	0.1.1
Andy Bottom	04/27/2013	Implemented almost all of the use cases and diagrams	0.2
Andy Bottom	05/01/2013	Finished first version	1.0
Andy Bottom	08/18/2013	Proofread. It all looks good	1.1

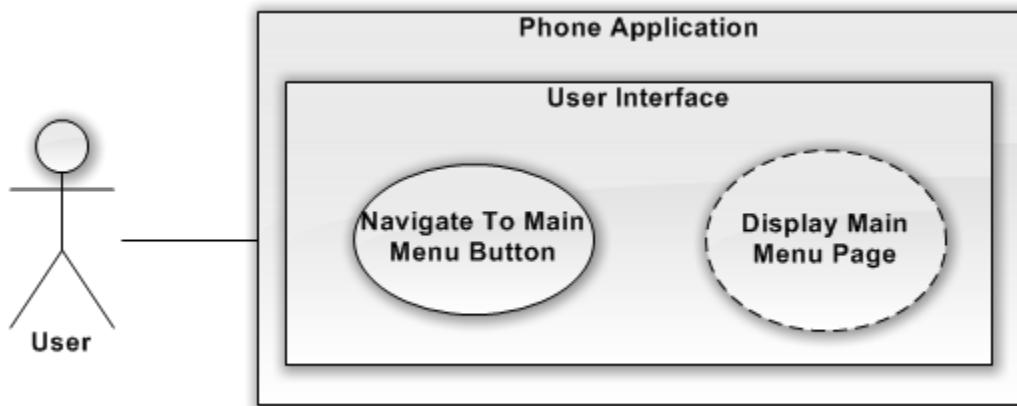


2 Use Cases

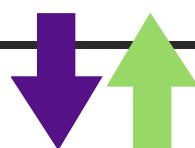
Below is a list of use cases for the phone application, web services and the web admin.

2.1 Phone Application

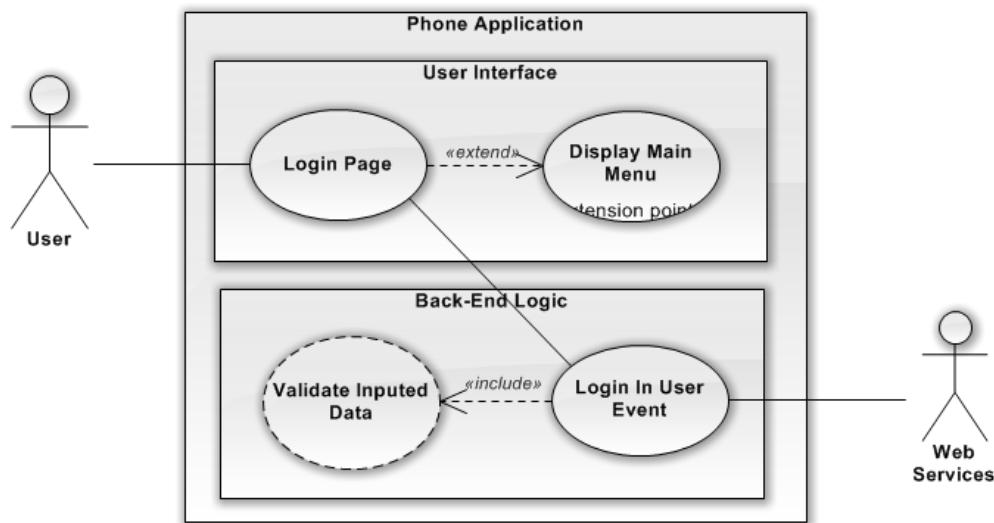
UC-01: See Main Menu



Use Case Name	UC-01: See Main Menu
Purpose	How the user can get to the main menu
Summary	The User will get to the main menu
Work Products	Phone Application
Users/Actors	User
Entry Criteria	User is viewing a screen in the application.
Preconditions	A User is on any screen in the application.
Basic Course of Events	01: User hit the home / menu button on the phone. 02: The main menu screen will be displayed
Alternative Paths	AP-01: The application is not started and the user starts the application. AP-02: The application should show the main menu on start-up
Postconditions	The about us page content has been updated
Exit Criteria	Admin would see the content and be happy with it.



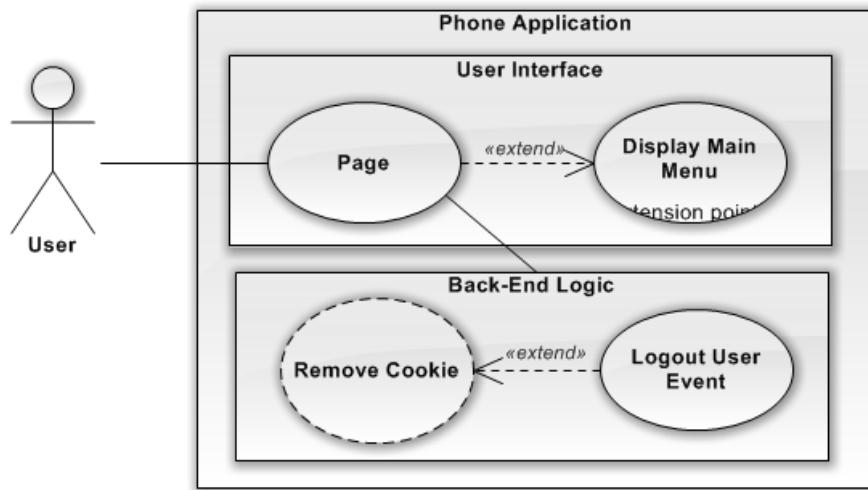
UC-02: User Sign In



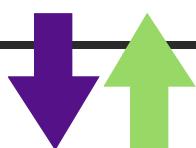
Use Case Name	UC-02: User Sign In
Purpose	Gives the user the ability to log in
Summary	Sign In Feature
Rationale	This way a user can take advantage of better features of the system.
Work Products	Phone Application
Users/Actors	User; Web Services
Entry Criteria	The User has navigated to the Sign In Screen
Preconditions	The User is not signed in already in the device
Basic Course of Events	<p>01: User enters their username</p> <p>02: User enters their password / pin</p> <p>03: User presses the submit button</p> <p>03-1: Upon Submit, the application will verify the inputs from the user.</p> <p>03-2: Upon a successful verification it sends sign in request to the server.</p> <p>04: User waits for response to return.</p> <p>04-1: The services will verify that the user and password match.</p> <p>04-2: Upon a successful match, the services will return the user information to login.</p> <p>05: Successful Credentials received</p> <p>06: The phone creates a cookie to store the logged in user.</p> <p>07: User sees successful login message and is directed to the main menu.</p>
Alternative Paths	<p>AP-03-1: If a user does not submit a valid username or password, upon submit, it will display an error to the user. (Such as blank would cause error)</p> <p>AP-04-2: Upon a unsuccessful match the system returns error back and the error is then sent to user to try again.</p>
Postconditions	The user is on the main menu page and is logged in to their user account.
Exit Criteria	User is logged in.



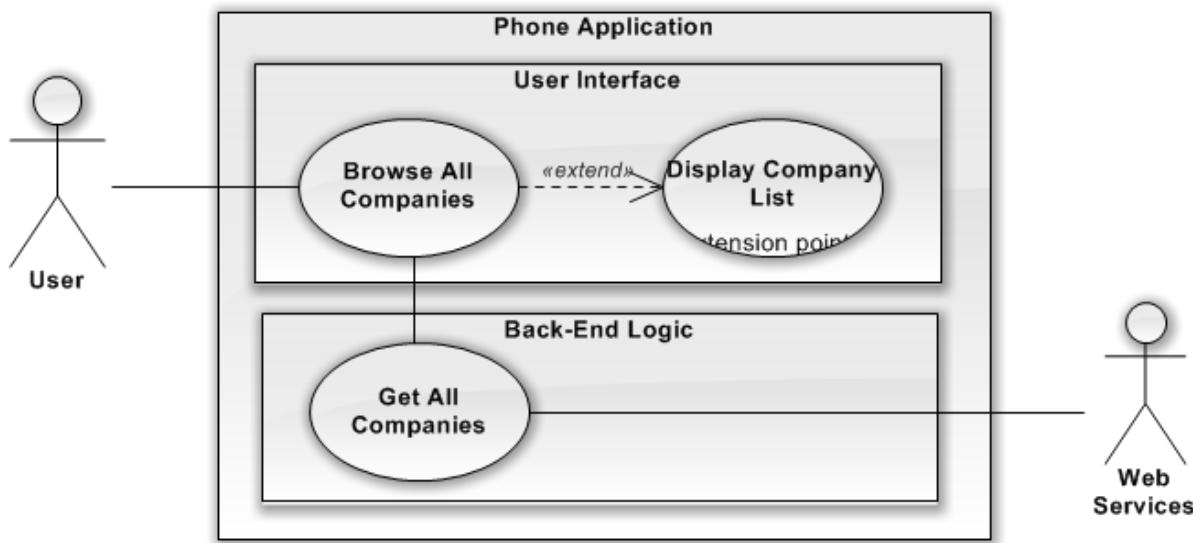
UC-03: User Sign Out



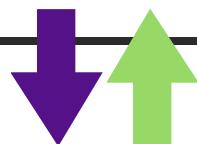
Use Case Name	UC-03: User Sign Out
Description	The user can sign out of their account in the application and become just an anonymous user
Work Products	Phone Application
Users/Actors	User
Assumptions	The User has navigated to the Sign Out Screen
Preconditions	The User is already signed in
Steps	<p>01: User Clicks the Logout Button</p> <p>01-1: Message Prompts are you sure.</p> <p>01-2: User clicks accept</p> <p>02: Cookie is deleted from phone.</p> <p>03: Message Returned saying that they were successfully logged out.</p> <p>04: User returned to main menu</p>
Variations	<p>AP-02: There may be a need to send information to the web service to set some sort of properties. In this case, they will need to wait for this to successes.</p> <p>AP-01-2: If user clicks cancel, then the logout process is stopped and the user is returned to the screen they were on and are still logged in.</p>
Postconditions	The user is no longer logged in on the application.
Exit Criteria	User is logged out.
Notes	There may or may not be needed information to store about logout information. So Web Services at this time will not be updated.



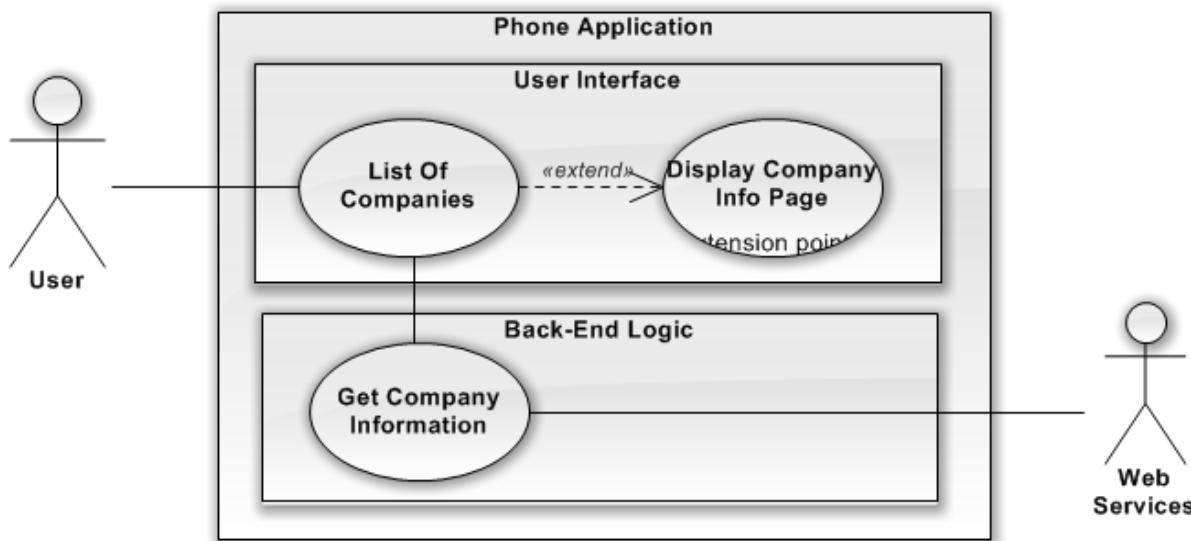
UC-04: Browse All Companies



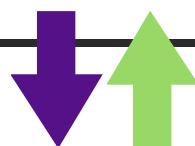
Use Case Name		UC-04: Browse All Companies
Description	A Screen that lists all the companies that are in the system.	
Work Products	Phone Application	
Users/Actors	User; Web Services	
Assumptions	The User navigates to Browse All Companies Page	
Preconditions	None	
Steps	01: The User Navigates to the Browse All Companies Page 01-1: Sends request to webserver to obtain a list of all the companies 02: User sees loading icon, to wait. 02-1: The app receives the requests, parses it and outputs to user 03: User sees a list of companies	
Variations	AP-02-1: If the app receives error, then it should handle error to try again or that it is not available. AP-01-1: If there is a local cache of the companies, then it should parse that.	
Postconditions	User sees a list of companies	
Exit Criteria	Companies are displayed	
Notes	The ability to sort companies by alpha, category type, etc... would be nice.	
References	UC-09;	



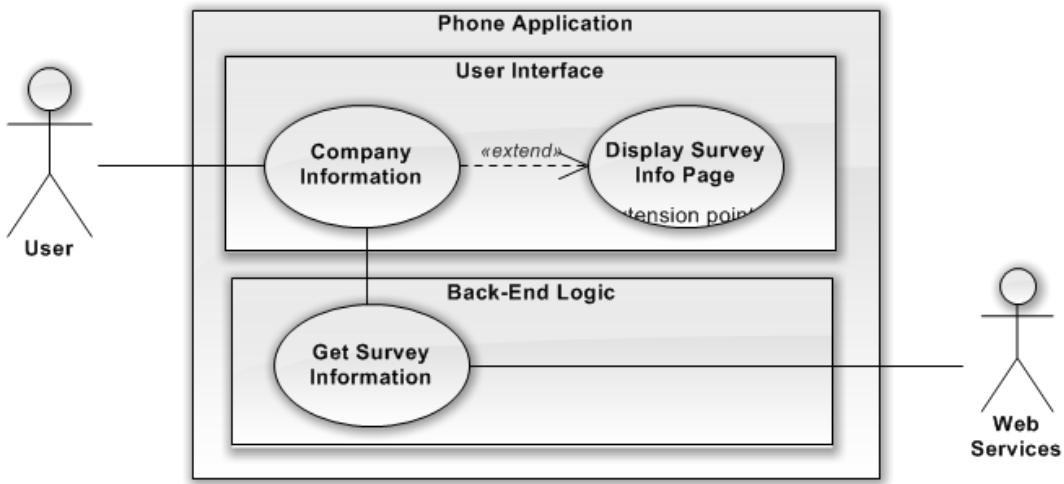
UC-05: View Company Information



Use Case Name		UC-05: View Company Information
Description	Displays a screen containing background information about a specific company	
Work Products	Phone Application	
Users/Actors	User	
Assumptions	The User chooses a company to see the information about	
Preconditions	The user has chosen a company	
Steps	<p>01: The User Chooses a company to see information about.</p> <p>01-1: A Request is sent to web server to get information about the company</p> <p>02: User sees waiting icon</p> <p>02-1: App receives information and parses it and outputs to user.</p> <p>03: The Company information page is displayed to the user</p>	
Variations	<p>AP-02-1: If the app receives error, then it should handle error to try again or that it is not available.</p> <p>AP-01-1: If there is a local cache of the company, then it should parse that.</p>	
Postconditions	User sees the Company Information Page	
Exit Criteria	Company is displayed	
Note	If the company is already stored in cache, then there would be no need to query the web services again.	



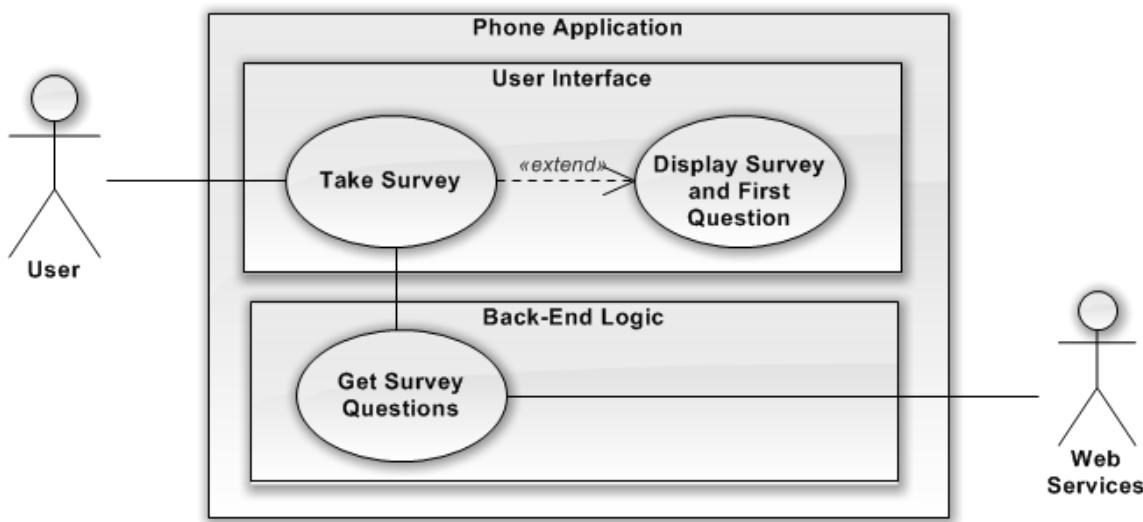
UC-06: View Survey Information



Use Case Name		UC-06: View Survey Information
Description	Displays a screen containing information about a specific Survey of a Company	
Work Products	Phone Application	
Users/Actors	User	
Assumptions	The User chooses a Survey to see further information	
Preconditions	The user has chosen a survey	
Steps	<ol style="list-style-type: none">01: The User Chooses a survey to see information about.01-1: A Request is sent to web server to get information about the company02: User sees waiting icon02-1: App receives information and parses it and outputs to user.03: The Survey information page is displayed to the user	
Variations	<p>AP-02-1: If the app receives error, then it should handle error to try again or that it is not available.</p> <p>AP-01-1: If there is a local cache of the survey, then it should parse that.</p>	
Postconditions	User sees the Survey Information Page	
Exit Criteria	Survey Information is displayed	
Notes	If the survey information is already stored in cache, then there would be no need to query the web services again.	



UC-07: Take Survey

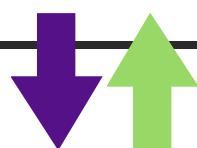


Use Case Name		UC-07: Take Survey
Description	Start the Survey	
Work Products	Phone Application	
Users/Actors	User	
Assumptions	The User is on the Survey Page	
Preconditions	The User is on a specific survey for a specific company	
Steps	01: The User Chooses to Start the Survey 01-1: A Request is sent to web server to get the survey questions. It sends along user's logged in ID if it is logged in. 02: User sees waiting icon 02-1: App receives information and parses it and outputs to user. 03: The Survey Questions will be displayed to the user to take the survey	
Variations	AP-01-1-A: If a user is logged in and then it will receive questions with some answers already filled out since we know the information already. AP-01-1-B: If a user is anonymous then it will receive all questions of the survey	
Postconditions	User sees the Survey of a Company	
Exit Criteria	Survey is displayed	

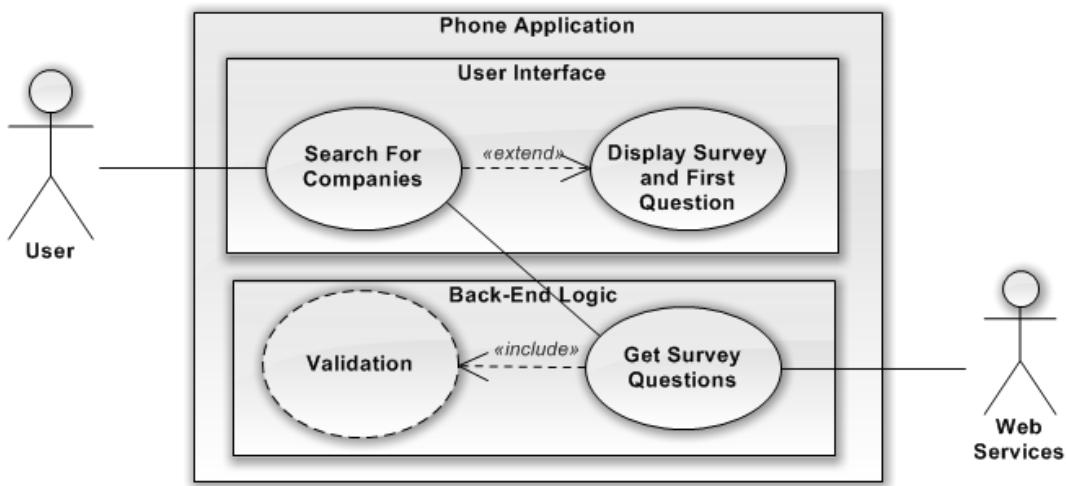


UC-08: Answer Survey Question

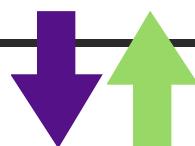
Use Case Name	UC-08: Answer Survey Question
Description	Answering a Question
Work Products	Phone Application
Users/Actors	User
Assumptions	The user is currently taking a survey
Preconditions	A user is taking a unique survey and is on a unique question
Steps	<p>01: A question is displayed to the user along with a input element (radio, select box, textbox, etc...)</p> <p>01-1: A Request is sent to web server to get the survey questions. It sends along a user's logged in ID if it is logged in.</p> <p>02: User sees waiting icon</p> <p>02-1: App receives information and parses it and outputs to user.</p> <p>03: The Survey Questions will be displayed to the user to take the survey</p>
Variations	<p>AP-01-1-A: If a user is logged in and then it will receive questions with some answers already filled out since we know the information already.</p> <p>AP-01-1-B: If a user is anonymous then it will receive all questions of the survey</p>
Postconditions	User sees the Survey of a Company
Exit Criteria	Survey is displayed



UC-10: User Search for Companies

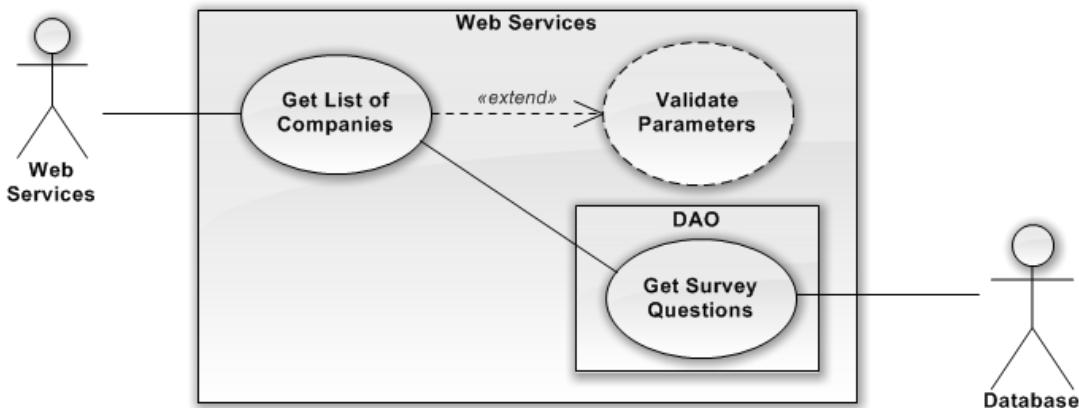


Use Case Name	UC-10: User Search For Companies
Description	User may also use a search to lookup a company. They will use the form to select certain parameters to search on including a search text; company type;
Work Products	Phone Application
Users/Actors	User; Web Services
Assumptions	The User has navigated to the search for company page.
Preconditions	None;
Steps	<p>01: The User fills out the Search for Company Page;</p> <p>01-01: This may include searching for a string</p> <p>01-02: Searching by Company Category type.</p> <p>02: User Clicks the Search button</p> <p>03: User see's message, "Searching, Please Wait."</p> <p>03-01: Search is sent to web service along with the parameters;</p> <p>03-02: Response is received</p> <p>03-03: The Response is parsed in put into Company Objects</p> <p>03-04: The company objects will be outputted as a list for the user.</p> <p>04: The List of Company Results is displayed.</p>
Variations	<p>AP-03-01: If the app receives error, then it should handle error to try again or that it is not available.</p> <p>AP-03-02: If Response contains no company objects, message will say, "There were no companies found. Please Change Search Criteria."</p>
Postconditions	A list of companies has been obtained.
Exit Criteria	User sees the Company Search Results Page.
References	UC-09;



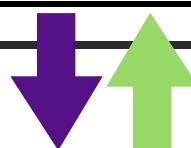
2.2 Web Services

UC-09: Get List of Companies

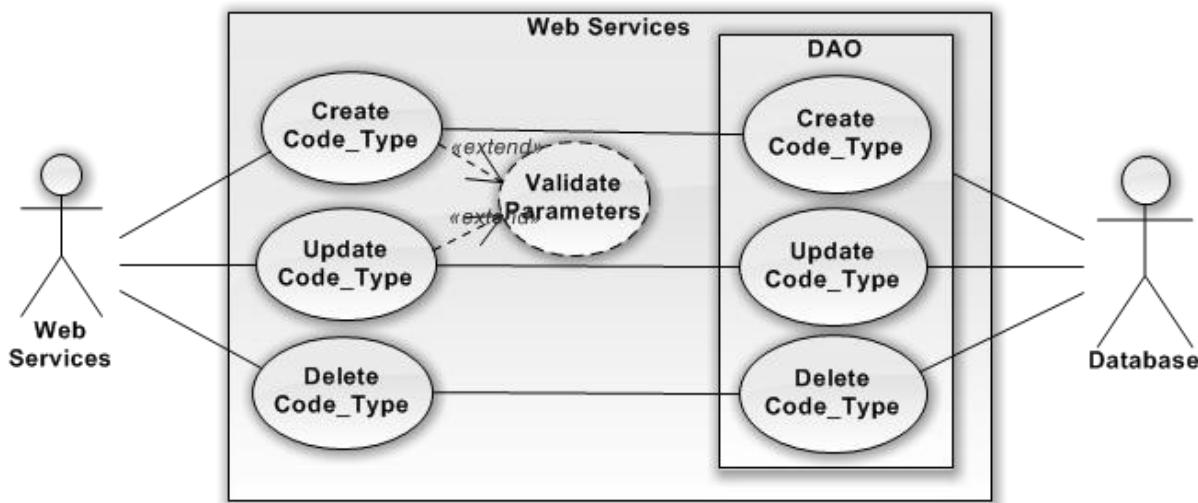


Use Case Name UC-09: Get List Of Companies

Description	The Web Services method that gets call to get a list of Companies pending on the certain criteria
Work Products	Web Services;
Users/Actors	Web Services; Database;
Assumptions	Received Parameters is Validated;
Preconditions	The Method gets called; Receives Certain Search Criteria
Steps	<p>01: Web Services Get List of Companies gets Called</p> <p>02: Validates received parameters</p> <p>03: Calls DAO to get list of companies from database and passes the searched criteria</p> <p>03-01: DAO calls the company table and queries on the received parameters. Returns a list of Company Objects</p> <p>04: List of Company Objects are returned.</p>
Variations	AP-03-01: If can't connect to database, it will timeout and return error.
Postconditions	A list of company objects is returned
Exit Criteria	Web Services wait for next call
Notes	The types of parameters may include, user favorites, company categories, search based on name or description.

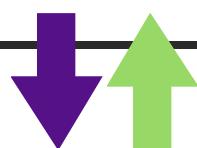


UC-22: Add / Edit / Delete Code Type

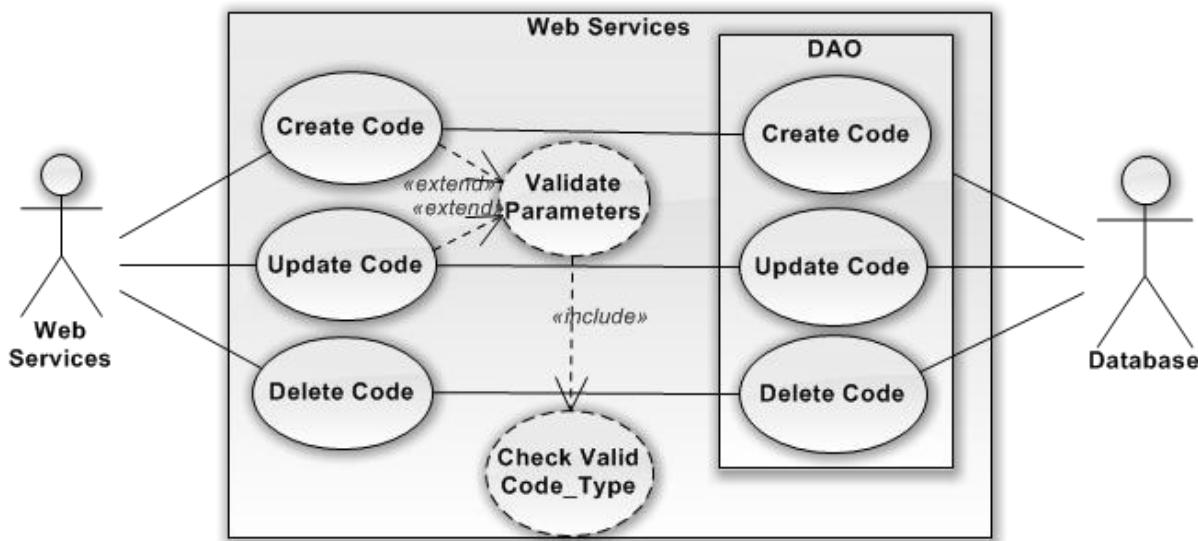


Use Case Name UC-22: Add / Edit / Delete Code Type

Description	The Web Services method that gets called to create, edit or delete a Code Type Object.
Work Products	Web Services;
Users/Actors	Web Services; Database;
Assumptions	Received Parameters is Validated;
Preconditions	The Method gets called; Receives Certain Parameters Criteria
Steps	<p>01: The Web Service Method Add / Edit / Delete Code Type gets called</p> <p>02: Validates received parameters</p> <p>03: Calls DAO to perform the needed logic in the database</p> <p>03-01: DAO calls the Code_Type table and queries on the received parameters. Returns success message</p> <p>04: Return Success Message</p>
Variations	AP-03-01: If can't connect to database, it will timeout and return error.
Postconditions	The correct database action for the Code_Type table has been performed
Exit Criteria	Web Services wait for next call

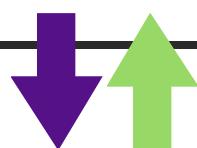


UC-23: Add / Edit / Delete Code

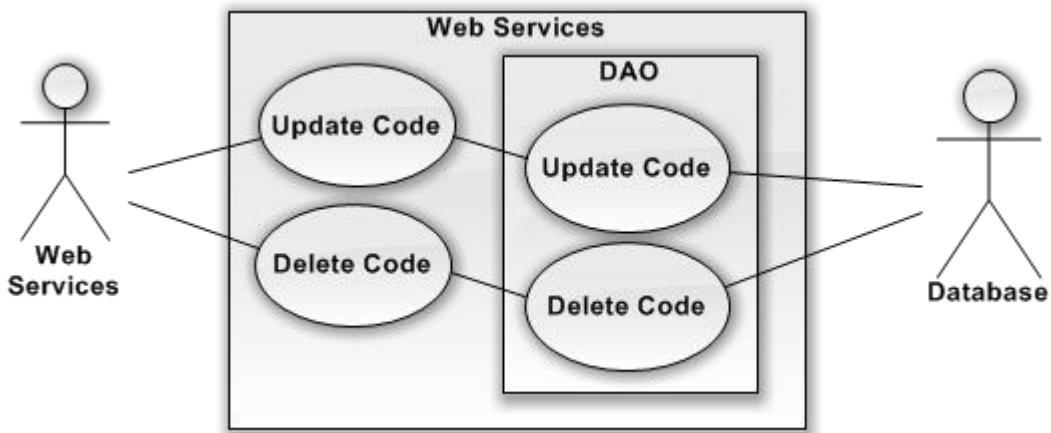


Use Case Name UC-23: Add / Edit / Delete Code

Description	The Web Services method that gets called to create, edit or delete a Code Object.
Work Products	Web Services;
Users/Actors	Web Services; Database;
Assumptions	Received Parameters is Validated;
Preconditions	The Method gets called; Receives Certain Parameters Criteria
Steps	<p>01: The Web Service Method Add / Edit / Delete Code gets called</p> <p>02: Validates received parameters</p> <p>03: Calls DAO to perform the needed logic in the database</p> <p>03-01: DAO calls the Code table and queries on the received parameters. Returns success message</p> <p>04: Return Success Message</p>
Variations	AP-03-01: If can't connect to database, it will timeout and return error.
Postconditions	The correct database action for the Code table has been performed
Exit Criteria	Web Services wait for next call

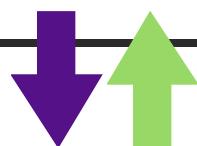


UC-24: Edit / Delete User

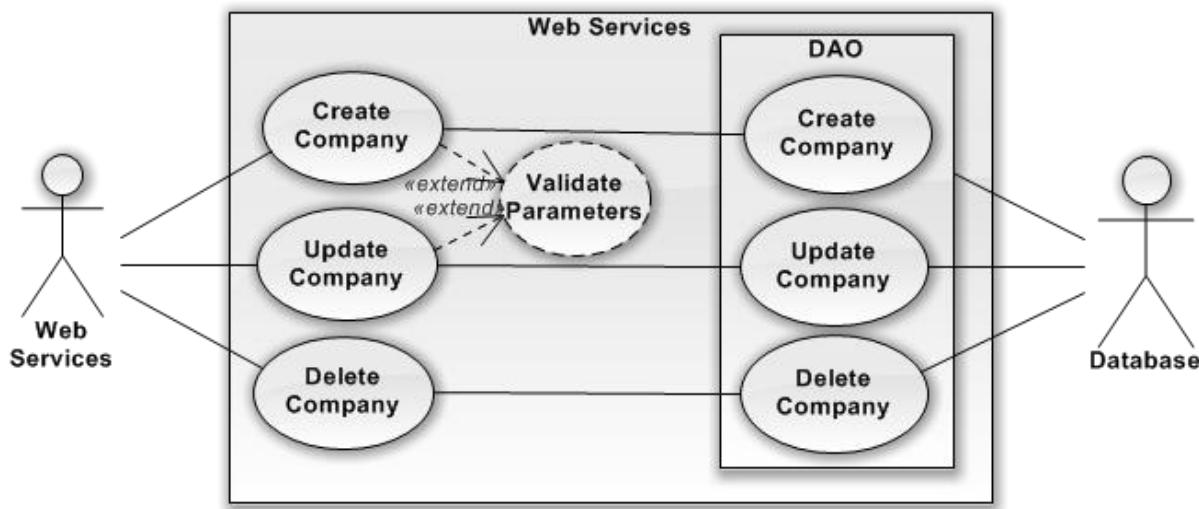


Use Case Name UC-24: Edit / Delete Code

Description	The Web Services method that gets called to edit or delete a User Object.
Work Products	Web Services;
Users/Actors	Web Services; Database;
Assumptions	Received Parameters is Validated;
Preconditions	The Method gets called; Receives Certain Parameters Criteria
Steps	<p>01: The Web Service Method Edit / Delete User gets called</p> <p>02: Validates received parameters</p> <p>03: Calls DAO to perform the needed logic in the database</p> <p>03-01: DAO calls the User table and queries on the received parameters. Returns success message</p> <p>04: Return Success Message</p>
Variations	AP-03-01: If can't connect to database, it will timeout and return error.
Postconditions	The correct database action for the User table has been performed
Exit Criteria	Web Services wait for next call



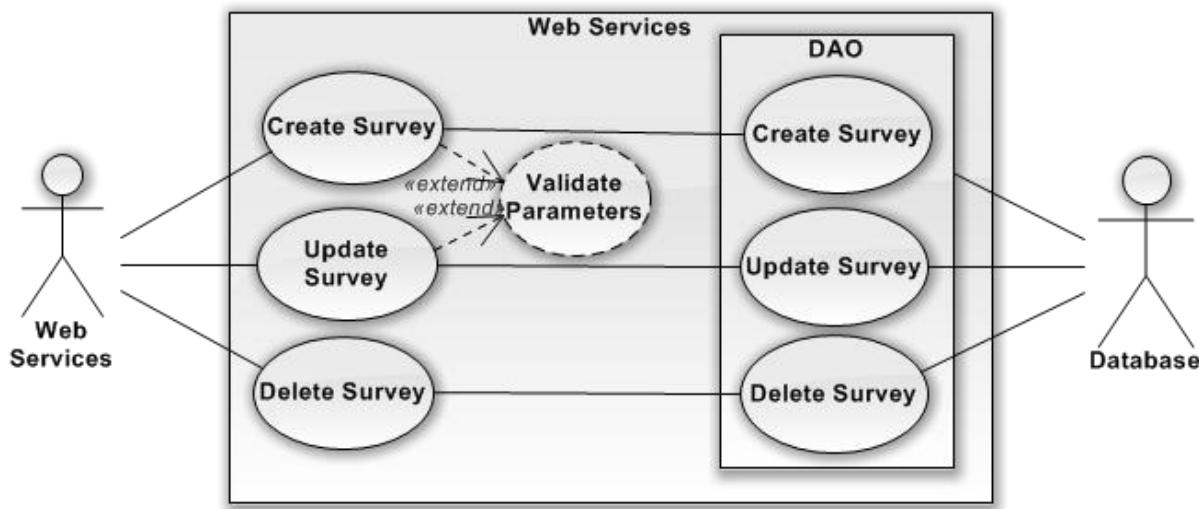
UC-25: Add / Edit / Delete Company



Use Case Name		UC-25: Add / Edit / Delete Company
Description	The Web Services method that gets called to create, edit or delete a Company Object.	
Work Products	Web Services;	
Users/Actors	Web Services; Database;	
Assumptions	Received Parameters is Validated;	
Preconditions	The Method gets called; Receives Certain Parameters Criteria	
Steps	01: The Web Service Method Add / Edit / Delete Company gets called 02: Validates received parameters 03: Calls DAO to perform the needed logic in the database 03-01: DAO calls the Company table and queries on the received parameters. Returns success message 04: Return Success Message	
Variations	AP-03-01: If can't connect to database, it will timeout and return error.	
Postconditions	The correct database action for the Company table has been performed	
Exit Criteria	Web Services wait for next call	



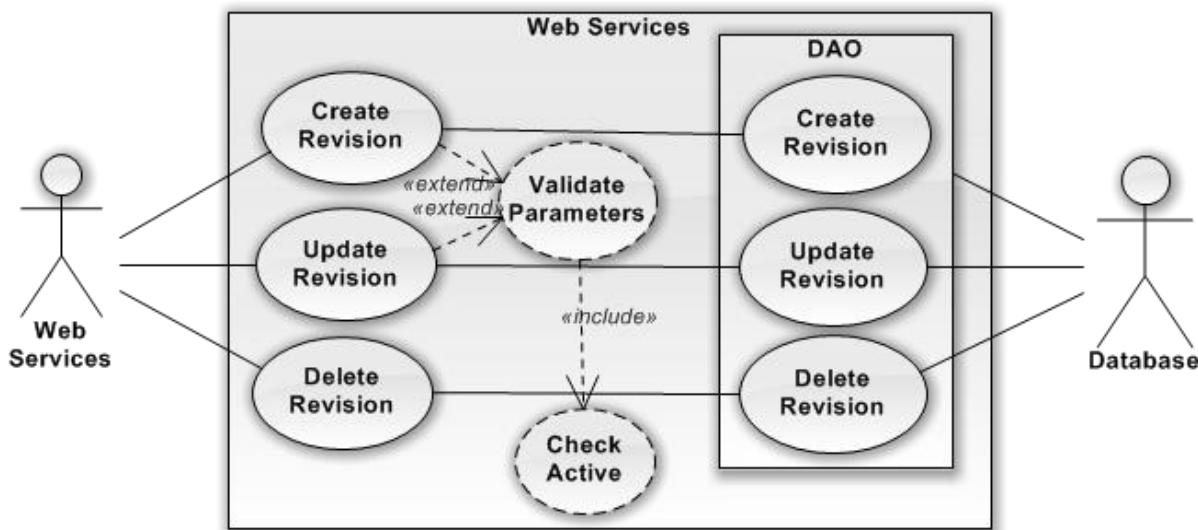
UC-26: Add / Edit / Delete Survey



Use Case Name	UC-26: Add / Edit / Delete Survey
Description	The Web Services method that gets called to create, edit or delete a Survey Object.
Work Products	Web Services;
Users/Actors	Web Services; Database;
Assumptions	Received Parameters is Validated;
Preconditions	The Method gets called; Receives Certain Parameters Criteria
Steps	<p>01: The Web Service Method Add / Edit / Delete Survey gets called</p> <p>02: Validates received parameters</p> <p>03: Calls DAO to perform the needed logic in the database</p> <p>03-01: DAO calls the Survey table and queries on the received parameters. Returns success message</p> <p>04: Return Success Message</p>
Variations	AP-03-01: If can't connect to database, it will timeout and return error.
Postconditions	The correct database action for the Survey table has been performed
Exit Criteria	Web Services wait for next call

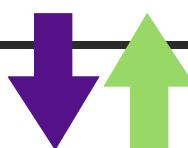


UC-27: Add / Edit / Delete Revision

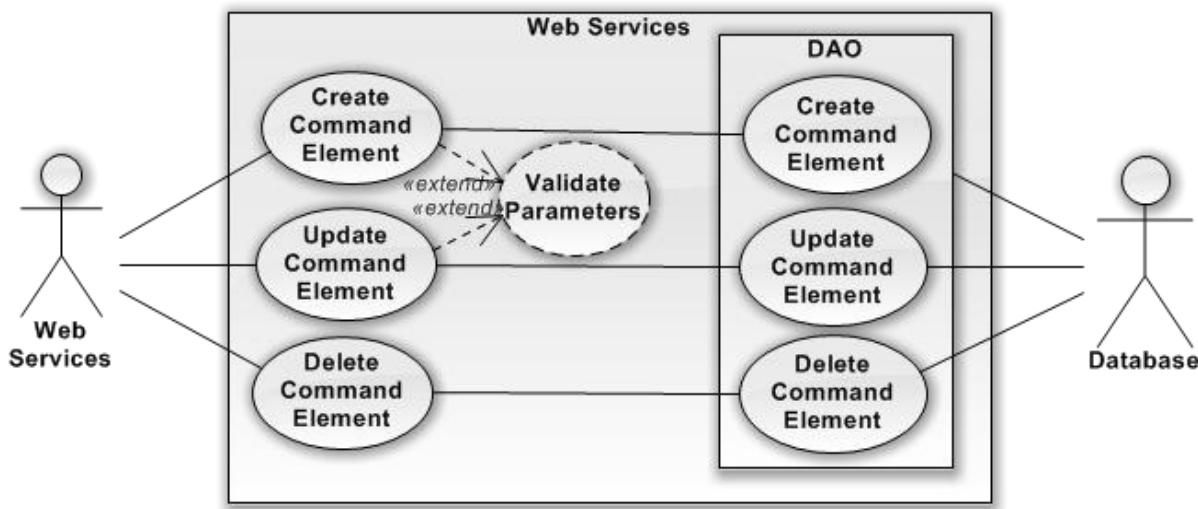


Use Case Name UC-27: Add / Edit / Delete Revision

Description	The Web Services method that gets called to create, edit or delete a Revision Object.
Work Products	Web Services;
Users/Actors	Web Services; Database;
Assumptions	Received Parameters is Validated;
Preconditions	The Method gets called; Receives Certain Parameters Criteria
Steps	<p>01: The Web Service Method Add / Edit / Delete Revision gets called</p> <p>02: Validates received parameters</p> <p>03: Calls DAO to perform the needed logic in the database</p> <p>03-01: DAO calls the Revision table and queries on the received parameters. Returns success message</p> <p>04: Return Success Message</p>
Variations	AP-03-01: If can't connect to database, it will timeout and return error.
Postconditions	The correct database action for the Revision table has been performed
Exit Criteria	Web Services wait for next call



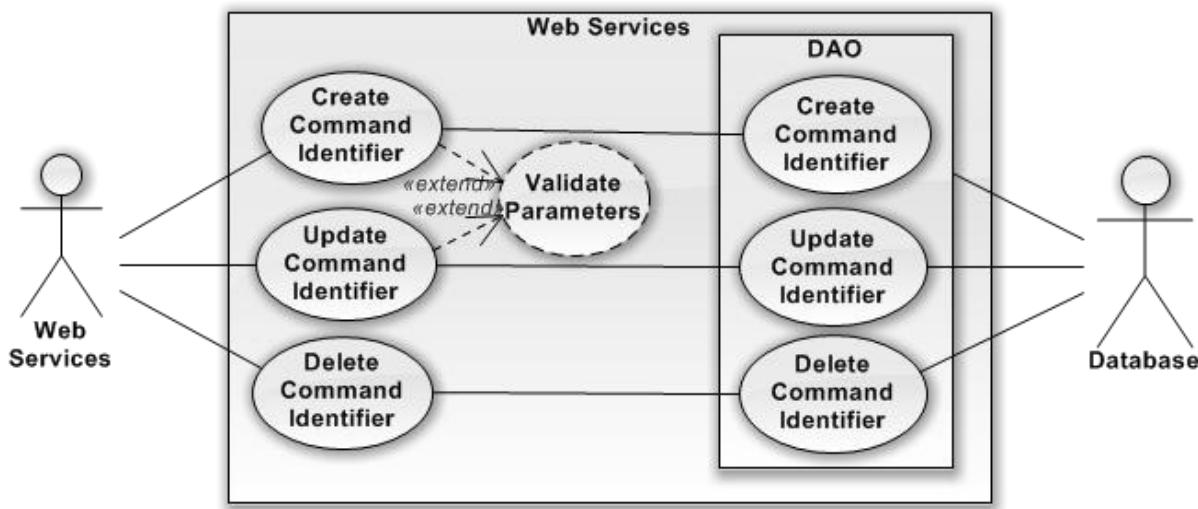
UC-28: Add / Edit / Delete Command Element



Use Case Name		UC-28: Add / Edit / Delete Command Element
Description	The Web Services method that gets called to create, edit or delete a Command Element Object.	
Work Products	Web Services;	
Users/Actors	Web Services; Database;	
Assumptions	Received Parameters is Validated;	
Preconditions	The Method gets called; Receives Certain Parameters Criteria	
Steps	01: The Web Service Method Add / Edit / Delete Command Element gets called 02: Validates received parameters 03: Calls DAO to perform the needed logic in the database 03-01: DAO calls the Command Element table and queries on the received parameters. Returns success message 04: Return Success Message	
Variations	AP-03-01: If can't connect to database, it will timeout and return error.	
Postconditions	The correct database action for the Command Element table has been performed	
Exit Criteria	Web Services wait for next call	



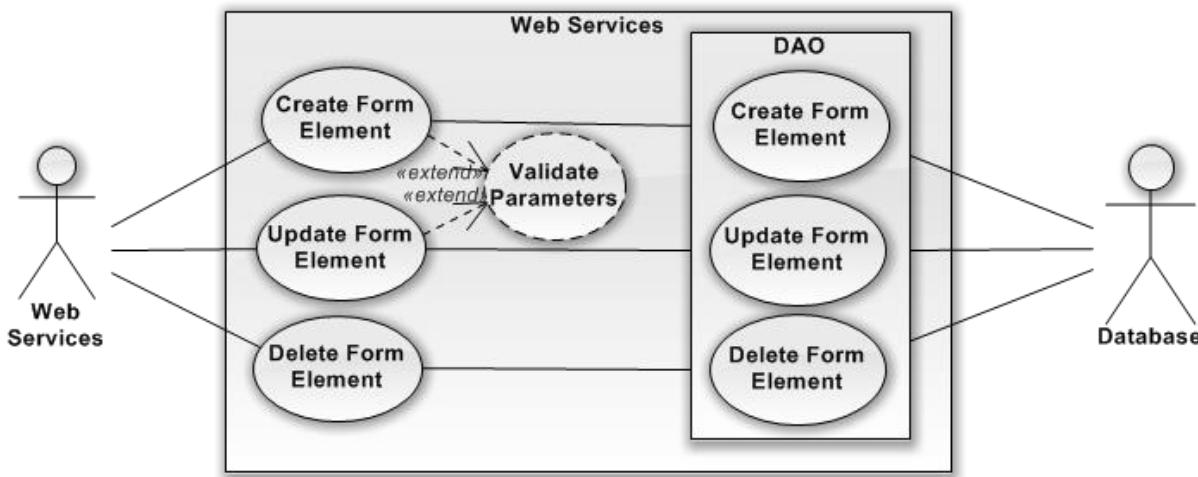
UC-29: Add / Edit / Delete Command Identifier



Use Case Name		UC-29: Add / Edit / Delete Command Identifier
Description	The Web Services method that gets called to create, edit or delete a Command Identifier Object.	
Work Products	Web Services;	
Users/Actors	Web Services; Database;	
Assumptions	Received Parameters is Validated;	
Preconditions	The Method gets called; Receives Certain Parameters Criteria	
Steps	01: The Web Service Method Add / Edit / Delete Command Identifier gets called 02: Validates received parameters 03: Calls DAO to perform the needed logic in the database 03-01: DAO calls the Command Identifier table and queries on the received parameters. Returns success message 04: Return Success Message	
Variations	AP-03-01: If can't connect to database, it will timeout and return error.	
Postconditions	The correct database action for the Command Identifier table has been performed	
Exit Criteria	Web Services wait for next call	

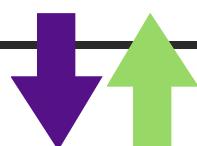


UC-30: Add / Edit / Delete Form Element

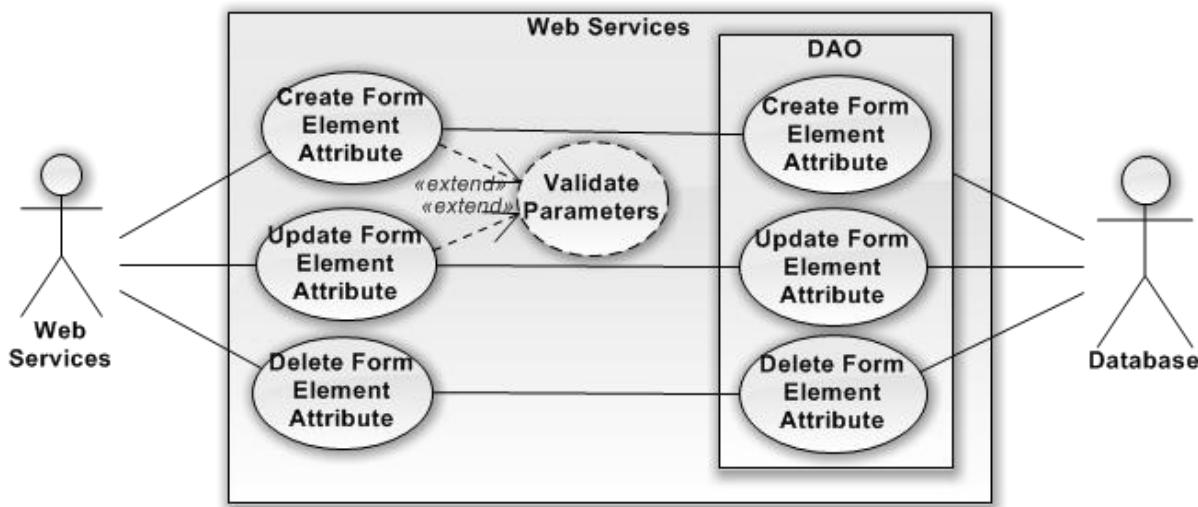


Use Case Name UC-30: Add / Edit / Delete Form Element

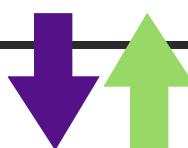
Description	The Web Services method that gets called to create, edit or delete Form Element Object.
Work Products	Web Services;
Users/Actors	Web Services; Database;
Assumptions	Received Parameters is Validated;
Preconditions	The Method gets called; Receives Certain Parameters Criteria
Steps	<p>01: The Web Service Method Add / Edit / Delete Form Element gets called</p> <p>02: Validates received parameters</p> <p>03: Calls DAO to perform the needed logic in the database</p> <p>03-01: DAO calls the Form Element table and queries on the received parameters. Returns success message</p> <p>04: Return Success Message</p>
Variations	AP-03-01: If can't connect to database, it will timeout and return error.
Postconditions	The correct database action for the Form Element table has been performed
Exit Criteria	Web Services wait for next call



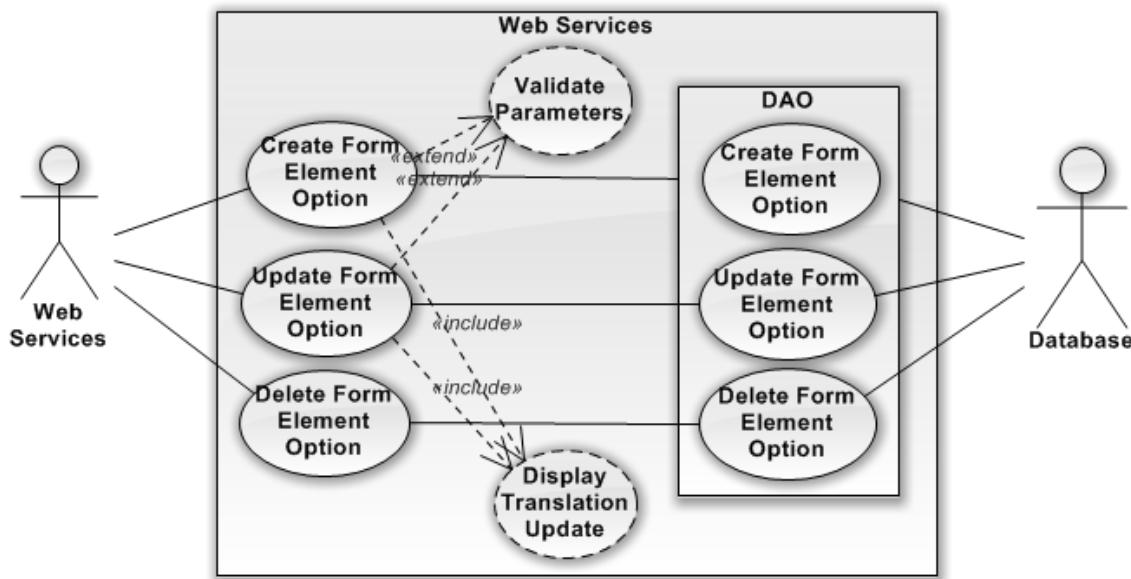
UC-31: Add / Edit / Delete Form Element Attribute



Use Case Name	UC-31: Add / Edit / Delete Form Element Attribute
Description	The Web Services method that gets called to create, edit or delete Form Element Attribute Object.
Work Products	Web Services;
Users/Actors	Web Services; Database;
Assumptions	Received Parameters is Validated;
Preconditions	The Method gets called; Receives Certain Parameters Criteria
Steps	<p>01: The Web Service Method Add / Edit / Delete Form Element Attribute gets called</p> <p>02: Validates received parameters</p> <p>03: Calls DAO to perform the needed logic in the database</p> <p>03-01: DAO calls the Form Element Attribute table and queries on the received parameters. Returns success message</p> <p>04: Return Success Message</p>
Variations	AP-03-01: If can't connect to database, it will timeout and return error.
Postconditions	The correct database action for the Form Element Attribute table has been performed
Exit Criteria	Web Services wait for next call



UC-32: Add / Edit / Delete Form Element Option



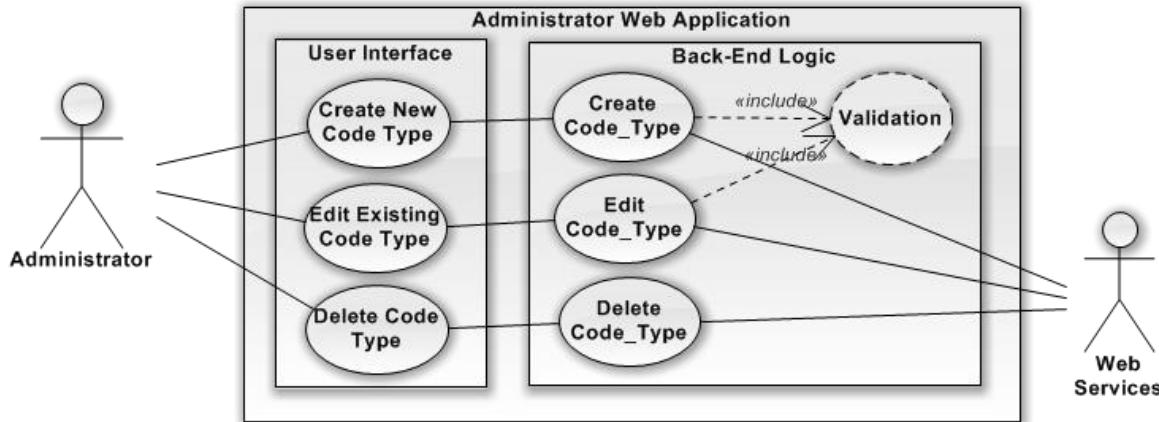
Use Case Name UC-32: Add / Edit / Delete Form Element Option

Description	The Web Services method that gets called to create, edit or delete Form Element Option Object.
Work Products	Web Services;
Users/Actors	Web Services; Database;
Assumptions	Received Parameters is Validated;
Preconditions	The Method gets called; Receives Certain Parameters Criteria
Steps	<p>01: The Web Service Method Add / Edit / Delete Form Element Option gets called</p> <p>02: Validates received parameters</p> <p>03: Calls DAO to perform the needed logic in the database</p> <p>03-01: DAO calls the Form Element Option table and queries on the received parameters. Returns success message</p> <p>04: Return Success Message</p>
Variations	AP-03-01: If can't connect to database, it will timeout and return error.
Postconditions	The correct database action for the Form Element Option table has been performed
Exit Criteria	Web Services wait for next call



2.3 Administrator Web-Application

UC-11: Add / Edit / Delete Code Types

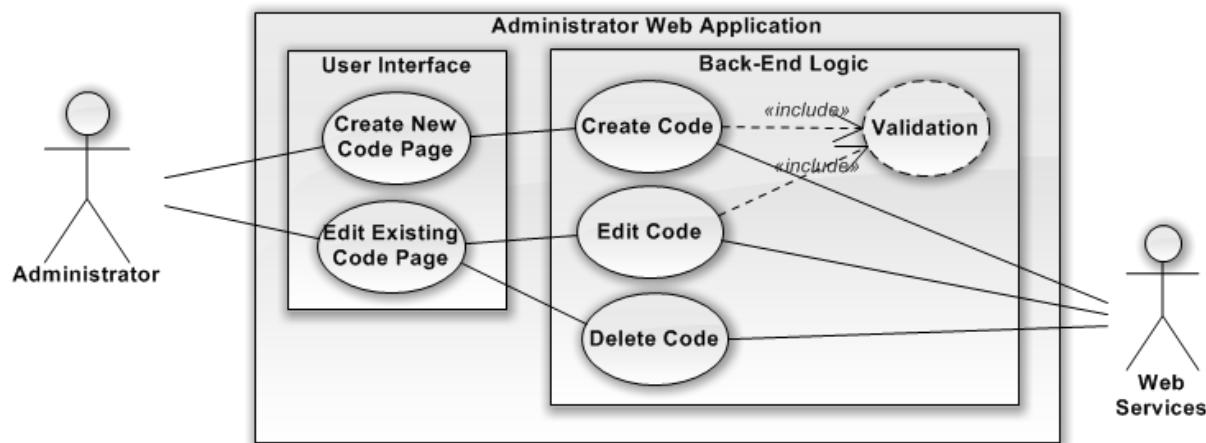


Use Case Name UC-11: Add / Edit/ Delete Code Types

Description	Administrator can manage all CRUD operations for the Code_Type table
Work Products	Admin Web App;
Users/Actors	Administrator; Web Services;
Assumptions	Administrator is logged in;
Preconditions	None;
Steps	<ol style="list-style-type: none">01: Use Inputs Needed values for the Code Types02: User Clicks Submit03: User Sees Waiting Message03-01: Validation of all inputted values03-02: Request made to Web Services03-03: Response is received and information is parsed.04: Page is refreshed to show the updated content
Variations	<p>AP-03-01: If any fields are invalid, it will output error message to user.</p> <p>AP-03-02: If can't connect to database, it will timeout and return error.</p>
Postconditions	CRUD Operations is performed on Code Types
Exit Criteria	None



UC-12: Add / Edit / Delete Code

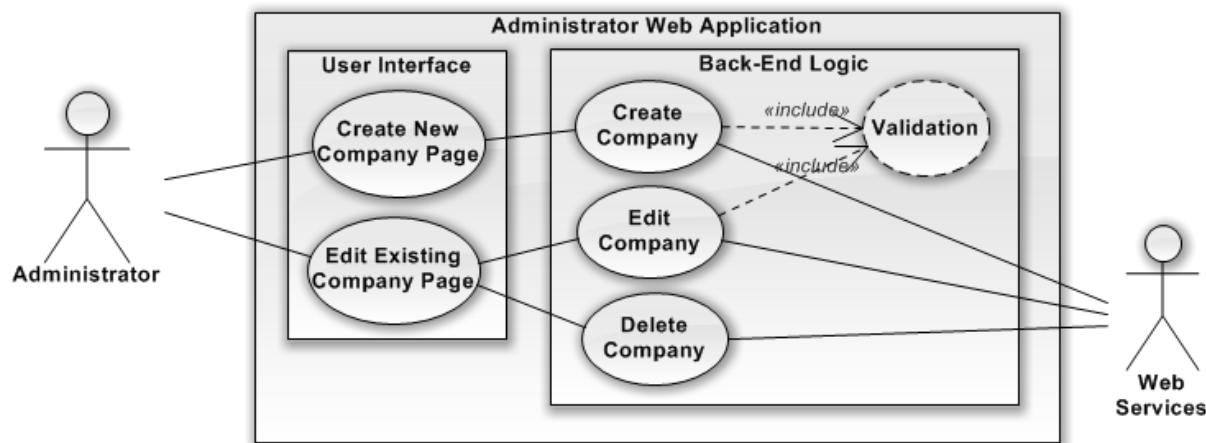


Use Case Name UC-12: Add / Edit/ Delete Code

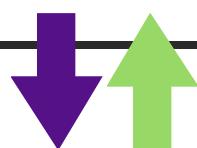
Description	The Administrator can manage all CRUD operations of the Codes Table
Work Products	Admin Web App;
Users/Actors	Administrator; Web Services;
Assumptions	Administrator is logged in;
Preconditions	None;
Steps	<p>01: Use Inputs Needed values for the Code</p> <p>02: User Clicks Submit</p> <p>03: User Sees Waiting Message</p> <p>03-01: Validation of all inputted values</p> <p>03-02: Request made to Web Services</p> <p>03-03: Response is received and information is parsed.</p> <p>04: Page is refreshed to show the updated content</p>
Variations	<p>AP-03-01: If any fields are invalid, it will output error message to user.</p> <p>AP-03-02: If can't connect to database, it will timeout and return error.</p>
Postconditions	CRUD Operations is performed on Codes
Exit Criteria	None



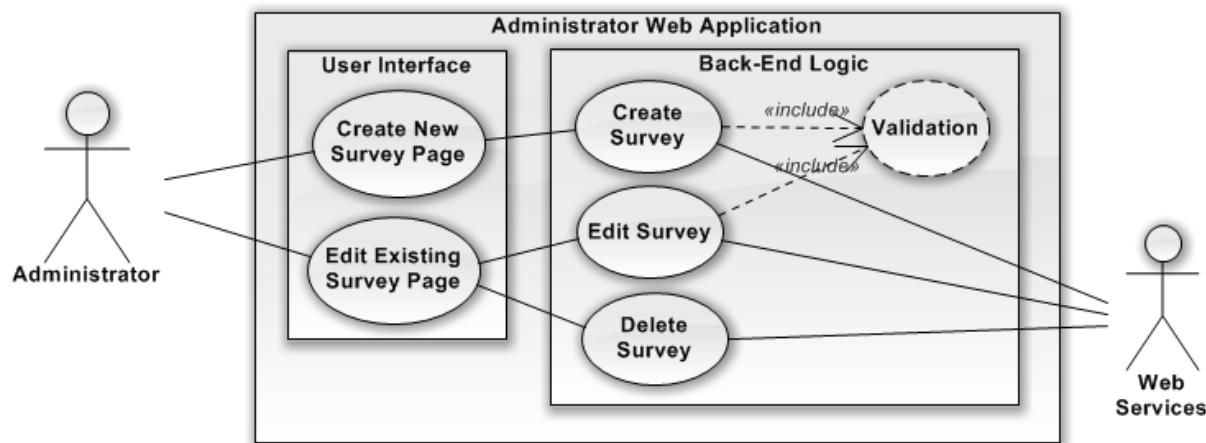
UC-13: Add / Edit / Delete Company



Use Case Name	
Description	The Administrator can manage all CRUD operations of the Company Table
Work Products	Admin Web App;
Users/Actors	Administrator; Web Services;
Assumptions	Administrator is logged in;
Preconditions	None;
Steps	01: Use Inputs Needed values for the Company 02: User Clicks Submit 03: User Sees Waiting Message 03-01: Validation of all inputted values 03-02: Request made to Web Services 03-03: Response is received and information is parsed. 04: Page is refreshed to show the updated content
Variations	AP-03-01: If any fields are invalid, it will output error message to user. AP-03-02: If can't connect to database, it will timeout and return error.
Postconditions	CRUD Operations is performed on Company
Exit Criteria	None

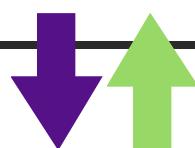


UC-14: Add / Edit / Delete Survey

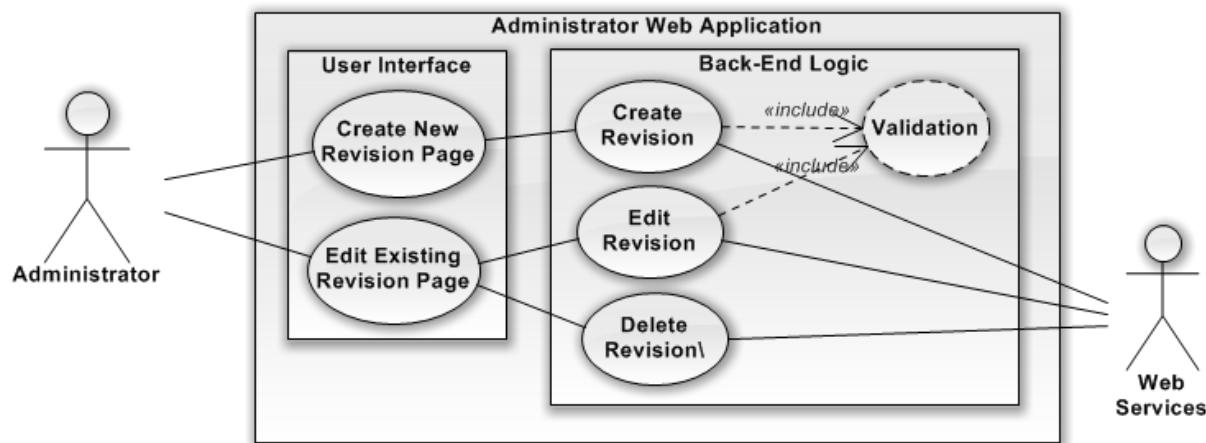


Use Case Name UC-14: Add / Edit/ Delete Survey

Description	The Administrator can manage all CRUD operations of the Survey Table
Work Products	Admin Web App;
Users/Actors	Administrator; Web Services;
Assumptions	Administrator is logged in;
Preconditions	None;
Steps	01: Use Inputs Needed values for the Survey 02: User Clicks Submit 03: User Sees Waiting Message 03-01: Validation of all inputted values 03-02: Request made to Web Services 03-03: Response is received and information is parsed. 04: Page is refreshed to show the updated content
Variations	AP-03-01: If any fields are invalid, it will output error message to user. AP-03-02: If can't connect to database, it will timeout and return error.
Postconditions	CRUD Operations is performed on Survey
Exit Criteria	None

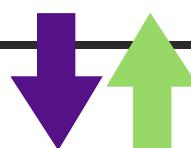


UC-15: Add / Edit / Delete Revision

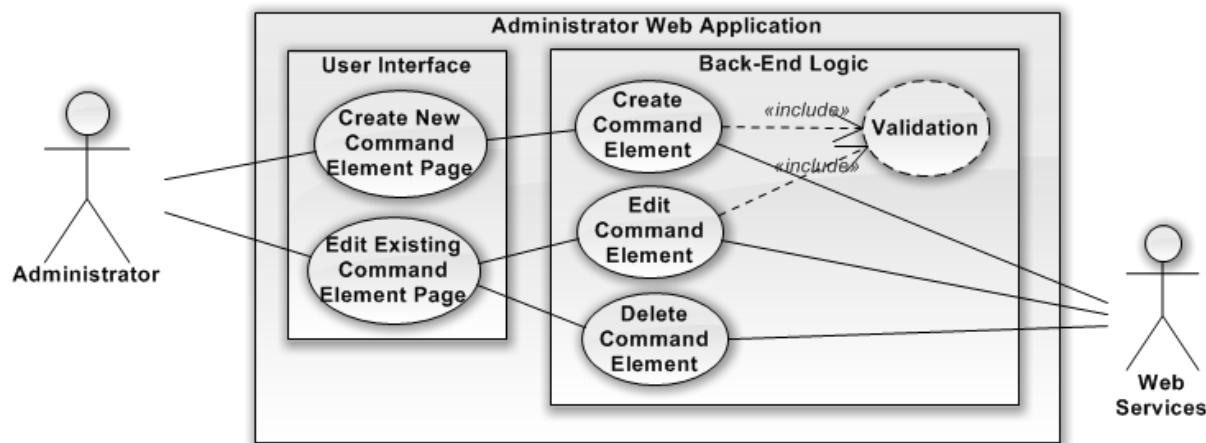


Use Case Name UC-15: Add / Edit/ Delete Revision

Description	The Administrator can manage all CRUD operations of the Revision Table
Work Products	Admin Web App;
Users/Actors	Administrator; Web Services;
Assumptions	Administrator is logged in;
Preconditions	None;
Steps	<p>01: Use Inputs Needed values for the Revision</p> <p>02: User Clicks Submit</p> <p>03: User Sees Waiting Message</p> <p>03-01: Validation of all inputted values</p> <p>03-02: Request made to Web Services</p> <p>03-03: Response is received and information is parsed.</p> <p>04: Page is refreshed to show the updated content</p>
Variations	<p>AP-03-01: If any fields are invalid, it will output error message to user.</p> <p>AP-03-02: If can't connect to database, it will timeout and return error.</p>
Postconditions	CRUD Operations is performed on Revision
Exit Criteria	None

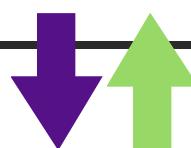


UC-16: Add / Edit / Delete Command Element

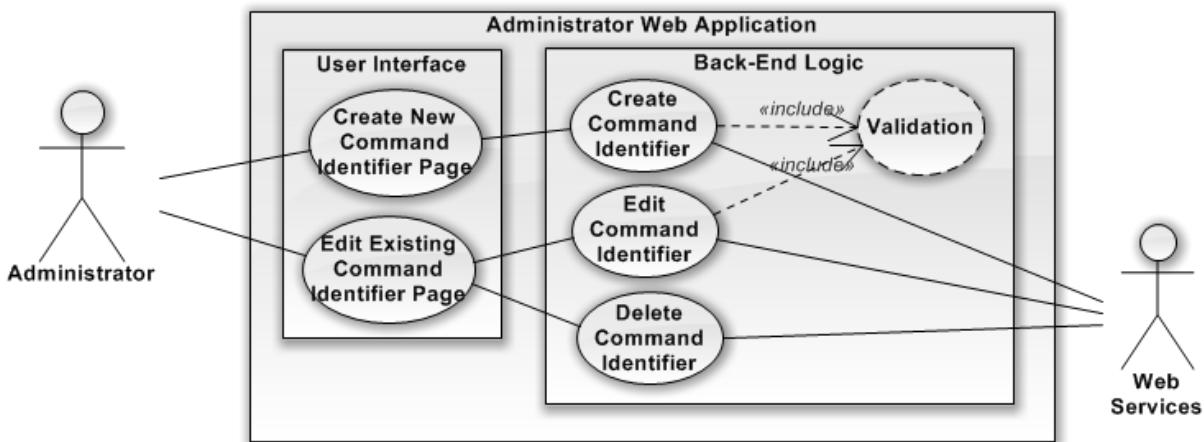


Use Case Name UC-16: Add / Edit/ Delete Command Element

Description	The Administrator can manage all CRUD operations of the Command Element Table
Work Products	Admin Web App;
Users/Actors	Administrator; Web Services;
Assumptions	Administrator is logged in;
Preconditions	None;
Steps	<p>01: Use Inputs Needed values for the Command Element</p> <p>02: User Clicks Submit</p> <p>03: User Sees Waiting Message</p> <p>03-01: Validation of all inputted values</p> <p>03-02: Request made to Web Services</p> <p>03-03: Response is received and information is parsed.</p> <p>04: Page is refreshed to show the updated content</p>
Variations	<p>AP-03-01: If any fields are invalid, it will output error message to user.</p> <p>AP-03-02: If can't connect to database, it will timeout and return error.</p>
Postconditions	CRUD Operations is performed on Command Element
Exit Criteria	None

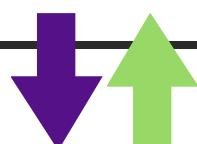


UC-17: Add / Edit / Delete Command Identifier

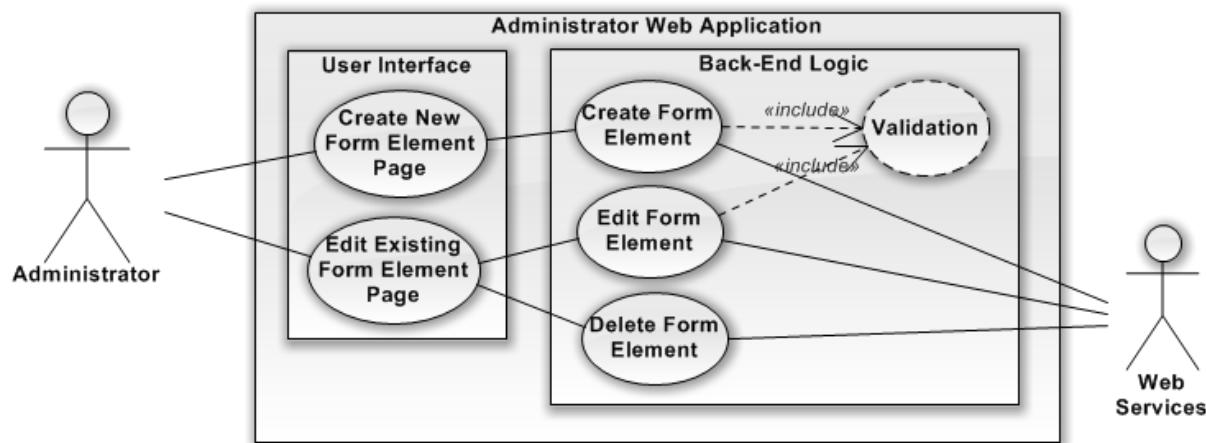


Use Case Name UC-17: Add / Edit/ Delete Command Identifier

Description	The Administrator can manage all CRUD operations of the Command Identifier Table
Work Products	Admin Web App;
Users/Actors	Administrator; Web Services;
Assumptions	Administrator is logged in;
Preconditions	None;
Steps	<p>01: Use Inputs Needed values for the Command Identifier</p> <p>02: User Clicks Submit</p> <p>03: User Sees Waiting Message</p> <p>03-01: Validation of all inputted values</p> <p>03-02: Request made to Web Services</p> <p>03-03: Response is received and information is parsed.</p> <p>04: Page is refreshed to show the updated content</p>
Variations	<p>AP-03-01: If any fields are invalid, it will output error message to user.</p> <p>AP-03-02: If can't connect to database, it will timeout and return error.</p>
Postconditions	CRUD Operations is performed on Command Identifier
Exit Criteria	None

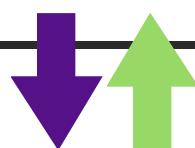


UC-18: Add / Edit / Delete Form Element

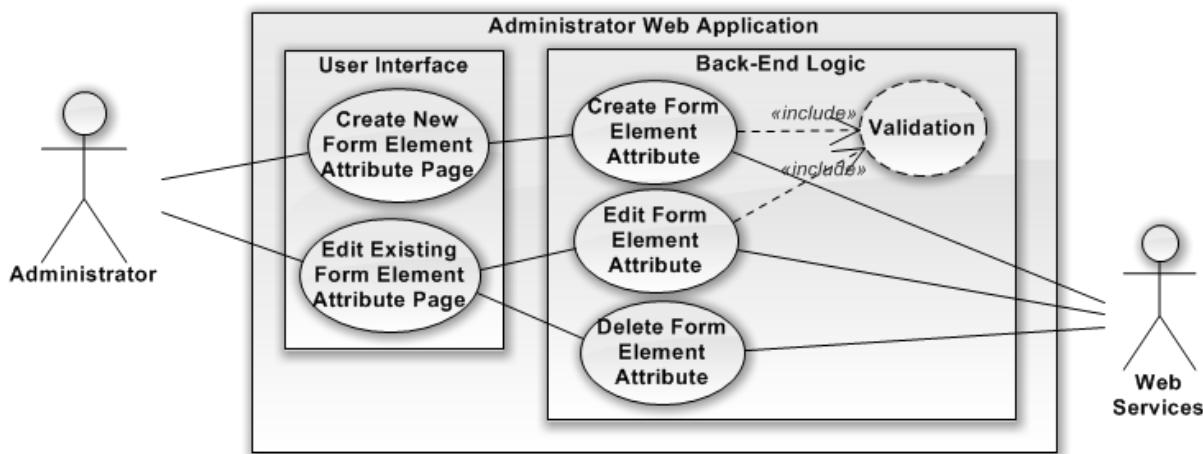


Use Case Name UC-18: Add / Edit/ Delete Form Element

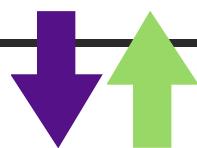
Description	The Administrator can manage all CRUD operations of the Form Element Table
Work Products	Admin Web App;
Users/Actors	Administrator; Web Services;
Assumptions	Administrator is logged in;
Preconditions	None;
Steps	01: Use Inputs Needed values for the Form Element 02: User Clicks Submit 03: User Sees Waiting Message 03-01: Validation of all inputted values 03-02: Request made to Web Services 03-03: Response is received and information is parsed. 04: Page is refreshed to show the updated content
Variations	AP-03-01: If any fields are invalid, it will output error message to user. AP-03-02: If can't connect to database, it will timeout and return error.
Postconditions	CRUD Operations is performed on Form Element
Exit Criteria	None



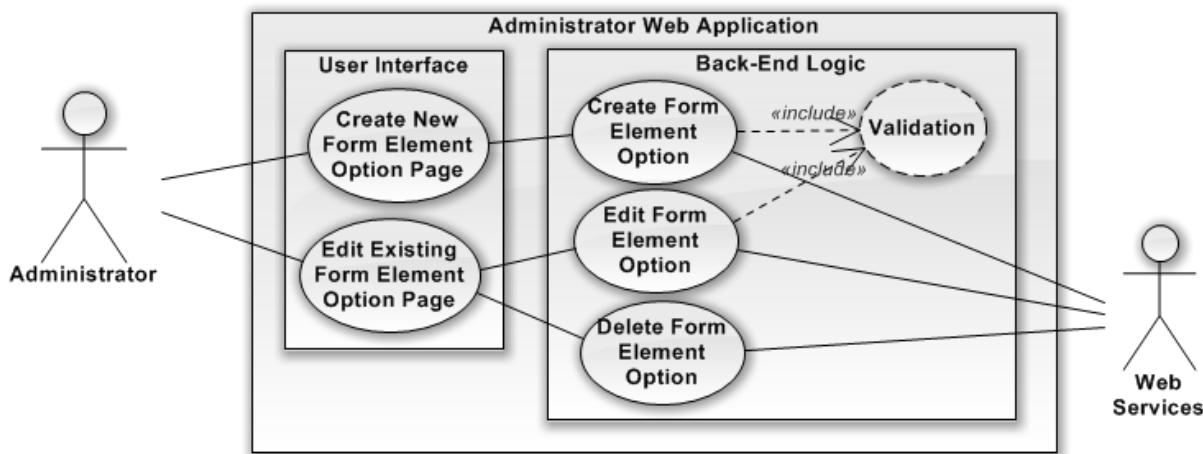
UC-19: Add / Edit / Delete Form Element Attribute



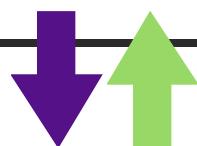
Use Case Name		UC-19: Add / Edit/ Delete Form Element Attribute
Description	The Administrator can manage all CRUD operations of the Form Element Attribute Table	
Work Products	Admin Web App;	
Users/Actors	Administrator; Web Services;	
Assumptions	Administrator is logged in;	
Preconditions	None;	
Steps	01: Use Inputs Needed values for the Form Element Attribute 02: User Clicks Submit 03: User Sees Waiting Message 03-01: Validation of all inputted values 03-02: Request made to Web Services 03-03: Response is received and information is parsed. 04: Page is refreshed to show the updated content	
Variations	AP-03-01: If any fields are invalid, it will output error message to user. AP-03-02: If can't connect to database, it will timeout and return error.	
Postconditions	CRUD Operations is performed on Form Element Attribute	
Exit Criteria	None	



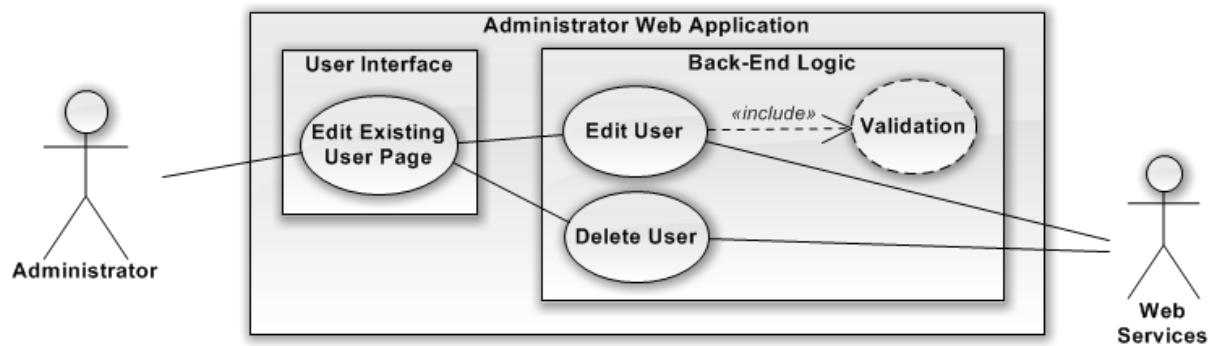
UC-20: Add / Edit / Delete Form Element Option



Use Case Name		UC-20: Add / Edit/ Delete Form Element Option
Description	The Administrator can manage all CRUD operations of the Form Element Option Table	
Work Products	Admin Web App;	
Users/Actors	Administrator; Web Services;	
Assumptions	Administrator is logged in;	
Preconditions	None;	
Steps	01: Use Inputs Needed values for the Form Element Option 02: User Clicks Submit 03: User Sees Waiting Message 03-01: Validation of all inputted values 03-02: Request made to Web Services 03-03: Response is received and information is parsed. 04: Page is refreshed to show the updated content	
Variations	AP-03-01: If any fields are invalid, it will output error message to user. AP-03-02: If can't connect to database, it will timeout and return error.	
Postconditions	CRUD Operations is performed on Form Element Option	
Exit Criteria	None	

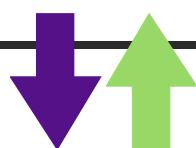


UC-21: Add / Edit / Delete User



Use Case Name UC-21: Add / Edit/ Delete User

Description	The Administrator can manage all CRUD operations of the User Table
Work Products	Admin Web App;
Users/Actors	Administrator; Web Services;
Assumptions	Administrator is logged in;
Preconditions	None;
Steps	<p>01: Use Inputs Needed values for the User</p> <p>02: User Clicks Submit</p> <p>03: User Sees Waiting Message</p> <p>03-01: Validation of all inputted values</p> <p>03-02: Request made to Web Services</p> <p>03-03: Response is received and information is parsed.</p> <p>04: Page is refreshed to show the updated content</p>
Variations	<p>AP-03-01: If any fields are invalid, it will output error message to user.</p> <p>AP-03-02: If can't connect to database, it will timeout and return error.</p>
Postconditions	CRUD Operations is performed on User
Exit Criteria	None



WORK BREAKDOWN STRUCTURE

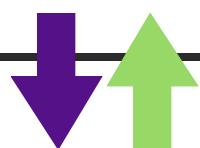
8/23/2013

Graduate Capstone



Table of Contents

0 Introduction	3
0.1 References.....	3
0.2 Revision History	3
1. Project.....	4
1.1. Pre-Project	4
1.1.1. Research.....	4
1.1.2. Project Setup	4
1.1.3. Brainstorm Ideas	4
1.2. Inception.....	4
1.3. Design.....	5
1.3.1. Overall System.....	5
1.3.2. Database.....	5
1.3.3. Web Services.....	5
1.3.4. Phone Development.....	5
1.4. Development.....	5
1.4.1. Database.....	5
1.4.2. Web Services.....	5
1.4.3. Administrator Website.....	5
1.4.4. Phone OS.....	5
1.5. Testing	6
1.5.1. Integration Testing	6
1.5.2. Performance Testing	6
1.5.3. User Acceptance Testing	6
1.5.4. Web Services.....	6
1.5.5. Web Admin	6
1.5.6. Phones	6
1.6. Release	7
1.7. Maintenance.....	7



0 Introduction

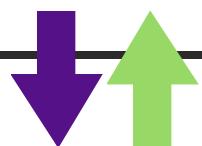
The purpose of this document is to show a detailed structure of the tasks in the project.

0.1 References

- <http://www.harborlightmanagement.com/Publications/WBS%20outline.pdf>

0.2 Revision History

Name	Date	Reason For Change	Version
Andy Bottom	04/16/2013	Created Work Breakdown Structure	1.0



1. Project

1.1. Pre-Project

1.1.1. Research

- 1.1.1.1. Mobile Development
- 1.1.1.2. Implementation
- 1.1.1.3. Restful Web Services

1.1.2. Project Setup

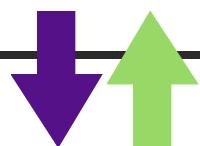
- 1.1.2.1. Software
- 1.1.2.2. Web Server
- 1.1.2.3. Database
- 1.1.2.4. Obtain Licenses

1.1.3. Brainstorm Ideas

- 1.1.3.1. Choose Project
- 1.1.3.2. Research
- 1.1.3.3. Market Need

1.2. Inception

- 1.2.1. Vision and Scope Document
- 1.2.2. System Scope
- 1.2.3. Risks
- 1.2.4. Work Breakdown Structure
- 1.2.5. Setup Website Documentation
- 1.2.6. Test Plan
- 1.2.7. User Experience Guide
- 1.2.8. User Manuals
- 1.2.9. Developer Guide



1.3. Design

1.3.1. Overall System

1.3.2. Database

- 1.3.2.1. ERD
- 1.3.2.2. Structure
- 1.3.2.3. Tables

1.3.3. Web Services

- 1.3.3.1. Structure
- 1.3.3.2. Diagram

1.3.4. Phone Development

- 1.3.4.1. Class Diagram
- 1.3.4.2. Design Pattern

1.4. Development

1.4.1. Database

- 1.4.1.1. Create Tables
- 1.4.1.2. Implement SQL

1.4.2. Web Services

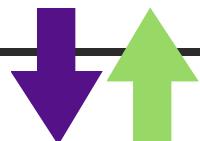
- 1.4.2.1. Implement Database Query
- 1.4.2.2. Implement URI Mapping

1.4.3. Administrator Website

- 1.4.3.1. Classes
- 1.4.3.2. Code-Behind
- 1.4.3.3. User Interface

1.4.4. Phone OS

- 1.4.4.1. Backend Code (Cross Platform)
- 1.4.4.2. Windows Phone UI / Hookup
- 1.4.4.3. Android UI / Hookup
- 1.4.4.4. iOS UI / Hookup



1.5. Testing

1.5.1. Integration Testing

- 1.5.1.1. Web Service to Database
- 1.5.1.2. Admin CMS to Web Services
- 1.5.1.3. Web Services to Web Driver
- 1.5.1.4. PCL to Web Services
- 1.5.1.5. Windows Phone to PCL
- 1.5.1.6. Android Phone to PCL

1.5.2. Performance Testing

- 1.5.2.1. Database Connections
- 1.5.2.2. Recursive Database Calls
- 1.5.2.3. Web Service Response Time
- 1.5.2.4. Web Service JSON File Size
- 1.5.2.5. PCL Async Response Time

1.5.3. User Acceptance Testing

- 1.5.3.1. Web Admin CMS
- 1.5.3.2. Windows Phone
- 1.5.3.3. Android Phone

1.5.4. Web Services

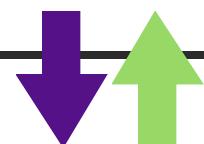
- 1.5.4.1. Set up Correctly
- 1.5.4.2. Expecting correct logic

1.5.5. Web Admin

- 1.5.5.1. Valid Data Checking

1.5.6. Phones

- 1.5.6.1. Backend Logic Correct
- 1.5.6.2. Handle data effectively
- 1.5.6.3. Effect UI / Self Intuitive
- 1.5.6.4. Data Validation

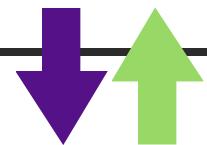


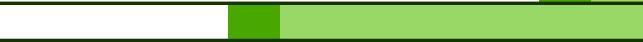
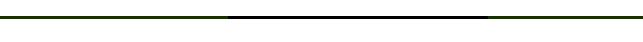
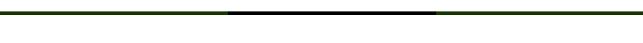
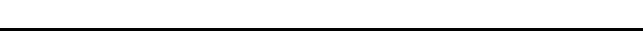
1.6. Release

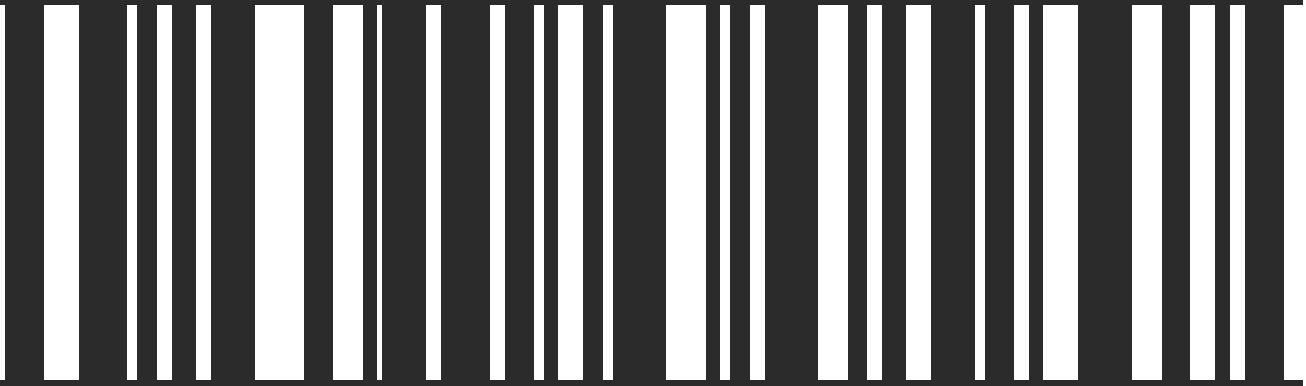
- 1.6.1. Effective Deployment
- 1.6.2. Out to Marketplace

1.7. Maintenance

- 1.7.1. Bug Fixes
- 1.7.2. Feature Updates



43	1.4.4	Phone Application	
44	1.4.4.1	Binding Code	
45	1.4.4.2	User Interface and Hookup	
46	1.4.4.2.1	Windows Phone	
47	1.4.4.2.2	Android	
48	1.4.4.2.3	iOS	
93	1.4.5	PCL	
94	1.4.5.1	Classes	
95	1.4.5.2	View Models	
96	1.4.5.3	Async for Web Services	
49	1.5	Testing	
80	1.5.1	Integration Testing	
81	1.5.1.1	Web Service To Database	
82	1.5.1.2	Admin CMS to Web Services	
83	1.5.1.3	Web Services to Web Driver	
84	1.5.1.4	PCL to Web Services	
85	1.5.1.5	Windows Phone to PCL	
86	1.5.1.6	Android Phone to PCL	
87	1.5.2	Performance Testing	
88	1.5.2.1	Database Connections	
89	1.5.2.2	Recursive Database Calls	
90	1.5.2.3	Web Service Response Time	
91	1.5.2.4	Web Service JSON File Size	
92	1.5.2.5	PCL Async Response Time	
50	1.5.3	Database	
51	1.5.3.1	Data Stored Correctly	
52	1.5.3.2	Data Integrity	
54	1.5.4	Web Services	
55	1.5.4.1	Set Up Correctly	
56	1.5.4.2	Expected Logic	
58	1.5.5	Administrator Website	
60	1.5.5.1	Validate Data Input	
62	1.5.6	Phones	
63	1.5.6.1	Back-End Logic Correct	
65	1.5.6.2	Handle Data Effectively	
67	1.5.6.3	Effective UI / Self Intuitive	
68	1.5.6.4	Data Validation	
69	1.6	Release	
70	1.6.1	Effective Deployment	
71	1.6.2	Deployed Successfully to Stores	
72	1.7	Maintainence	
73	1.7.1	Bug Fixes	
74	1.7.2	Feature Updates	



SOFTWARE DESIGN SPECIFICATION

8/23/2013

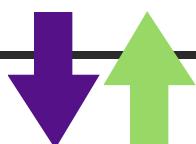
Graduate Capstone

RECEIPT

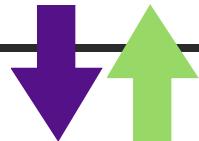
REWARDS

Table of Contents

1 Introduction	4
1.1 Intended Audience	4
1.2 References	4
1.3 Revision History.....	4
2 System Overview	5
2.1 Section Overview	5
2.2 General Constraints	5
2.3 Data Design.....	5
2.4 Actors.....	5
2.4.1 Phone	5
2.4.2 Admin Website.....	5
2.4.3 Database	6
2.4.4 Web Services	6
2.4.5 Survey Website	6
2.5 Domain Model.....	7
2.6 Goals and Guidelines	7
3 Architecture Design	8
3.1 Section Overview	8
3.2 Architectural Strategies.....	8
3.3 System Architecture.....	9
3.3.1 Web Services	9
3.3.2 Automation Process.....	14
3.3.3 Admin CMS	14
3.3.4 Phone Application	15
3.3.5 Reverse Proxy	22
3.4 Detailed System Design	23
4 User Interface Design	24
4.1 Section Overview	24
4.2 GUI Components	24
4.3 Prototype	24
5 Design decisions and tradeoffs	25
5.1 Web Services.....	25
5.1.1 SOAP.....	25
5.1.2 RESTful.....	26



5.1.3 Decision	26
5.2 Mobile Cross-Platform Development	27
5.2.1 Hybrid App.....	27
5.2.2 Xamarin	28
5.2.3 Choice	28
5.3 Mobile Cross-Platform Design Pattern	29
5.3.1 Model View View-Model (MVVM).....	29
5.3.2 Portable Class Library (PCL).....	29
5.4 Caching.....	30
5.4.1 Reverse Proxy	30
5.4.2 Phone Caching.....	30
5.5 Database Access	31
5.1 Data Access Layer through Web Services	31
6 Pseudocode for System	32
6.1 Admin CMS	32
6.1.1 Element Manager Page (Back-End).....	32
6.1.2 Element Implementation.....	34
6.2 Web Services.....	37
6.2.1 Service.....	37
6.2.2 Implementation	38



1 Introduction

The purpose of this document is to provide description regarding the architecture and other design aspects of the system. The document will be used by the development team to develop the functionality. Clients can also use this to verify the agreed upon functionality.

1.1 Intended Audience

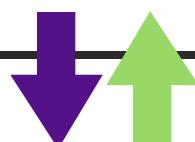
The document is intended for individuals of a medium-high technical or business background.

1.2 References

- http://web.cs.dal.ca/~hawkey/3130/SDS_outline.doc
- http://www.cs.utah.edu/~jamesj/ayb2005/docs/SDS_v2.htm
- http://www.se.rit.edu/~vdkrit/design/VDK-RIT_SDS.doc

1.3 Revision History

Name	Date	Reason For Change	Version
Andy Bottom	04/28/2013	Started compiling all architecture diagrams into this document	0.1
Andy Bottom	05/07/2013	Completed the finishing touches on this document	1.0
Andy Bottom	05/21/2013	Updated and added additional design decision including PCLs and caching	1.1



2 System Overview

2.1 Section Overview

This section is used for explanations on generically high-level overview of the entire system. This includes information regarding arts of this system.

2.2 General Constraints

The [Software Requirement Specification](#) for information about general constraints of the system.

2.3 Data Design

Please refer to the [Database Design Document](#) for a further analysis of the Data and Database Design.

2.4 Actors

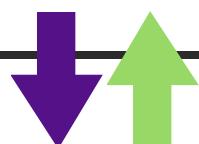
The actors are the main endpoints that utilize the functionality of the system.

2.4.1 Phone

The phones are the phone applications which the Users interact with to use the systems features.

2.4.2 Admin Website

The admin website is what the administrators interact with to manage the data in the system.



2.4.3 Database

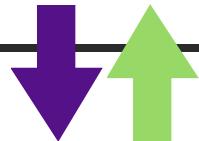
The database holds the information for the system and gets used by the other actors. All interactions with the database happen via the Web Services.

2.4.4 Web Services

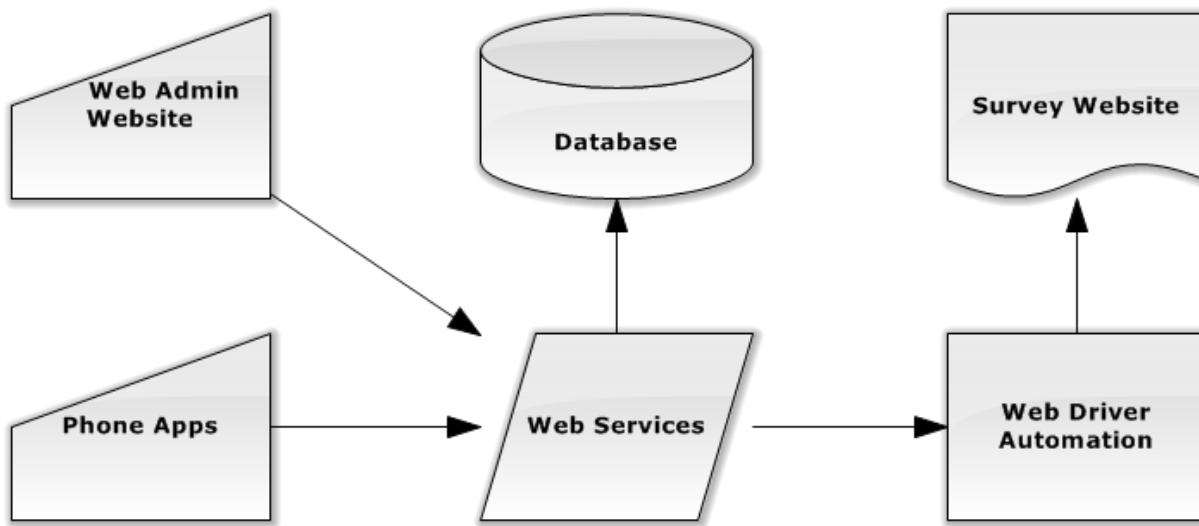
The Web Services is the way that both the Phone and Admin Website performs common logic and obtains the data. The Web Services acts as our data access layer for both the client applications to be able to get data from the database. The Web Services also acts as the kickoff for the Automation Web Driver process.

2.4.5 Survey Website

The Survey website is a pre-existing website that the company uses for the Satisfaction Surveys. Typically these websites are created by a company that offers to create these types of surveys and the origin company uses their services for this reason. It is important to note that the Company Survey Website is completely independent from our system and code base. The only interaction we have with the website is that it is the endpoint that is our Web Driver uses to perform the form automation on.

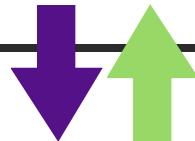


2.5 Domain Model



2.6 Goals and Guidelines

- A large emphasis is put on convenience of the entire project. The goal is to provide an easier and more convenient way to take the surveys.
- Convenience will be done via very good Usability of the Phone Application.



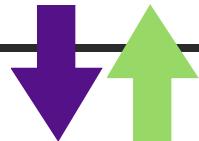
3 Architecture Design

3.1 Section Overview

The following section explains the architectural designs of the major system in the project.

3.2 Architectural Strategies

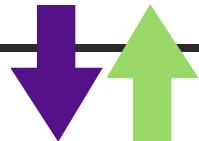
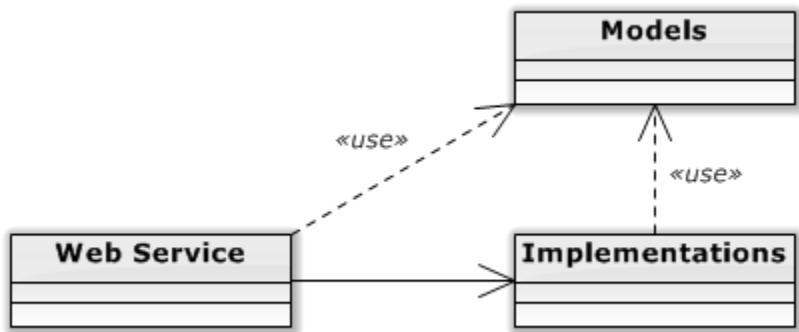
- Object Oriented Design is always the forefront of any application that I develop for. The idea of being able to get the most flexibility and reuse out of my code is always a goal that I strive for.
- The use of a DAL is implemented with the web services so that I can keep separation between the user devices and the rest of the system. It also provides reuse of all data calls.
- The automated process is also located inside the web services layer to keep abstraction between the devices and the process.
- The Web Admin site is a very object oriented approach and follows very closely along the lines of the database. The Web Admin is a Content Management System for the project.
- In the phone applications, we will create a PCL that will be referenced by the apps to incorporate a large reuse of the backend page logic.



3.3 System Architecture

3.3.1 Web Services

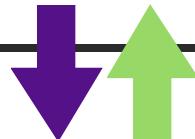
3.3.1.1 High-Level Class Diagram

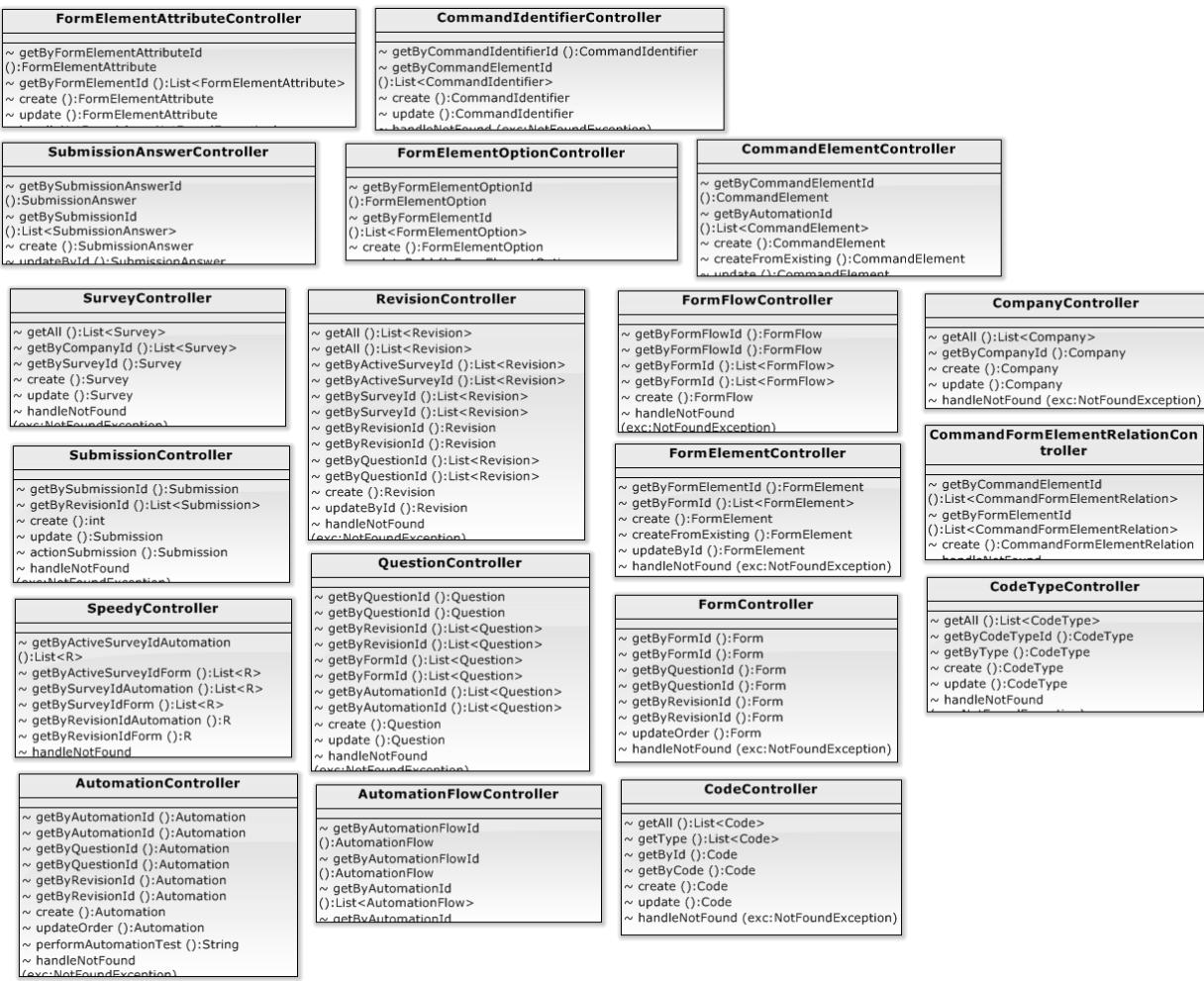


3.3.1.2 Class Diagram

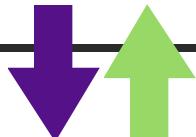


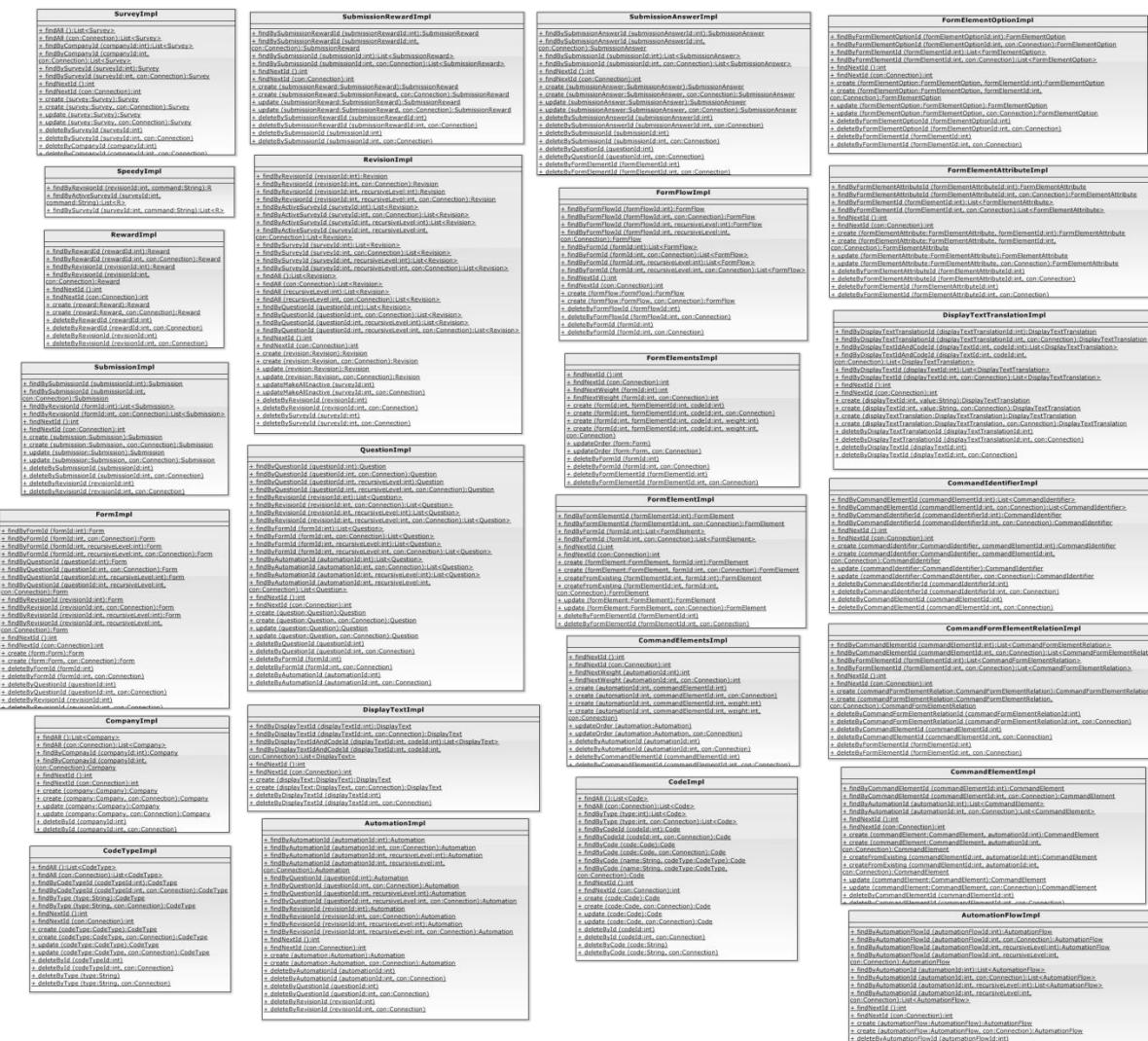
Class Diagram of the Common Package

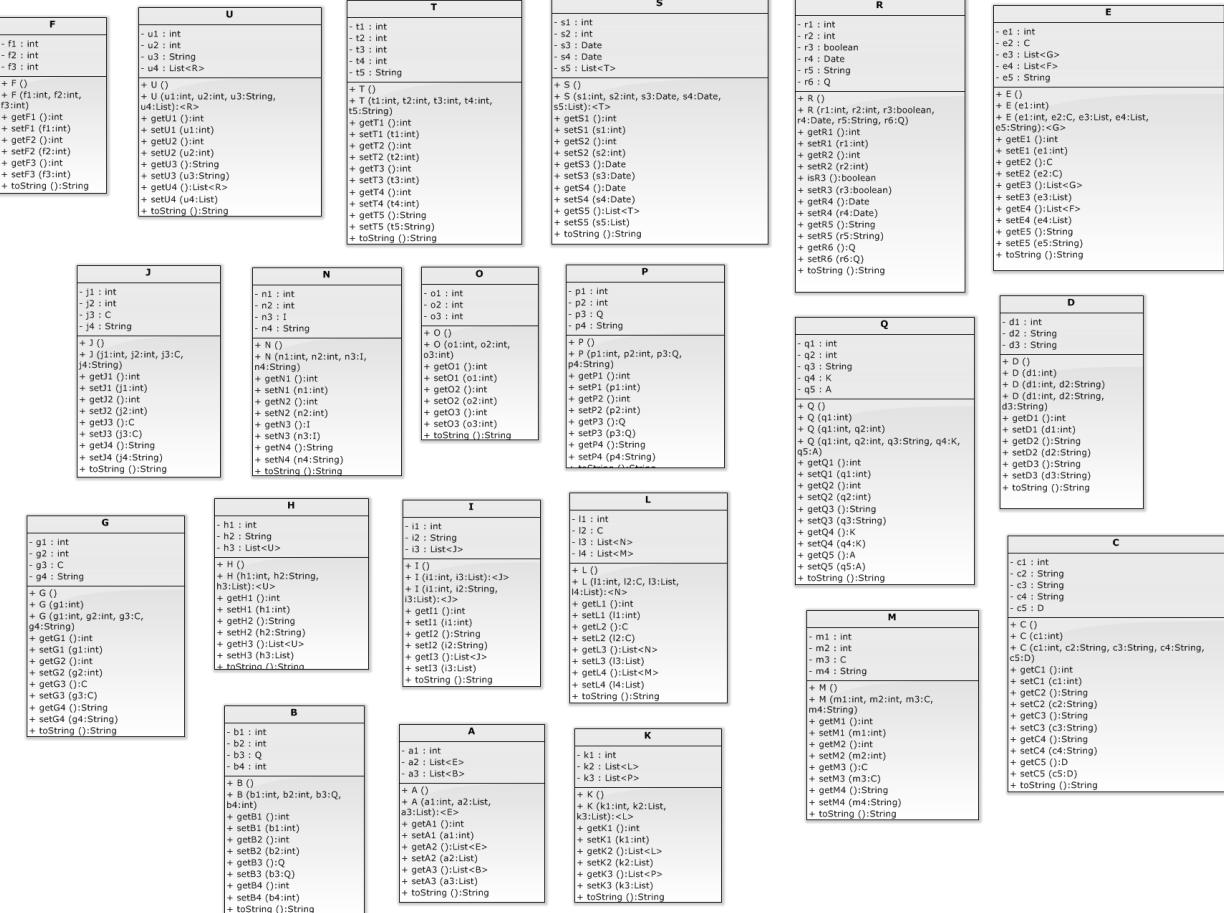




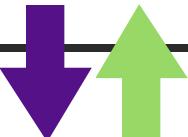
Class Diagram of the Models Package





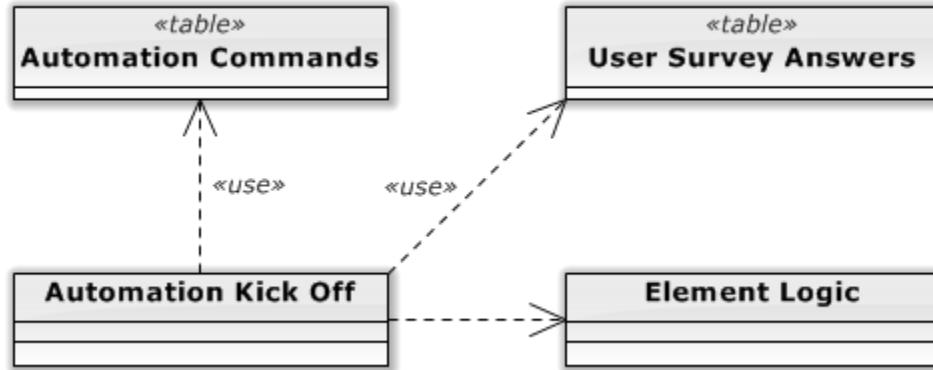


Class Diagram of the Alias Package



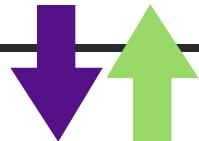
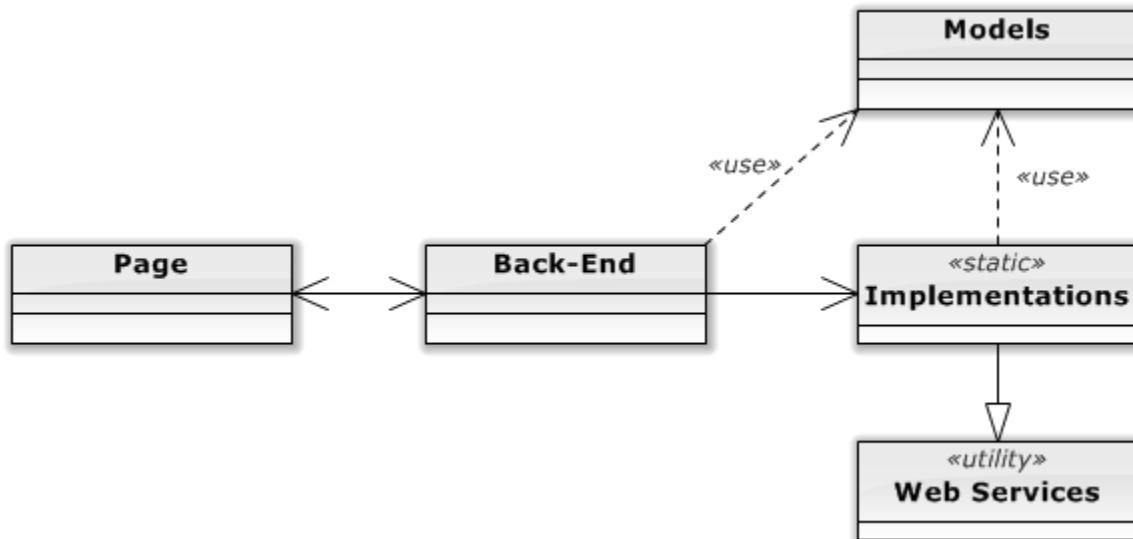
3.3.2 Automation Process

3.3.2.1 High-Level Class Diagram



3.3.3 Admin CMS

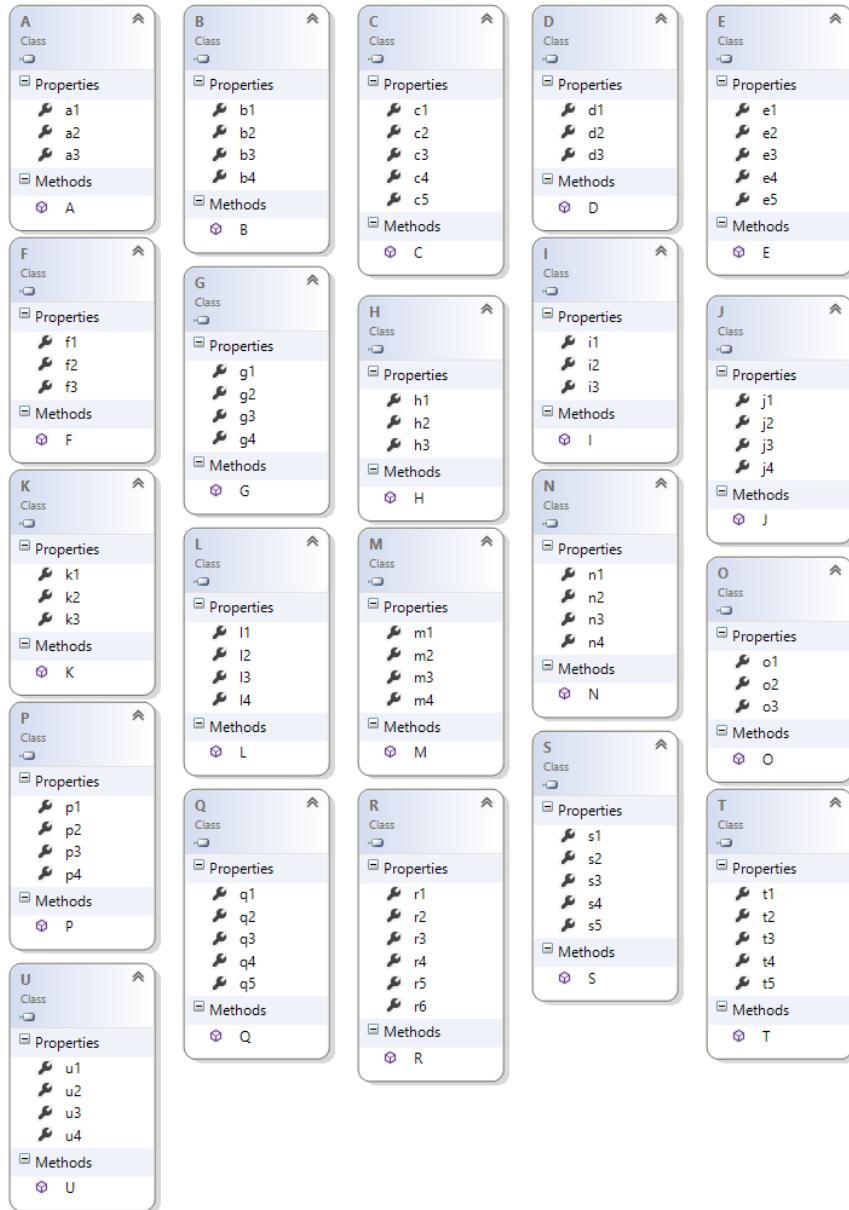
3.3.3.1 High-Level Diagram



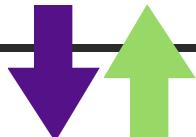
3.3.4 Phone Application

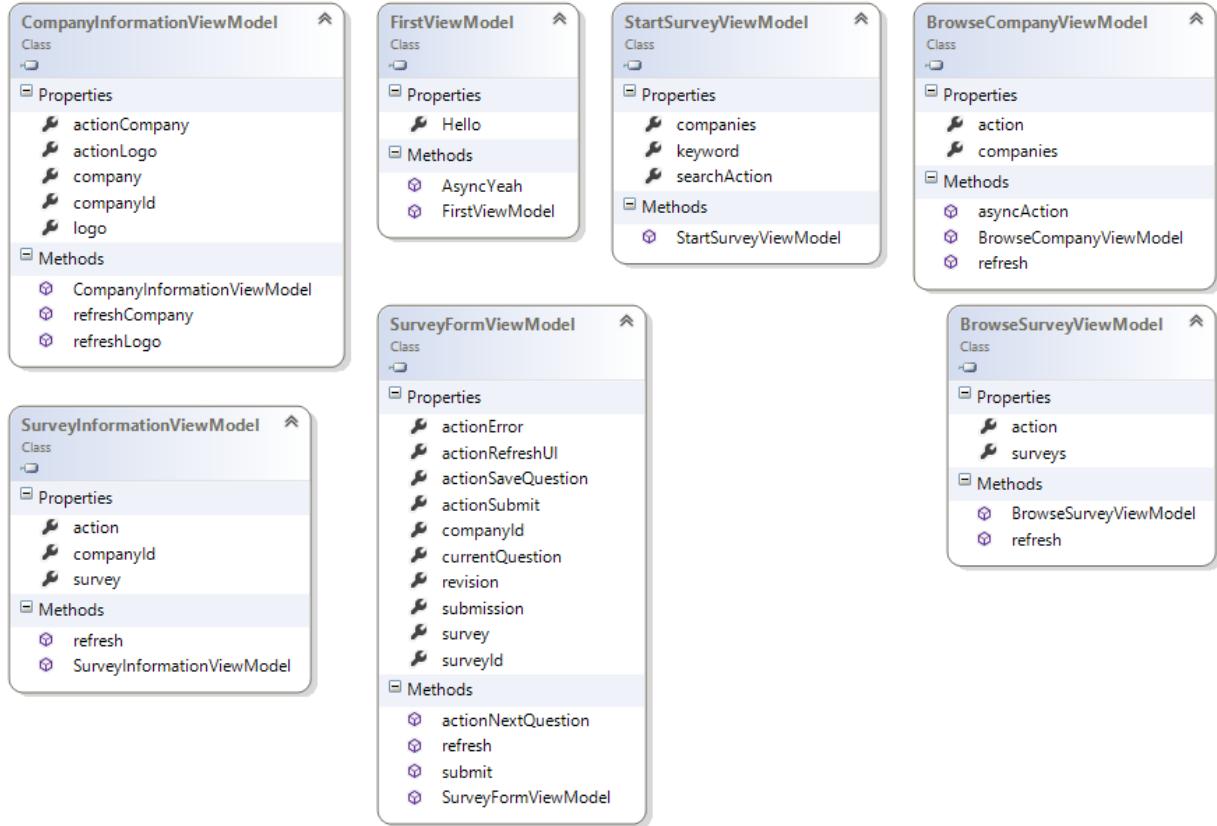
3.3.4.1 Portable Class Library (Class Diagram)

The following diagrams are the class diagrams of the classes located in the Portable Class Library.

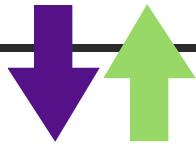


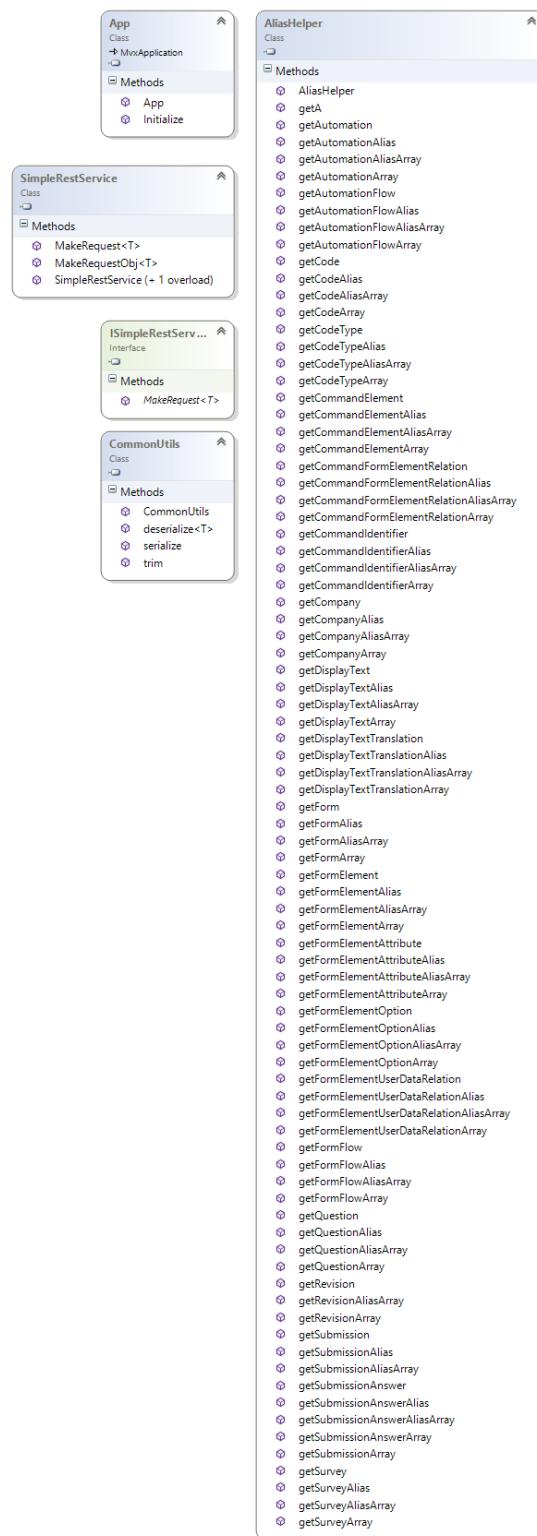
Class Diagram of the Alias Package



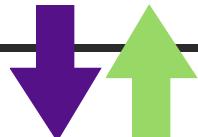


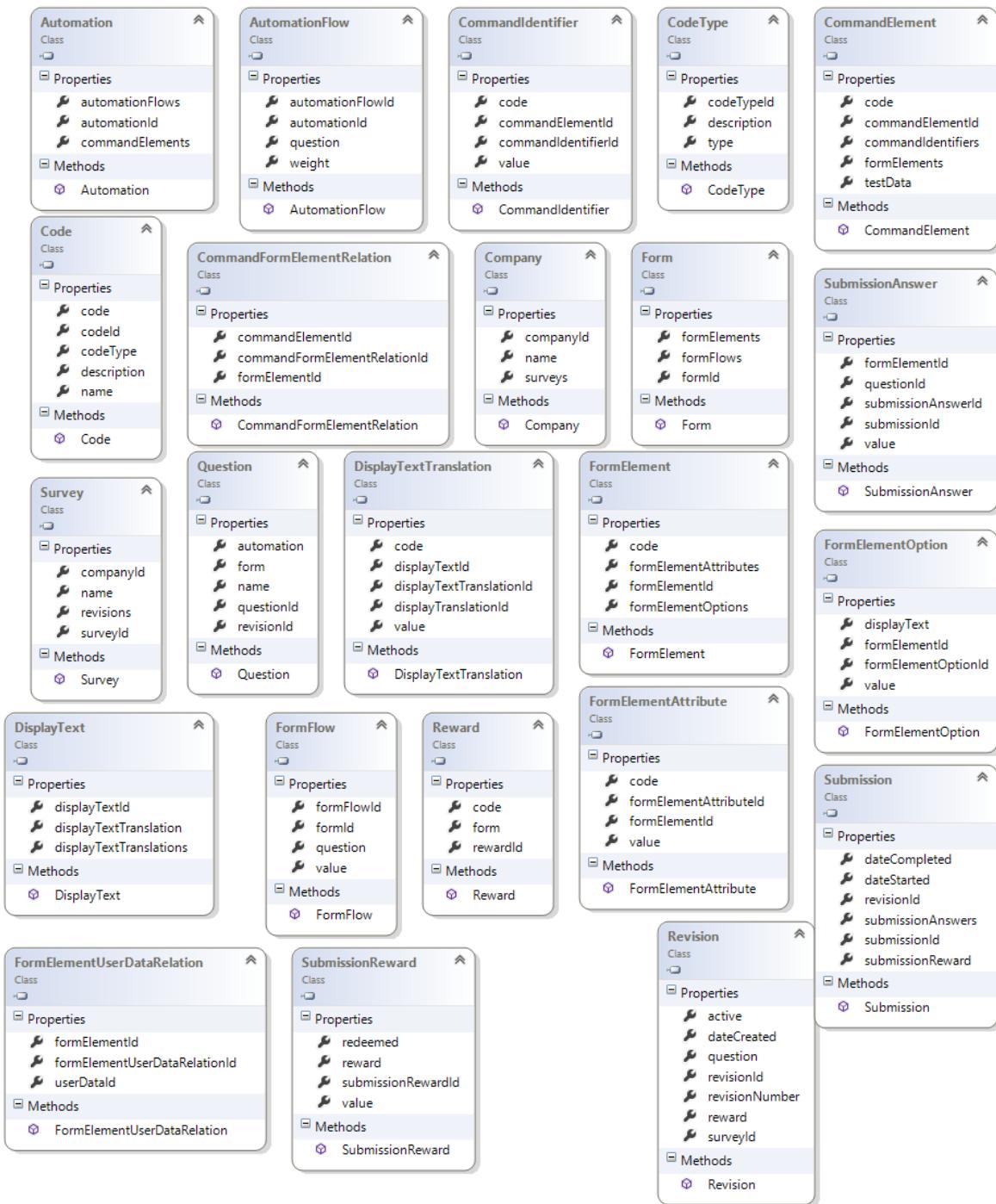
Class Diagram of the View Model Package



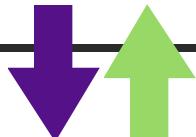


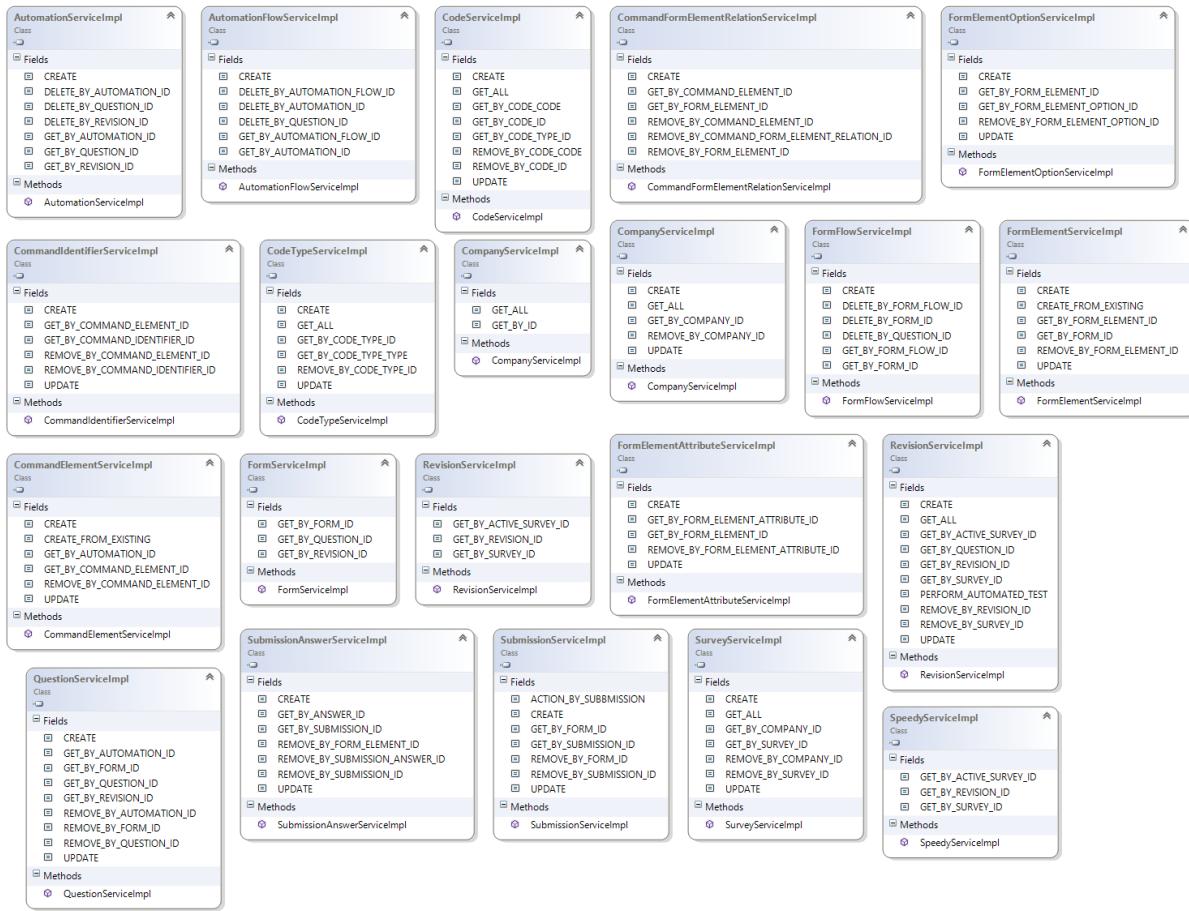
Class Diagram of the Common Package



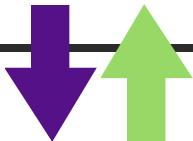


Class Diagram of the Model Package

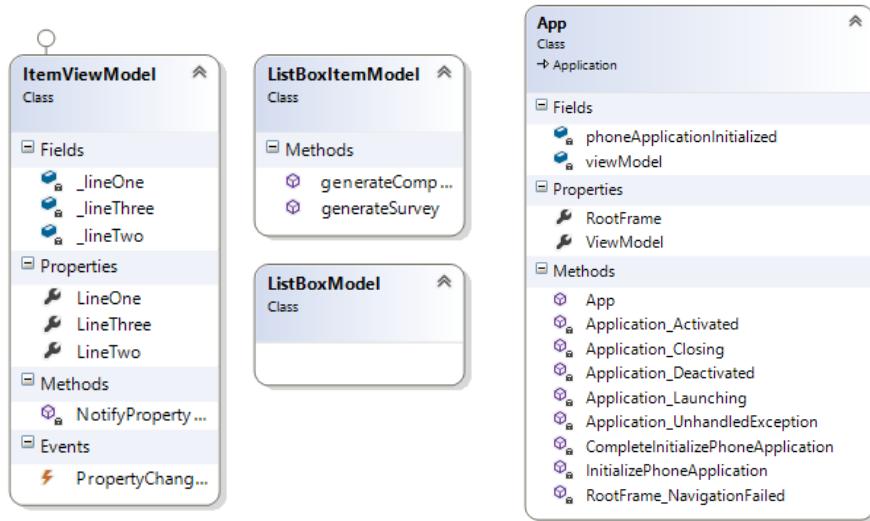




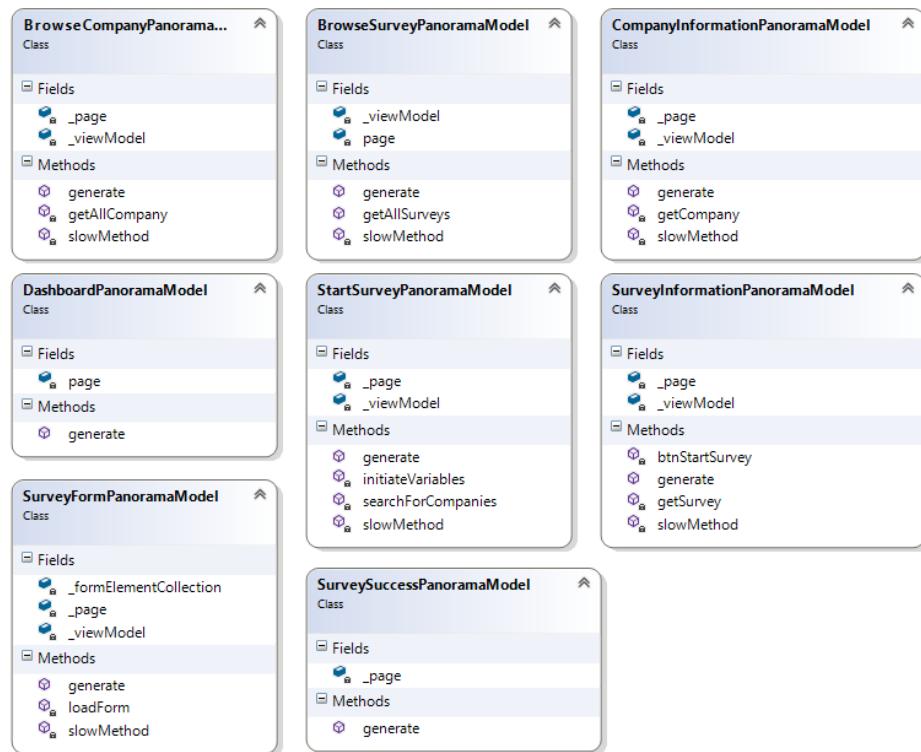
Class Diagram of the Service Access Layer Package



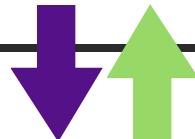
3.3.4.2 Windows Phone Application

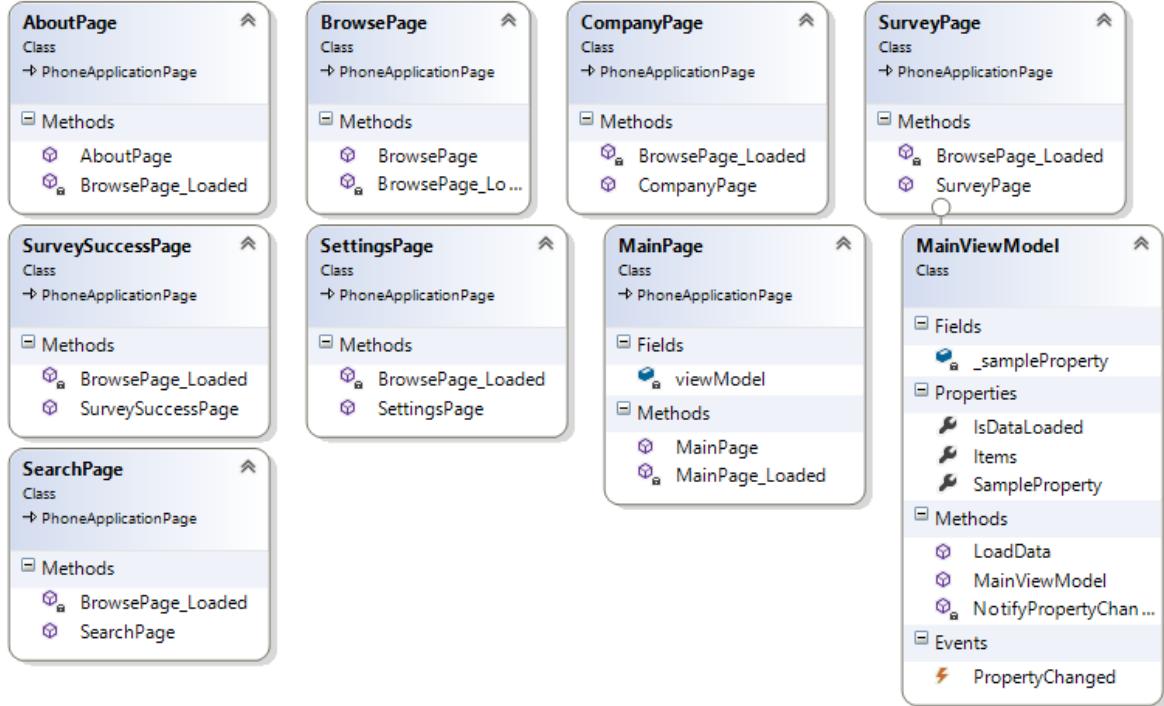


Class Diagram of the Miscellaneous Package

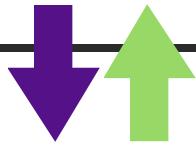


Class Diagram of the Panorama Model Package



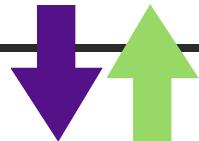
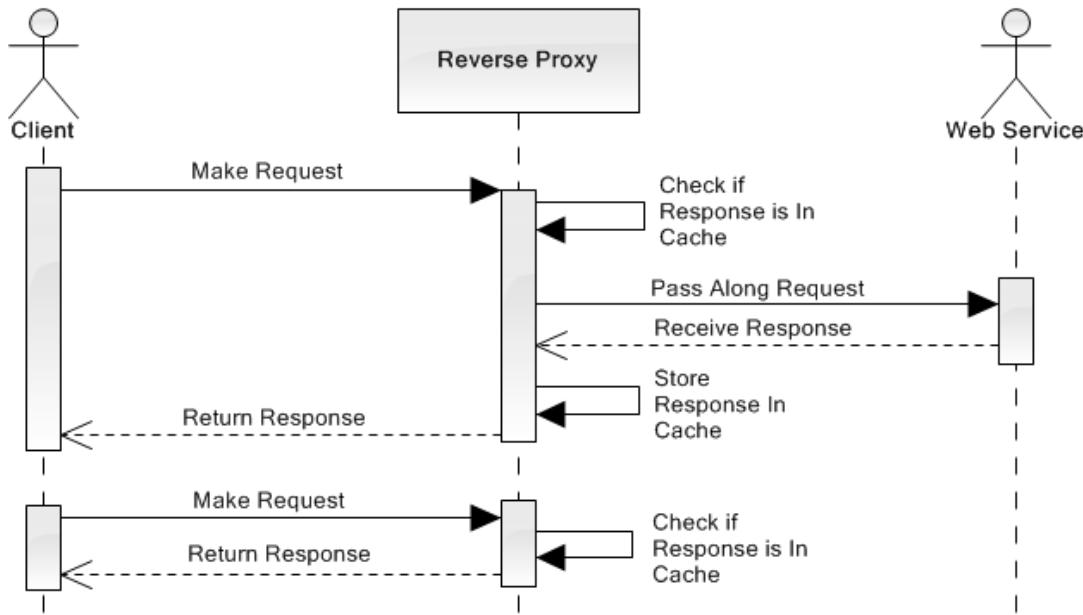


Class Diagram of the Page Package

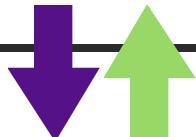
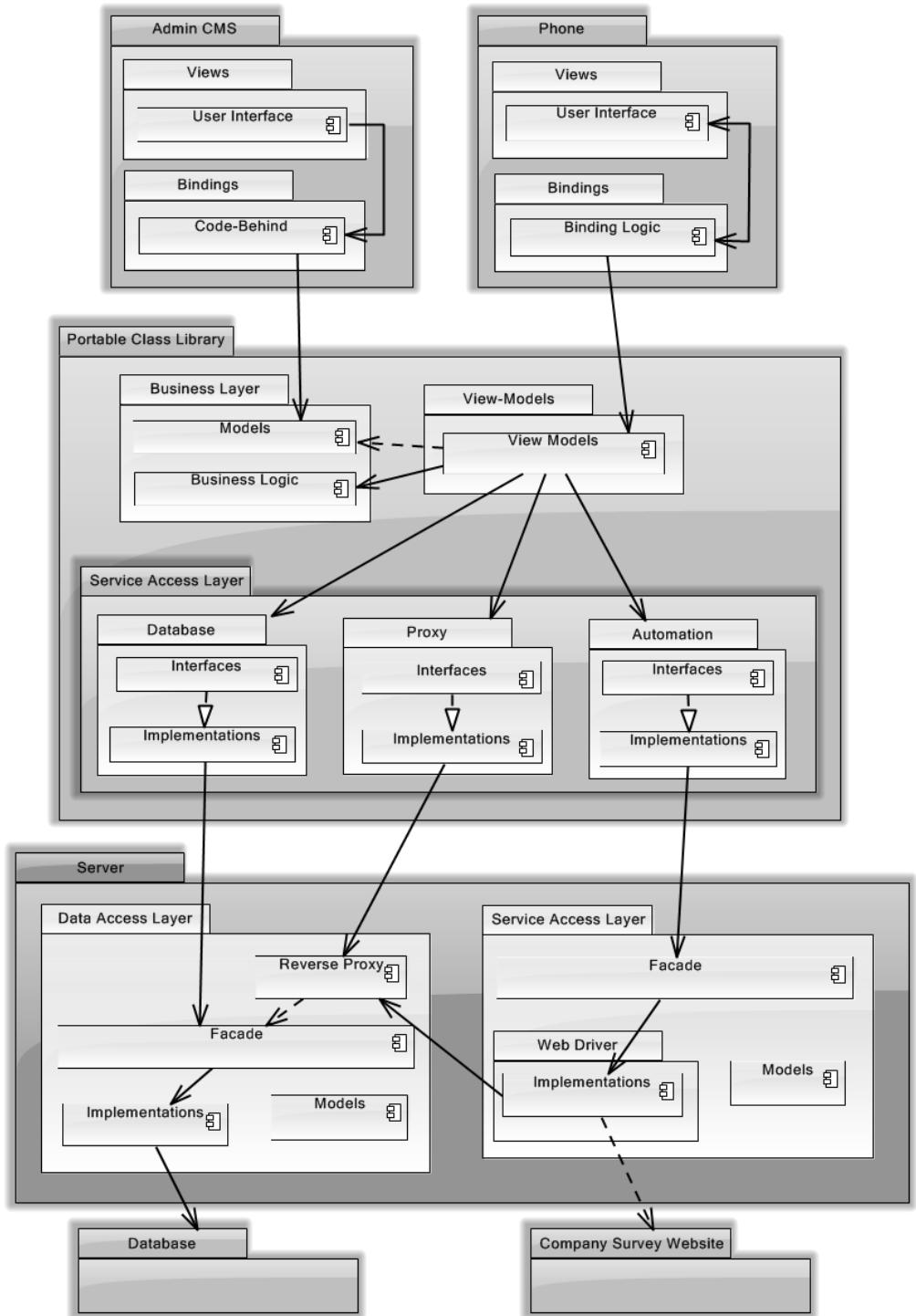


3.3.5 Reverse Proxy

3.3.5.1 Sequence Diagrams



3.4 Detailed System Design



4 User Interface Design

4.1 Section Overview

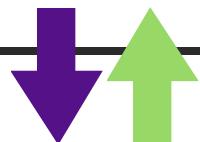
This section is used as a guide for designing the user interface for both the Admin CMS and the Phone Application. For more detailed information about the User Interface, refer to the [User Experience Guide](#).

4.2 GUI Components

Refer to the [User Experience Guide](#) about the components of the user interface.

4.3 Prototype

Please refer to the [User Experience Guide](#) for the wireframe of the phone application.



5 Design decisions and tradeoffs

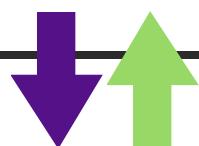
In this section, in-depth explanations containing reasoning for choices made in regard to the implementation and design of the system are explained. The options, pros and cons and overall choice and reason will be explained.

5.1 Web Services

Since web services are a vital part of our application, it was needed to be decided as to which time of web services would be implemented in the system. The choices were SOAP and RESTful. Below are a brief description of each and the choice made.

5.1.1 SOAP

Soap was the first type of web service created. I also have had experience with SOAP through an internship, so I already know the concept and how to implement them. The main part of SOAP is that it uses XML formatting and is relatively stricter in handling data and formatting. SOAP also tends to be more secure and is a better choice when using very highly sensitive data. Unfortunately, the data in our system will not be immensely sensitive.

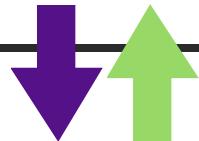


5.1.2 RESTful

After researching more about Web Services, I came upon RESTful. It seemed to me that currently, that RESTful web services seemed to be the more trendy design choice for Web Services. I thought it would be a good idea to gain experience and learn how to implement them. The conceptual flow of RESTful is extremely similar to the SOAP Web Services. The advantages for RESTful are that the data can be sent in as XML or as JSON. Also, the formatting of the system of RESTful tends to be simpler and more flexible. This causes the downside of it being less secure and not a good choice for handling highly sensitive data. Fortunately there are still other ways several designs to help make RESTful web services.. Also, one last note, which was more of a deciding factor, is that the ability to handle the data as JSON would be advantageous for mobile development because JSON is less labor intensive in compiling and parsing than the XML.

5.1.3 Decision

After weighing the pros and cons, it was decided that RESTful Web Services would be the best fit for the less labor intensive fashion of dealing with JSON and that RESTful seems to be the go to choice in the future.

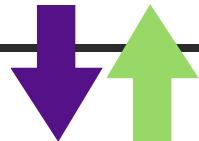


5.2 Mobile Cross-Platform Development

There was a need to develop the phone application to be cross-platform. When proposed with this problem there are relatively two directions you can take. You can implement a Web Wrapper Application or use a framework to develop the application, such as Xamarin. Below I discuss the advantages of each and the choice made.

5.2.1 Hybrid App

A hybrid app is essentially a fake app. By this I mean is that it is an application that is written with HTML, CSS and JavaScript. Essentially it is a mobile website that is then “wrapped” inside a phone application so it can be “installed” on a phone. But it isn’t interpretive like a native app, but instead uses the browser rendering technology to display. The advantage of this is that Hybrid Apps are highly multi-platform friendly as it uses Web Standards and not the phone to render. The disadvantage of this is that it doesn’t utilize the true functionality of a phone to the extent that native apps can. Also, another major disadvantage is that the UI of the app doesn’t look native. To explain further, each smartphone OS has its own unique look. Apple has the glossy rounded look whereas Windows Phone has its Metro look. A hybrid app that is developed for both of the platforms will function properly, but the User Experience of the app is very different than what the user is used to. Due to the nature of how “Web Apps” look and feel, it is evident that the application was not created for that OS. Thus, they basically don’t fit in.



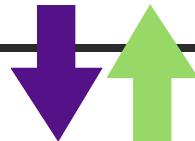
5.2.2 Xamarin

The other option is to create a Native Application. Unfortunately, all the OS use a different programming language to develop a application. This realization makes it impossible to create an application that is cross-platform. There would be no reuse of code and the developers would be stuck maintaining 3 separate code bases. This is why a use of a third-party framework can be used to create an application for all three OS but in the same language. The standout option that I selected was Xamarin.

Xamarin is a Framework that allows developers to create a .Net Native Application. The framework will then compile the application to a Native App of all the OS. The advantage of having the native app is that you can utilize a very large potential of the phone's functionality and also, the applications will be displayed using Native UI elements of the OS, thus the applications will feel very natural to the user. By using Xamarin, the UI and the Back-end Logic can be separated. By implementing in this fashion, the only customized code that needs to be developed in the UI for each operating system. The Logic on the backend will can all be reused between all OS. The disadvantage is that you will need to purchase a license to utilize Xamarin.

5.2.3 Choice

After weighing all the options, I decided to go with Xamarin to develop cross-platform Native Applications. The ability to have a native experience for the user is important and the ability to achieve this with a high percentage of code-reuse is undeniably beneficial.



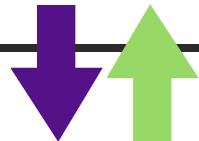
5.3 Mobile Cross-Platform Design Pattern

5.3.1 Model View View-Model (MVVM)

I initially was going to develop the apps following the MVVM pattern. To implement it this way, I found a framework which is very similar to Xamarin called MVVM Cross (MVX) which allows the user to use the MVX libraries and have a core library which could very easily have large code reuse for all mobile operating systems. At the beginning of my phone development process, I looked very strongly into the possibility of using MVX as my framework and it looked like it would be possible. However, tricky aspect of my app would be the need to have a dynamic form displayed for the surveys. I looked at it at all angles and determined that where MVX is currently with their version 3 release, it does not have the correct capabilities yet to successfully implement all aspects of my application, thus the design decision to continue with the Xamarin Framework was made. However, I have immensely high hopes and expectations that MVX will be a great choice for app development in the future.

5.3.2 Portable Class Library (PCL)

A portable class library is essentially a package of classes that will be used by all OS versions of the project and will contain all the unified functionality of the system. Unfortunately, at this moment, PCLs aren't supported by Xamarin directly, (but will be in the near future), so instead, I will be making the PCL package, and just copy the package into each project. Inside of the PCL, will be business logic, data access object for accessing the database (phone, and the remote) and a service access layer (for automation and remote database.) Models will also be defined in the PCL.



5.4 Caching

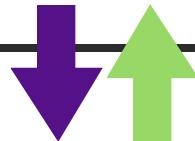
Since there will be a very heavy dependence on the web services in my application, the need to keep the web services performing very quickly and effectively will be vital. So we will help to lessen the load that actually be hitting the web services by implementing caching layers for content that is requested often.

5.4.1 Reverse Proxy

The purpose of a reverse proxy is to essentially have this proxy on our server side and have all the requests feed through it. By doing this, the one client will request from the web services and cache the content. Thus if 5 people request to see a list of companies, the first person will go through the reverse proxy and then hit the web services, the remaining four will call to the reverse proxy, but since the content would be cached after the first time, the web services will not be hit, and the cached content will be returned. Again, this won't be used for Admin Content, or for Updated of information. But mostly will be implemented for the most utilized calls to the system.

5.4.2 Phone Caching

There is no phone caching done with our phone application. The only caching that is done is the default caching of each operating system.

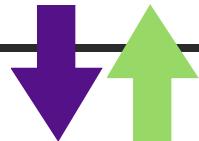


5.5 Database Access

The next decision was how the phone applications will access the database on the server. After researching, it has been advised by many that for security and performance reasons, having the application directly make a connection to a database is not a good idea.

5.1 Data Access Layer through Web Services

We will be implementing a Data Access Layer through which all database calls are made. This way the database side will be separated from the rest of the code bases. To access the data layer, it will be hooked up to web services that can be called and depending on the request parameters, can perform any CRUD functionality. This way the Admin CMS and the phone applications can all get to the database in a way that is secure and reused.



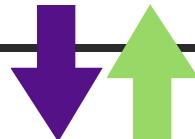
6 Pseudocode for System

Below are template files to be used to view the syntax and design to be followed in files of a similar nature.

6.1 Admin CMS

6.1.1 Element Manager Page (Back-End)

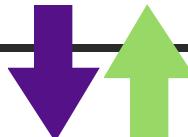
```
1  using ReceiptReward;
2  using System;
3  using System.Collections.Generic;
4  using System.Diagnostics;
5  using System.Web;
6  using System.Web.UI;
7  using System.Web.UI.WebControls;
8
9  public partial class Entity_Manager : System.Web.UI.Page {
10    // Global Variables
11    private int surveyId = 0;
12    private Survey survey = null;
13    private Company company = null;
14    private string type = "survey";
15
16    protected void Page_Load(object sender, EventArgs e) {
17
18      // Call Methods
19      getVariables();
20      if (!this.IsPostBack) {
21        getTitle();
22        getList();
23      }
24    }
25
26
27    /**
28     * Get The Data for the page
29     */
30    private void getVariables() {
31      if (Request.QueryString["surveyId"] != null) { surveyId = int.Parse(Request.QueryString["surveyId"]); }
32      survey = SurveyImpl.getBySurveyId(surveyId);
33      company = CompanyImpl.getById(survey.companyId);
34      lblScripts.Text = "<script type='text/javascript'> var bigId = " + surveyId + "; bigType = '" + type + "';</script>";
35    }
36
37
38    /**
39     * Get the title of the page
40     */
41    private void getTitle() {
42      lblTitle.Text = company.name + ":" + survey.name;
43      txtEditName.Text = survey.name;
44    }
45  }
```



```

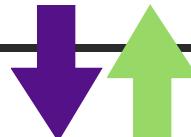
45
46
47 	/***
48 	* Creates a new element
49 	*/
50 	protected void Add_Click(Object sender, EventArgs e) {
51 		Revision revision = new Revision();
52 		revision.revisionId = 0;
53 		revision.active = chkAddActive.Checked;
54 		revision.automation = null;
55 		revision.dateCreated = DateTime.Now;
56 		revision.form = null;
57 		revision.revisionNumber = txtAddRevisionNumber.Text;
58 		revision.surveyId = survey.surveyId;
59 		RevisionImpl.create(revision);
60 	}
61
62
63 	/***
64 	* Edit element
65 	*/
66 	protected void Edit_Click(Object sender, EventArgs e) {
67 		Survey surveyEdit = new Survey();
68 		surveyEdit.surveyId = int.Parse(hdnSaveId.Value);
69 		surveyEdit.name = txtEditName.Text;
70 		SurveyImpl.update(surveyEdit);
71 	}
72
73
74 	/***
75 	* Delete element
76 	*/
77 	protected void Delete_Click(Object sender, EventArgs e) {
78 		if (hdnDeleteType.Value == "survey") {
79 			SurveyImpl.removeBySurveyId(int.Parse(hdnDeleteId.Value));
80 		} else if (hdnDeleteType.Value == "revision") {
81 			RevisionImpl.removeByRevisionId(int.Parse(hdnDeleteId.Value));
82 		}
83 	}
84
85 	public void getList() {
86 		// Instantiate Variables
87 		string html = "";
88 		int counter = 0, childrenCount = 0;
89 		List<Revision> list = RevisionImpl.getBySurveyId(survey.surveyId);
90
91 		// Iterate through list
92 		foreach (Revision item in list) {
93
94 			// Instantiate Variables
95 			childrenCount = 0; //Survey.containsNumOfQuestionGroups(surveyID);
96
97 			html += "" +
98 				"<div style='position: absolute; top: " + (counter * 25) + "px; left: 0px; height: 50px; width: 960px;'>" +
99 					"<div class='TableElementDiv' style='width: 200px;'>" + CommonUtils.trim(item.revisionNumber, 20) + "</div>" +
100 					"<div class='TableElementDiv' style='width: 100px;'>" + item.active + "</div>" +
101 					"<div class='TableElementDiv' style='width: 200px;'>" + item.dateCreated.ToShortDateString() + "</div>" +
102 					"<div class='TableElementDiv' style='width: 200px;'>" +
103 						"<a href='Revision.aspx?revisionId=" + item.revisionId + "'>Edit</a> | " +
104 						"<a class='isALink' onclick='$(\"#dlgDelete\").dialog(\"open\"); $(" + hdnDeleteId + ").val(\"" + item.revisionId + "\")';" +
105 							"$(\" + hdnDeleteType + \").val(\"" + "revision" + "\")';>Delete</a>" +
106 						"</div>" +
107 						"</div>" +
108 						counter++;
109
110 			// Add More to the text
111 			html = "<div id='movingObjectsHolder' style='position: relative; width: 960px; height: " + ((counter - 1) * 25) + "px;'>" + html + "</div>";
112 			html = "" +
113 				"<div class='TableElementDiv' style='width: 200px;'><strong>Revision Number</strong></div>" +
114 				"<div class='TableElementDiv' style='width: 100px;'><strong>Active</strong></div>" +
115 				"<div class='TableElementDiv' style='width: 200px;'><strong>Date Created</strong></div>" +
116 				"<div class='TableElementDiv' style='width: 200px;'><strong>Options</strong></div>" + html;
117
118 			// Display Text
119 			lblList.Text = html;
120
121
122 }

```



6.1.2 Element Implementation

```
1  using System;
2  using System.Collections.Generic;
3  using System.Web;
4  using System.Configuration;
5  using System.Data.SqlClient;
6  using System.Net;
7  using System.IO;
8  using Newtonsoft.Json.Linq;
9  using System.Diagnostics;
10 
11 
12 namespace ReceiptReward
13 {
14 
15     public class SurveyImpl
16     {
17 
18         // Base URL
19         private const string BASE_URL = "http://140.104.69.94:8080/RestfulWebService/bta";
20         private const string DIRECTORY = SurveyImpl.BASE_URL + "/Survey";
21 
22         // GET Methods
23         private const string GET_ALL = SurveyImpl.DIRECTORY + "/All";
24         private const string GET_BY_COMPANY_ID = SurveyImpl.DIRECTORY + "/ById/Company";
25         private const string GET_BY_SURVEY_ID = SurveyImpl.DIRECTORY + "/ById/Survey";
26 
27         // POST Methods
28         private const string CREATE = SurveyImpl.DIRECTORY + "/Create";
29 
30         // UPDATE Methods
31         private const string UPDATE = SurveyImpl.DIRECTORY + "/Update";
32 
33         // DELETE Methods
34         private const string REMOVE_BY_COMPANY_ID = SurveyImpl.DIRECTORY + "/ById/Company";
35         private const string REMOVE_BY_SURVEY_ID = SurveyImpl.DIRECTORY + "/ById/Survey";
36 
37         /**
38          * Obtains a list of all surveys in the database
39          * Calls webservice
40          * @return A list of surveys objects
41          */
42         public static List<Survey> getAll()
43         {
44             // Instantiate Variables
45             string json = "";
46 
47             string json = "";
48             List<Survey> surveys = new List<Survey>();
49 
50             // Set up Request
51             HttpWebRequest httpWebRequest = (HttpWebRequest)WebRequest.Create(SurveyImpl.GET_ALL);
52             httpWebRequest.ContentType = "text/json";
53             httpWebRequest.Method = "GET";
54 
55             // Get the Response
56             HttpWebResponse httpResponse = (HttpWebResponse)httpWebRequest.GetResponse();
57             using (StreamReader streamReader = new StreamReader(httpResponse.GetResponseStream()))
58             {
59                 json = ("{" + "data": " " + streamReader.ReadToEnd() + "}");
60 
61                 // Convert JSON to object
62                 JObject jobject = JObject.Parse(json);
63                 JToken jCodes = jobject["data"];
64                 foreach (JToken jCode in jCodes.Children())
65                 {
66                     Survey survey = new Survey();
67                     survey.surveyId = (int)jCode["surveyId"];
68                     survey.companyId = (int)jCode["companyId"];
69                     survey.name = (string)jCode["name"];
70                     surveys.Add(survey);
71                 }
72 
73                 // Return
74                 return surveys;
75             }
76 
77             /**
78              * Obtains a Survey by Company Id from the database
79              * Calls webservice
80              * @return A Survey object
81              */
82             public static List<Survey> getByCompanyId(int companyId)
83             {
84                 // Instantiate Variables
85                 string json = "";
86                 List<Survey> surveys = new List<Survey>();
87 
88                 // Set up Request
89                 HttpWebRequest httpWebRequest = (HttpWebRequest)WebRequest.Create(SurveyImpl.GET_BY_COMPANY_ID + "/" + companyId);
90 
91             }
92         }
93     }
94 }
```



```

89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177

    HttpWebRequest httpWebRequest = (HttpWebRequest)WebRequest.Create(SurveyImpl.GET_BY_COMPANY_ID + "/" + companyId);
    httpWebRequest.ContentType = "text/json";
    httpWebRequest.Method = "GET";

    // Get the Response
    HttpWebResponse httpResponse = (HttpWebResponse)httpWebRequest.GetResponse();
    using (StreamReader streamReader = new StreamReader(httpResponse.GetResponseStream()))
    {
        json = ("(\"data\":" + streamReader.ReadToEnd() + ")");
    }

    // Convert JSON to object
    JObject jobject = JObject.Parse(json);
    JToken jCodes = jobject["data"];
    foreach (JToken jCode in jCodes.Children())
    {
        Survey survey = new Survey();
        survey.surveyId = (int)jCode["surveyId"];
        survey.companyId = (int)jCode["companyId"];
        survey.name = (string)jCode["name"];
        surveys.Add(survey);
    }

    // Return
    return surveys;
}

<**
 * Obtains a Survey by Id from the database
 * Calls webservice
 * @return A Survey object
 */
public static Survey getSurveyId(int surveyId)
{
    // Instantiate Variables
    string json = "";
    Survey survey = new Survey();

    // Set up Request
    HttpWebRequest httpWebRequest = (HttpWebRequest)WebRequest.Create(SurveyImpl.GET_BY_SURVEY_ID + "/" + surveyId);
    httpWebRequest.ContentType = "text/json";
    httpWebRequest.Method = "GET";

    // Get the Response
    HttpWebResponse httpResponse = (HttpWebResponse)httpWebRequest.GetResponse();
    using (StreamReader streamReader = new StreamReader(httpResponse.GetResponseStream()))
    {
        json = ("(\"data\":" + streamReader.ReadToEnd() + ")");
    }

    // Convert JSON to object
    JObject jobject = JObject.Parse(json);
    JToken jCodes = jobject["data"];
    foreach (JToken jCode in jCodes.Children())
    {
        survey.surveyId = (int)jCode["surveyId"];
        survey.companyId = (int)jCode["companyId"];
        survey.name = (string)jCode["name"];
    }

    // Return
    return survey;
}

<**
 * Create a new entry in the Database
 * Calls webservice
 */
public static void create(Survey survey)
{
    // Set up Request
    HttpWebRequest httpWebRequest = (HttpWebRequest)WebRequest.Create(SurveyImpl.CREATE);
    httpWebRequest.ContentType = "application/json";
    httpWebRequest.Method = "POST";

    // Send Request Data
    using (StreamWriter streamWriter = new StreamWriter(httpWebRequest.GetRequestStream()))
    {
        string json = ("(\"surveyId\":\"" + survey.surveyId + "\",\"companyId\":\"" + survey.companyId + "\",\"name\":\"" + survey.name + "\")");
        Debug.WriteLine(json);
        streamWriter.Write(json);
        streamWriter.Flush();
        streamWriter.Close();
    }

    // Get the Response
    HttpWebResponse httpResponse = (HttpWebResponse)httpWebRequest.GetResponse();
}

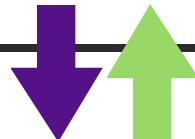
```



```

177     HttpResponse httpResponse = (HttpResponse)httpWebRequest.GetResponse();
178 }
179
180 /**
181 * Update entry in the Database
182 * Calls webservice
183 */
184
185 public static void update(Survey survey)
186 {
187     // Set up Request
188     HttpWebRequest httpWebRequest = (HttpWebRequest)WebRequest.Create(SurveyImpl.UPDATE);
189     httpWebRequest.ContentType = "application/json";
190     httpWebRequest.Method = "PUT";
191
192     // Send Request Data
193     using (StreamWriter streamWriter = new StreamWriter(httpWebRequest.GetRequestStream()))
194     {
195         string json = ("(\"surveyId\":\"" + survey.surveyId + "\",\"name\":\"" + survey.name + "\")");
196         Debug.WriteLine(json);
197         streamWriter.Write(json);
198         streamWriter.Flush();
199         streamWriter.Close();
200     }
201
202     // Get the Response
203     HttpResponse httpResponse = (HttpResponse)httpWebRequest.GetResponse();
204 }
205
206 /**
207 * Removes an entry in the Database by Company Id
208 * Calls webservice
209 */
210
211 public static void removeByCompanyId(int companyId)
212 {
213     // Set up Request
214     HttpWebRequest httpWebRequest = (HttpWebRequest)WebRequest.Create(SurveyImpl.REMOVE_BY_COMPANY_ID + "/" + companyId);
215     httpWebRequest.ContentType = "application/json";
216     httpWebRequest.Method = "DELETE";
217
218     // Get the Response
219     HttpResponse httpResponse = (HttpResponse)httpWebRequest.GetResponse();
220 }
221
222 /**
223 * Removes an entry in the Database by Survey Id
224 * Calls webservice
225 */
226
227 public static void removeBySurveyId(int surveyId)
228 {
229     // Set up Request
230     HttpWebRequest httpWebRequest = (HttpWebRequest)WebRequest.Create(SurveyImpl.REMOVE_BY_SURVEY_ID + "/" + surveyId);
231     httpWebRequest.ContentType = "application/json";
232     httpWebRequest.Method = "DELETE";
233
234     // Get the Response
235     HttpResponse httpResponse = (HttpResponse)httpWebRequest.GetResponse();
236 }
237
238
239 /**
240 * Get the breadcrumbs
241 */
242 static public string getBreadcrumbs(int id)
243 {
244     // Instantiate Variables
245     string tempStr = "";
246     Survey survey = SurveyImpl.getBySurveyId(id);
247     Company company = CompanyImpl.getById(survey.companyId);
248
249     // Survey
250     tempStr = "<a class='breadcrumbCurrent' href='Survey.aspx?surveyId=" + survey.surveyId + "'>" + survey.name + "</a>" + tempStr;
251
252     // Company
253     tempStr = "<a class='breadcrumbNotCurrent' href='Company.aspx?companyId=" + company.companyId + "'>" + company.name + "</a> >> " + tempStr;
254
255     // Homepage
256     tempStr = "<a class='breadcrumbNotCurrent' href='Index.aspx'>Home</a> >> " + tempStr;
257
258     // Output Links
259     return tempStr;
260 }
261
262 }
263 }

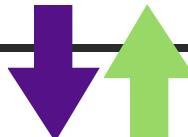
```



6.2 Web Services

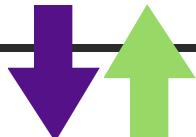
6.2.1 Service

```
1 package com.bti.ws.controller;
2
3 import com.bti.ws.implementations.SurveyImpl;
4 import com.bti.ws.model.Survey;
5 import java.util.List;
6 import org.springframework.http.HttpStatus;
7 import org.springframework.stereotype.Controller;
8 import org.springframework.web.bind.annotation.ExceptionHandler;
9 import org.springframework.web.bind.annotation.PathVariable;
10 import org.springframework.web.bind.annotation.RequestBody;
11 import org.springframework.web.bind.annotation.RequestMapping;
12 import org.springframework.web.bind.annotation.RequestMethod;
13 import org.springframework.web.bind.annotation.ResponseBody;
14 import org.springframework.web.bind.annotation.ResponseStatus;
15
16 @Controller
17 @RequestMapping("/Survey")
18 public class SurveyController {
19
20
21     @RequestMapping(value = "/All", method = RequestMethod.GET)
22     @ResponseBody
23     List<Survey> getAll() {
24         return SurveyImpl.findAll();
25     }
26
27     @RequestMapping(value = "/ById/Company/{companyId}", method = RequestMethod.GET)
28     @ResponseBody
29     List<Survey> getByCompanyId(@PathVariable("companyId") int companyId) {
30         return SurveyImpl.findById(companyId);
31     }
32
33     @RequestMapping(value = "/ById/Survey/{surveyId}", method = RequestMethod.GET)
34     @ResponseBody
35     List<Survey> getBySurveyId(@PathVariable("surveyId") int surveyId) {
36         return SurveyImpl.findBySurveyId(surveyId);
37     }
38
39     @RequestMapping(value = "/Create", method = RequestMethod.POST)
40     @ResponseStatus(HttpStatus.OK)
41     void create(@RequestBody Survey survey) {
42         SurveyImpl.create(survey);
43     }
44
45     @RequestMapping(value = "/Update", method = RequestMethod.PUT)
46     @ResponseStatus(HttpStatus.OK)
47     void update(@RequestBody Survey survey) {
48         SurveyImpl.update(survey);
49     }
50
51     @RequestMapping(value = "/ById/Survey/{surveyId}", method = RequestMethod.DELETE)
52     @ResponseStatus(HttpStatus.OK)
53     void removeBySurveyId(@PathVariable("surveyId") int surveyId) {
54         SurveyImpl.deleteBySurveyId(surveyId);
55     }
56
57     @RequestMapping(value = "/ById/Company/{companyId}", method = RequestMethod.DELETE)
58     @ResponseStatus(HttpStatus.OK)
59     void removeByCompanyId(@PathVariable("companyId") int companyId) {
60         SurveyImpl.deleteByCompanyId(companyId);
61     }
62
63     @ExceptionHandler(NotFoundException.class)
64     @ResponseStatus(value = HttpStatus.NOT_FOUND, reason="Code Type not found")
65     void handleNotFound(NotFoundException exc) {}
66
67     class NotFoundException extends RuntimeException {
68         private static final long serialVersionUID = 1L;
69     }
70 }
71
```



6.2.2 Implementation

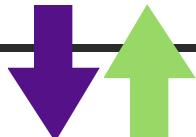
```
1  package com.bti.ws.implementations;
2
3  import com.bti.ws.common.DBConnection;
4  import com.bti.ws.model.Survey;
5  import java.sql.Connection;
6  import java.sql.PreparedStatement;
7  import java.sql.ResultSet;
8  import java.util.ArrayList;
9  import java.util.List;
10
11 public class SurveyImpl {
12
13     private static final String FIND_ALL = "SELECT * FROM survey;";
14     private static final String FIND_BY_SURVEY_ID = "SELECT * FROM survey WHERE survey_id = ?;";
15     private static final String FIND_BY_COMPANY_ID = "SELECT * FROM survey WHERE company_id = ?;";
16     private static final String FIND_NEXT_ID = "SELECT TOP 1 survey_id FROM survey ORDER BY survey_id DESC;";
17     private static final String CREATE = "INSERT INTO survey (survey_id, company_id, name) VALUES (?, ?, ?);";
18     private static final String UPDATE = "UPDATE survey SET name = ? WHERE survey_id = ?;";
19     private static final String DELETE_BY_SURVEY_ID = "DELETE FROM survey WHERE survey_id = ?;";
20     private static final String DELETE_BY_COMPANY_ID = "DELETE FROM survey WHERE company_id = ?;";
21
22
23     **** Find Methods ****
24     public static List<Survey> findAll() {
25         // Instantiate Variables
26         Connection con = null;
27         PreparedStatement ps = null;
28         ResultSet rs = null;
29         List<Survey> surveys = new ArrayList<Survey>();
30
31         try { // Get List of All
32             con = DBConnection.getConnection();
33             ps = con.prepareStatement(SurveyImpl.FIND_ALL);
34             rs = ps.executeQuery();
35             while(rs.next()) {
36                 surveys.add(new Survey(rs.getInt("survey_id"), rs.getInt("company_id"), rs.getString("name")));
37             }
38             rs.close();
39             ps.close();
40         } catch(Exception ex) {
41             System.out.println("Error: " + ex.getMessage());
42             // TODO: Handle Exception
43         } finally {
44             try {
45                 if(con != null) {
46                     con.close();
47                     con = null;
48                 }
49             } catch(Exception ex){}
50         }
51
52         // Return
53         return surveys;
54     }
55
56     public static List<Survey> findByCompanyId(int companyId) {
57         // Instantiate Variables
58         Connection con = null;
59         PreparedStatement ps = null;
60         ResultSet rs = null;
61         List<Survey> surveys = new ArrayList<Survey>();
```



```

63   try { // Get List By Id
64     con = DBConnection.getConnection();
65     ps = con.prepareStatement(SurveyImpl.FIND_BY_COMPANY_ID);
66     ps.setInt(1, companyId);
67     rs = ps.executeQuery();
68     while(rs.next()) {
69       surveys.add(new Survey(rs.getInt("survey_id"), rs.getInt("company_id"), rs.getString("name")));
70     }
71     rs.close();
72     ps.close();
73   } catch(Exception ex) {
74     System.out.println("Error: " + ex.getMessage());
75     // @TODO: Handle Exception
76   } finally {
77     try {
78       if(con != null) {
79         con.close();
80         con = null;
81       }
82     } catch(Exception ex){}
83   }
84
85   // Return
86   return surveys;
87 }
88
89 public static List<Survey> findBySurveyId(int surveyId) {
90   // Instantiate Variables
91   Connection con = null;
92   PreparedStatement ps = null;
93   ResultSet rs = null;
94   List<Survey> surveys = new ArrayList<Survey>();
95
96   try { // Get List By Id
97     con = DBConnection.getConnection();
98     ps = con.prepareStatement(SurveyImpl.FIND_BY_SURVEY_ID);
99     ps.setInt(1, surveyId);
100    rs = ps.executeQuery();
101   while(rs.next()) {
102     surveys.add(new Survey(rs.getInt("survey_id"), rs.getInt("company_id"), rs.getString("name")));
103   }
104   rs.close();
105   ps.close();
106 } catch(Exception ex) {
107   System.out.println("Error: " + ex.getMessage());
108   // @TODO: Handle Exception
109 } finally {
110   try {
111     if(con != null) {
112       con.close();
113       con = null;
114     }
115   } catch(Exception ex){}
116 }
117
118 // Return
119 return surveys;
120 }
121

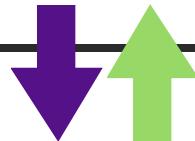
```



```

121
122     public static int findNextId() {
123         // Instantiate Variables
124         Connection con = null;
125         PreparedStatement ps = null;
126         ResultSet rs = null;
127         int surveyId = 1;
128
129         try { // Get All the Code Types
130             con = DBConnection.getConnection();
131             ps = con.prepareStatement(SurveyImpl.FIND_NEXT_ID);
132             rs = ps.executeQuery();
133             while(rs.next()) {
134                 surveyId = rs.getInt("survey_id") + 1;
135             }
136             rs.close();
137             ps.close();
138         } catch(Exception ex) {
139             System.out.println("Error");
140             // @TODO: Handle Exception
141         } finally {
142             try {
143                 if(con != null) {
144                     con.close();
145                     con = null;
146                 }
147             } catch(Exception ex){}
148         }
149
150         // Return
151         return surveyId;
152     }
153
154
155     /**** Create Methods *****/
156     public static void create(Survey survey) {
157         // Instantiate Variables
158         Connection con = null;
159         PreparedStatement ps = null;
160
161         try { // Set ID
162             survey.setSurveyId(SurveyImpl.findNextId());
163
164             // Create New Entry
165             con = DBConnection.getConnection();
166             ps = con.prepareStatement(SurveyImpl.CREATE);
167             ps.setInt(1, survey.getSurveyId());
168             ps.setInt(2, survey.getCompanyId());
169             ps.setString(3, survey.getName());
170             ps.executeUpdate();
171             ps.close();
172         } catch(Exception ex) {
173             System.out.println("Error: " + ex.getMessage());
174             // @TODO: Handle Exception
175         } finally {
176             try {
177                 if(con != null) {
178                     con.close();
179                     con = null;
180                 }
181             } catch(Exception ex){}

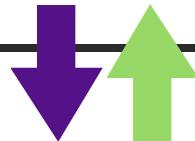
```

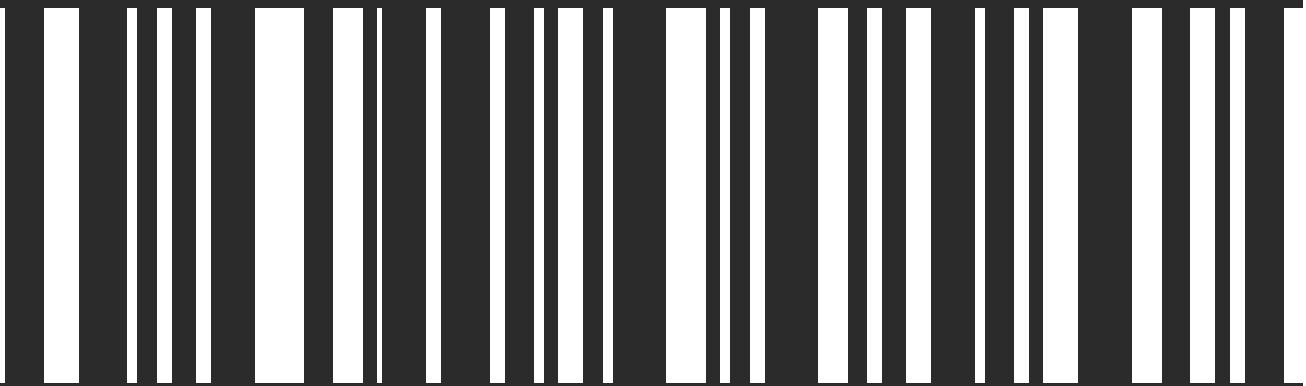


```

181         } catch(Exception ex) {}
182     }
183 }
184
185 **** Update Methods *****
186 public static void update(Survey survey) {
187     // Instantiate Variables
188     Connection con = null;
189     PreparedStatement ps = null;
190
191     try { // Update Existing Entry
192         con = DBConnection.getConnection();
193         ps = con.prepareStatement(SurveyImpl.UPDATE);
194         ps.setString(1, survey.getName());
195         ps.setInt(2, survey.getSurveyId());
196         ps.executeUpdate();
197         ps.close();
198     } catch(Exception ex) {
199         System.out.println("Error: " + ex.getMessage());
200         // @TODO: Handle Exception
201     } finally {
202         try {
203             if(con != null) {
204                 con.close();
205                 con = null;
206             }
207         } catch(Exception ex){}
208     }
209 }
210
211 **** Delete Methods *****
212 public static void deleteBySurveyId(int surveyId) {
213     // Instantiate Variables
214     Connection con = null;
215     PreparedStatement ps = null;
216
217     try { // Delete Existing Entry
218         con = DBConnection.getConnection();
219         ps = con.prepareStatement(SurveyImpl.DELETE_BY_SURVEY_ID);
220         ps.setInt(1, surveyId);
221         ps.executeUpdate();
222         ps.close();
223     } catch(Exception ex) {
224         System.out.println("Error: " + ex.getMessage());
225         // @TODO: Handle Exception
226     } finally {
227         try {
228             if(con != null) {
229                 con.close();
230                 con = null;
231             }
232         } catch(Exception ex){}
233     }
234 }
235
236
237 public static void deleteByCompanyId(int companyId) {
238     // Instantiate Variables
239     Connection con = null;
240     PreparedStatement ps = null;
241
242     try { // Delete Existing Entry
243         con = DBConnection.getConnection();
244         ps = con.prepareStatement(SurveyImpl.DELETE_BY_COMPANY_ID);
245         ps.setInt(1, companyId);
246         ps.executeUpdate();
247         ps.close();
248     } catch(Exception ex) {
249         System.out.println("Error: " + ex.getMessage());
250         // @TODO: Handle Exception
251     } finally {
252         try {
253             if(con != null) {
254                 con.close();
255                 con = null;
256             }
257         } catch(Exception ex){}
258     }
259 }
260 }
261

```





DATABASE DESIGN DOCUMENT

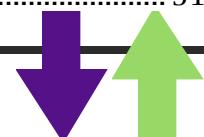
8/23/2013

Graduate Capstone

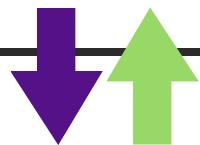


Table of Contents

1. Introduction.....	4
1.1 Intended Audience.....	4
1.2 References.....	4
1.3 Revision History	4
2.1 Appearance / Naming Convention.....	5
2.1.1 Table Naming Convention.....	5
2.1.2 Column Naming Convention	5
2.2 DBMS Platform	5
2.3 Operations	5
2.4 Maintenance.....	5
3.1 Entity Relationship Diagram.....	6
3.1.1 Entire System	6
3.1.2 Focus of Company and Survey.....	7
3.1.3 Focus on Revision	7
3.1.4 Focus on Question	8
3.1.5 Automation.....	9
3.1.6 Form	9
3.1.7 Focus on Submission and Reward.....	10
3.1.8 Focus on User and Language.....	11
3.2 Table Descriptions	12
3.2.1 Company Table.....	12
3.2.2 Survey Table	13
3.2.3 Revision Table.....	14
3.2.4 Automation Table	15
3.2.5 Command Elements Table	16
3.2.6 Command Element Table	17
3.2.7 Command Identifier Table	18
3.2.8 Form Table.....	19
3.2.9 Form Elements Table.....	20
3.2.10 Form Element Table	21
3.2.11 Form Element Attribute Table	22
3.2.12 Form Element Option Table	23
3.2.13 Display Text Table.....	24
3.2.14 Display Text Translation Table	25
3.2.15 Code Type.....	26
3.2.16 Code	27
3.2.17 User Table.....	28
3.2.18 User Data Table	29
3.2.19 User Setting Table.....	30
3.2.20 Submission Table.....	31



3.2.21 Submission Answer Table.....	32
3.2.22 Submission Reward Table.....	33
3.2.23 Question Table	34
3.2.24 Reward Table	35
3.2.25 Command Form Element Relation Table.....	36
3.2.26 Form Flow Table.....	37
3.2.27 Automation Flow Table	38
Appendix.....	39
Appendix A: Code Types	39



1. Introduction

The purpose of this document is to define the tables, fields and attributes of the database.

1.1 Intended Audience

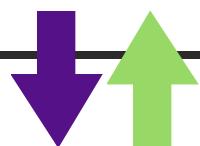
This document is intended for individuals of a high technical background.

1.2 References

- <http://wiki.healthmetricsnetwork.info/wiki-kigali/lib/exe/fetch.php?media=templates:databasedesigndocumenttemplate.dot>
- <http://www.mbrs.doe.gov.bz/dbdocs/tech/Design.pdf>
- <https://apps.bsu.edu/AdminConsole/Documentation/SQL/Design/Database.aspx>

1.3 Revision History

Name	Date	Reason For Change	Version
Andy Bottom	03/18/2013	Started Creating the Document; Created documentation about 75% of the tables;	0.1
Andy Bottom	04/27/2013	Created a more formal document; Implemented most of the remaining tables;	0.2
Andy Bottom	05/07/2013	Finalized the layout and added several attributes	1.0
Andy Bottom	08/18/2013	Updated all tables and diagrams to be current with what is currently modeled in the system.	2.0



2 Database-Wide Design Decisions

2.1 Appearance / Naming Convention

2.1.1 Table Naming Convention

All lowercase and words are separated by underscores (_). Ex. table_foobar

2.1.2 Column Naming Convention

All lowercase and words are separated by underscores (_). Ex. foobar_id

2.2 DBMS Platform

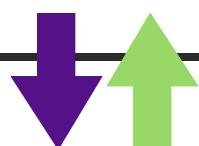
The database is hosted on a Carroll University Network Server. The platform of the database is Microsoft SQL Server. To manage the server, Microsoft SQL Server Management Studio is the software used.

2.3 Operations

Currently there isn't any system set up currently for backing up the server. As it becomes time to officially release the product, it will be looked into as possible solutions to back-up the system.

2.4 Maintenance

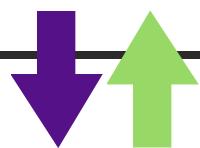
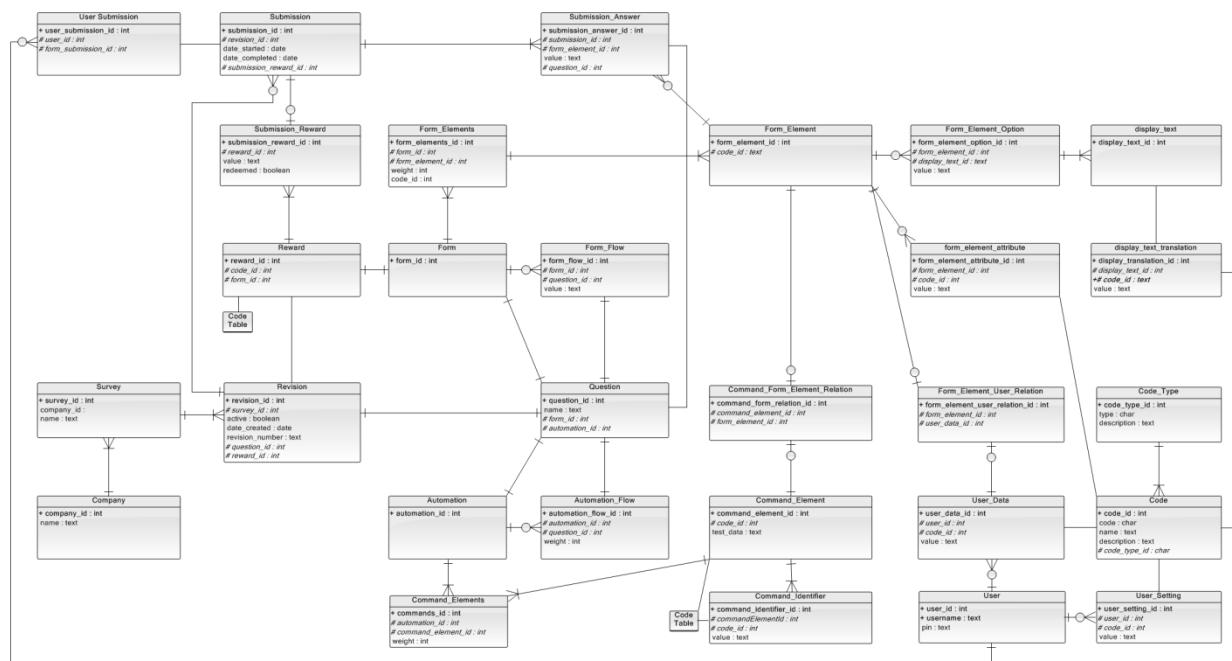
The data in the database will be managed mainly through the Administrator Back-Office Application. Also, monthly, any security updates for the server will be needed to go through and update.



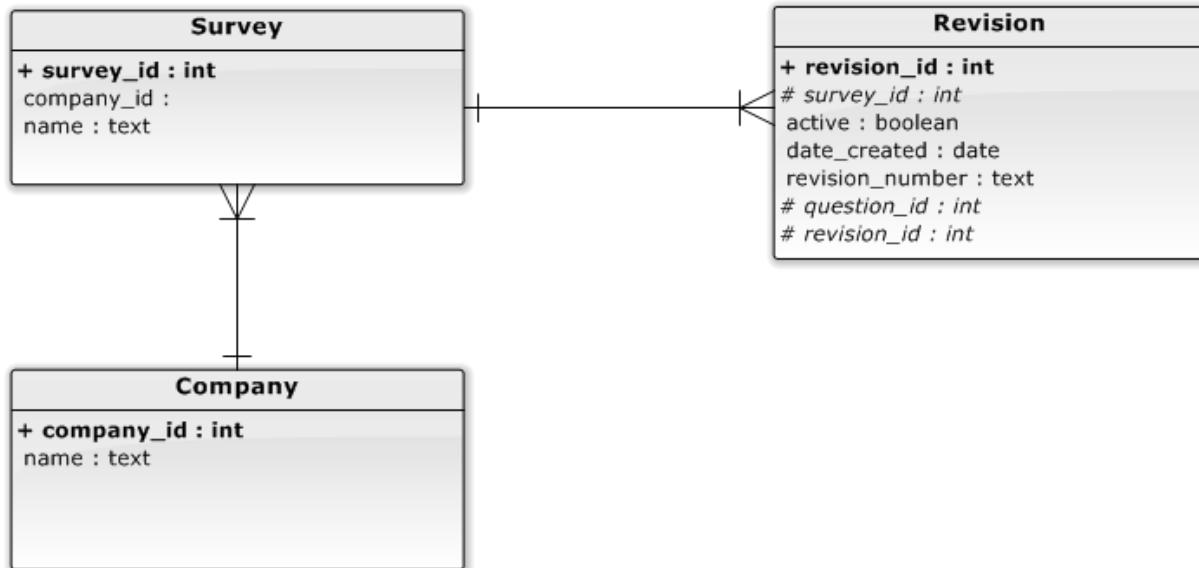
3 Detailed Database Design

3.1 Entity Relationship Diagram

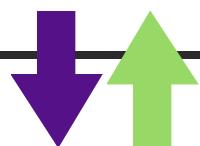
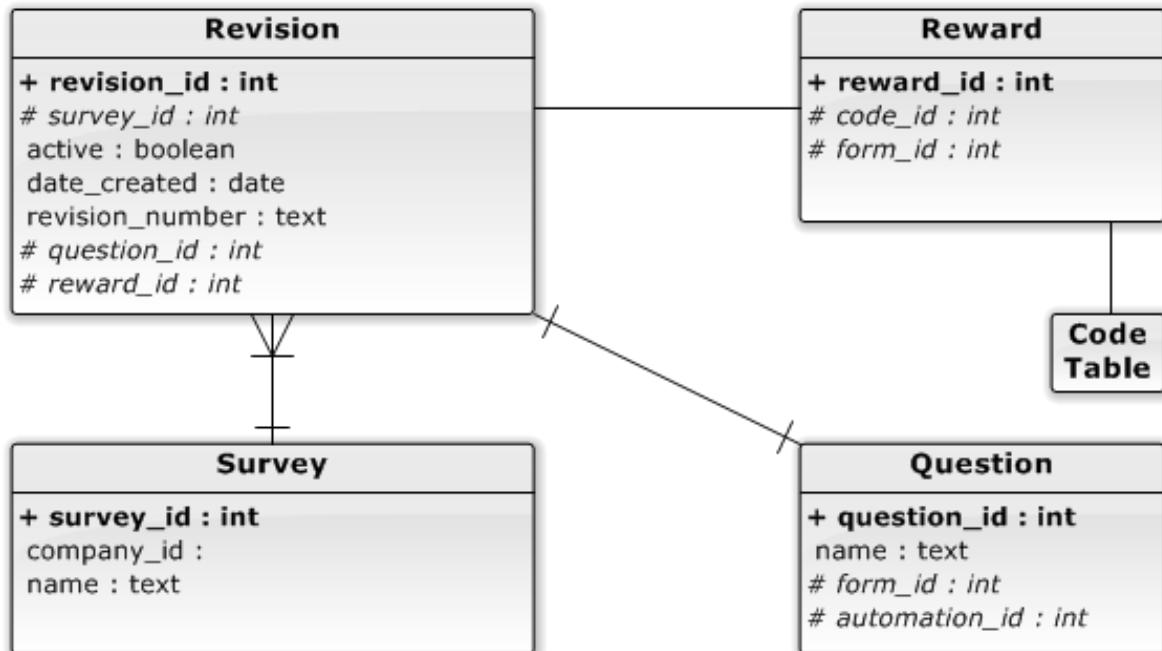
3.1.1 Entire System



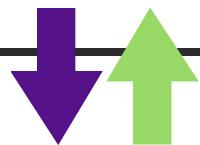
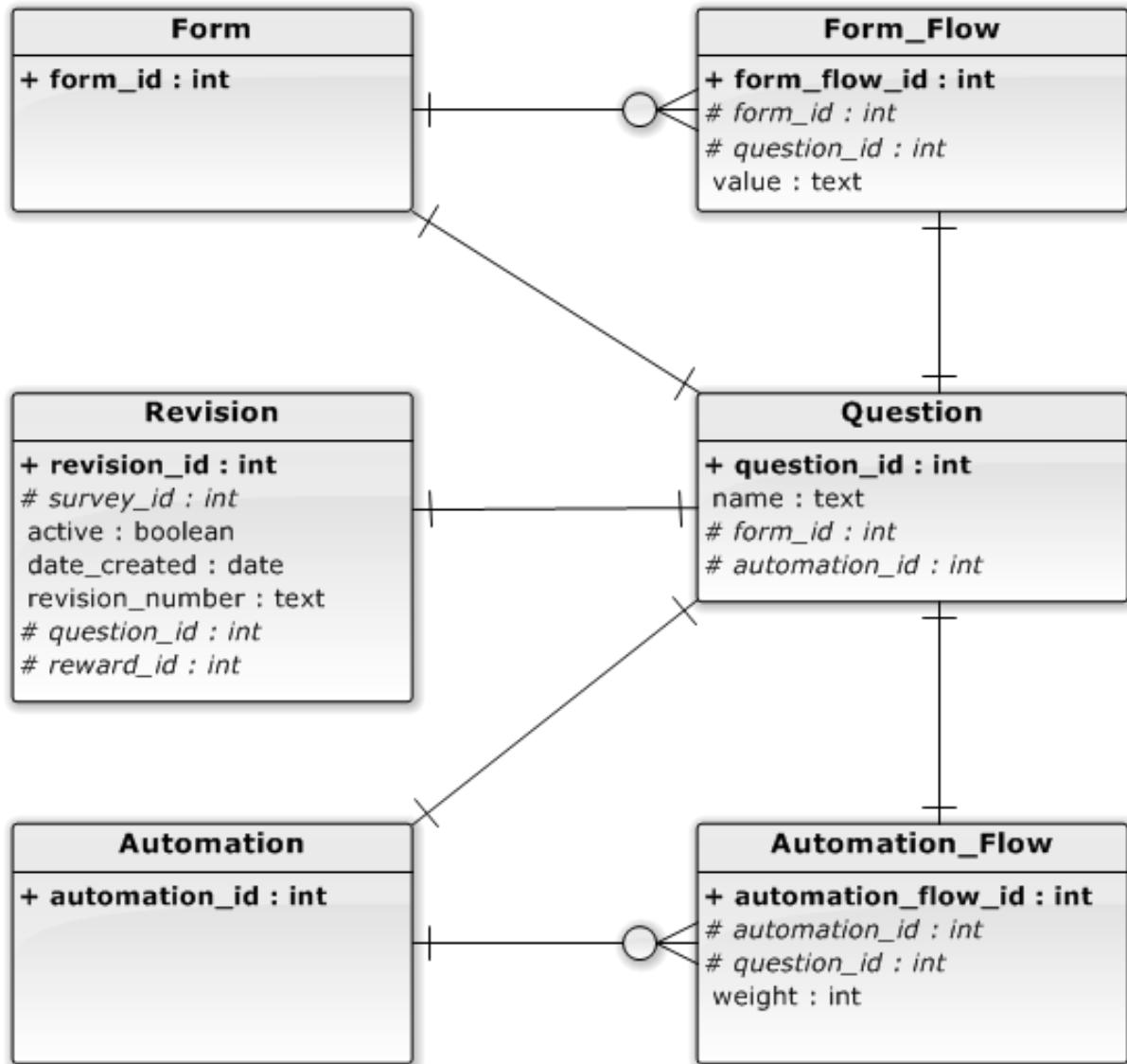
3.1.2 Focus of Company and Survey



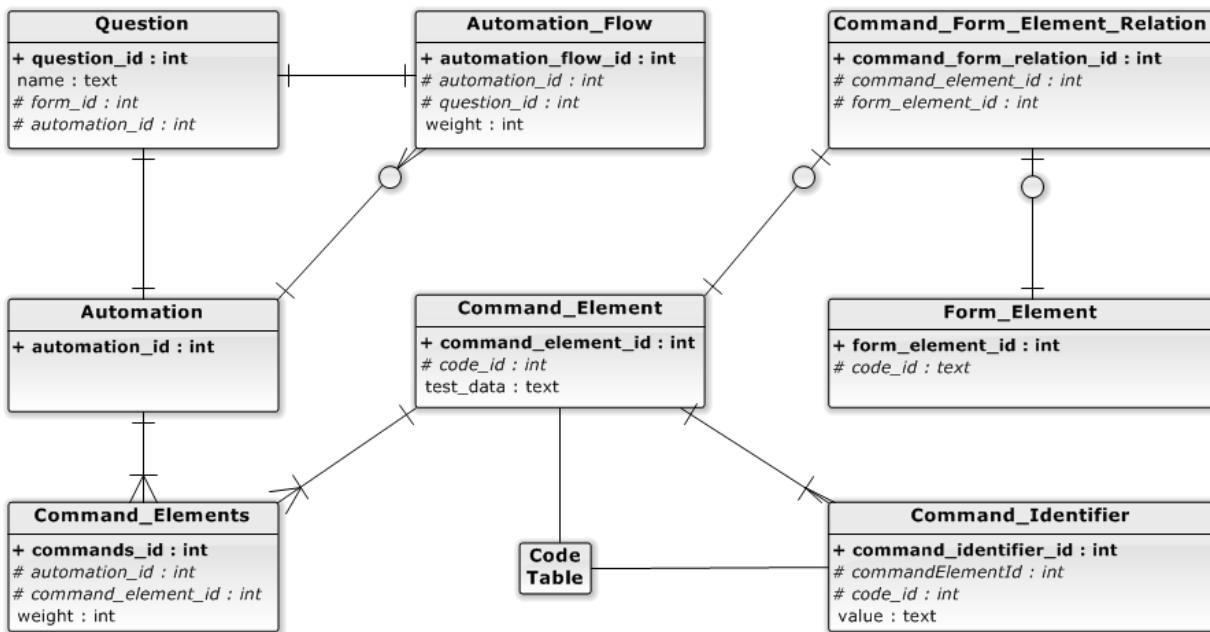
3.1.3 Focus on Revision



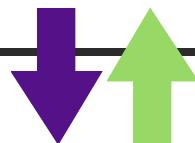
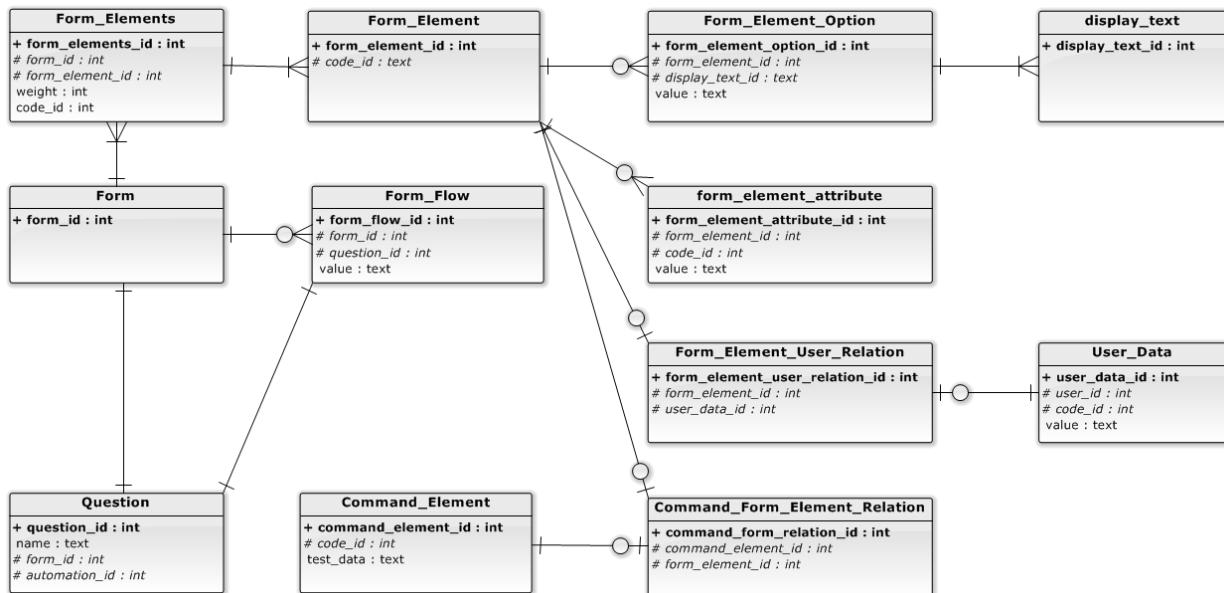
3.1.4 Focus on Question



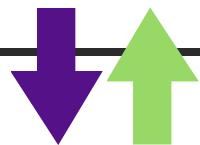
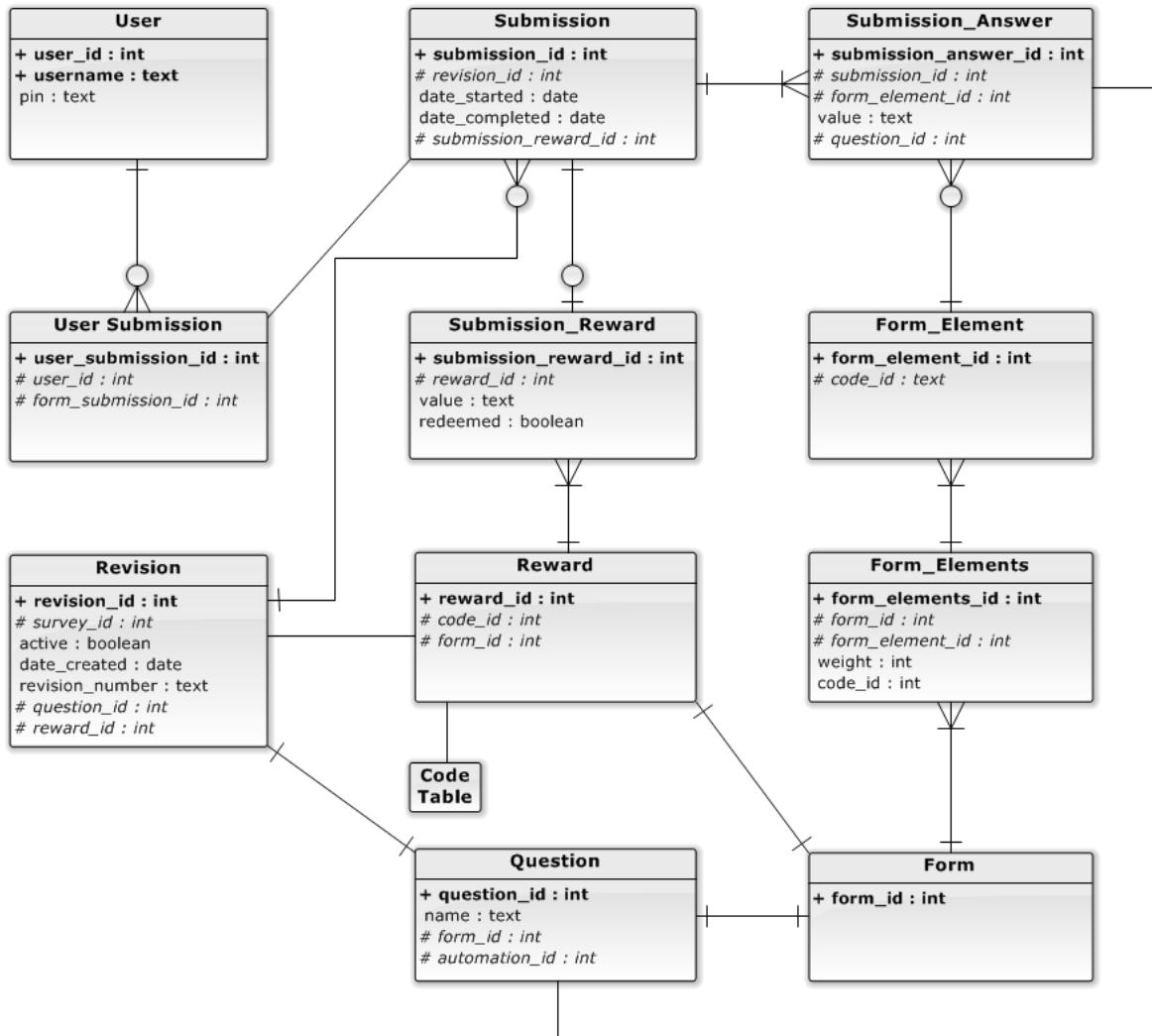
3.1.5 Automation



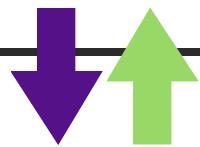
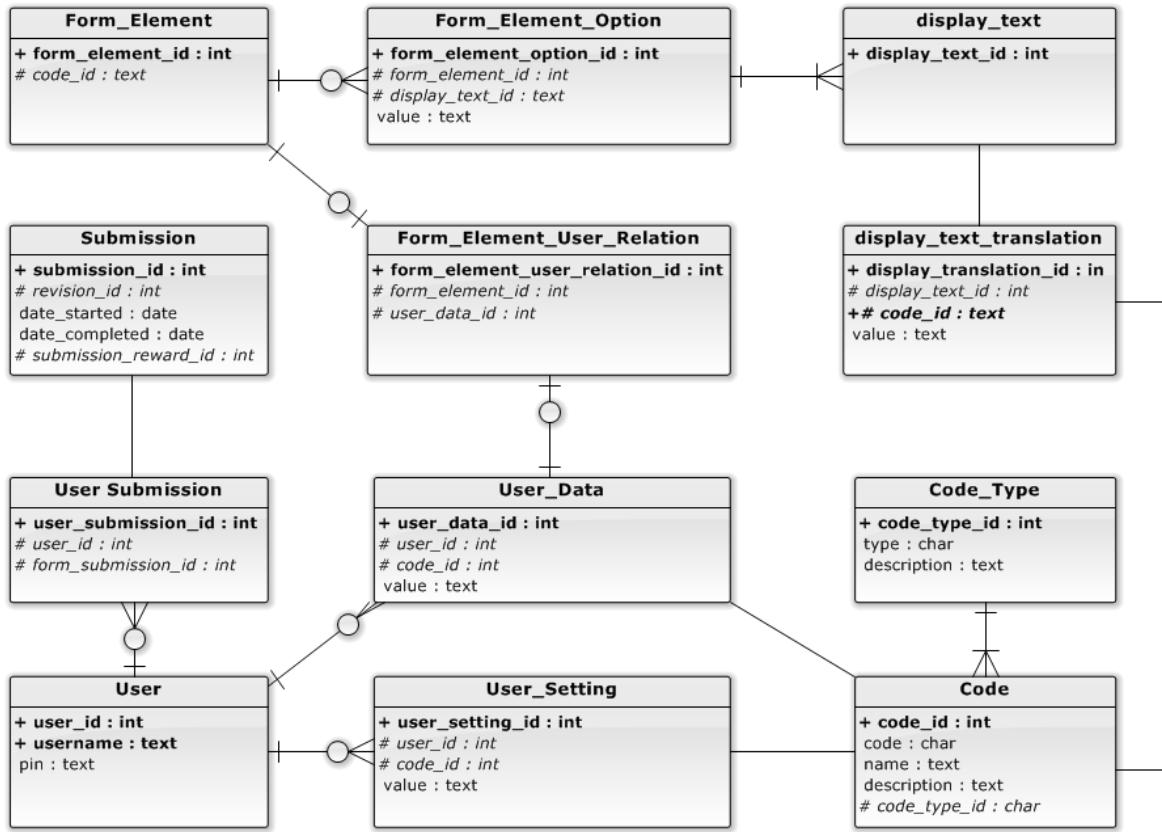
3.1.6 Form



3.1.7 Focus on Submission and Reward



3.1.8 Focus on User and Language



3.2 Table Descriptions

3.2.1 Company Table

3.2.1.1 Description

The company table is the top table in the theoretical hierarchy in the database. The company table takes the object oriented approach as representing the companies in the project.

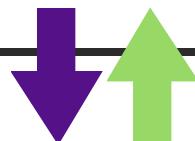
There aren't too many attributes in the company table right now; however I will eventually be referencing a company_attribute table to dynamically add attributes to the table without creating predefined attributes directly into the company table. Further down the implementation, attributes may be transitioned from the attribute_table to the company table when we find out which attributes will be necessary.

3.2.1.2 Table Info

Column Name	Type	Descriptive Name	Column Properties	Allow Nulls	Description
company_id	int	Company ID	Unique; Index; Primary_Key;	Not Nullable	The company id is the unique id to identify the individual company
name	text	Name		Not Nullable	The name of the company

3.2.1.3 Relationships

- A company can have multiple surveys
- A company may not necessarily always have a survey (although would be a very uncommon occurrence)



3.2.2 Survey Table

3.2.2.1 Description

The survey table is the next level underneath companies.

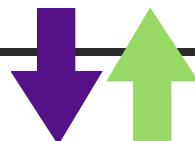
There aren't too many attributes in the survey table right now; however I will eventually be referencing a survey_attribute table to dynamically add attributes to the table without creating predefined attributes directly into the survey table. Further down the implementation, attributes may be transitioned from the attribute_table to the survey table when we find out which attributes will be necessary.

3.2.2.2 Table Info

Column Name	Type	Descriptive Name	Column Properties	Allow Nulls	Description
survey_id	int	Survey ID	Unique; Index; Primary_Key;	Not Nullable	The Survey ID is the unique id to identify the individual survey
company_id	int	Company ID	Index; Foreign_Key;	Not Nullable	The Company ID that the survey is attached to.
name	text	Name		Not Nullable	The name of the survey

3.2.2.3 Relationships

- A survey will always have one company that it is attached to.
- A survey can have multiple revisions
- A survey will always have at least one revision.
- A survey can only have one revision that is active. The others must be set to inactive.



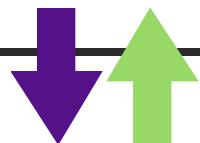
3.2.3 Revision Table

3.2.3.1 Description

The revision table contains the objects located underneath surveys. The revision object allows separation from the surveys and the forms and commands. The reason that these are separated from the survey table is to be able to maintain data integrity by keeping the history of the surveys intact, also keeping referential integrity. It is anticipated that survey questions may change. Thus revisions separate the references by adding a middle table away from the survey. The revision table references a question id which is the question that starts the flows of both the form and the automation. The reward is a reference to the reward table of what kind of reward is that table. It is not in the survey table because rewards may change, thus the separation.

3.2.3.2 Table Info

Column Name	Type	Descriptive Name	Column Properties	Allow Nulls	Description
revision_id	int	Revision ID	Unique Index; Primary Key	Not Null	The Revision ID is used to identify an individual revision
survey_id	int	Survey ID	Index; Foreign_Key	Not Null	Reference to the Survey ID that the revision is attached.
active	boolean	Active		Not Null	A boolean to determine whether the current revision is the current one.
date_created	DateTime	Date Created		Not Null	Stores the date when the revision was created
date_update	DateTime	Date Update		Not Null	Stores when the revision was last modified.
revision_number	text	Revision Number		Not Null	The human identifiable label to describe the revision.
question_id	int	Question ID	Foreign_Key	Not Null	The reference to the question_id.
reward_id	int	Reward ID	Foreign_Key	Not Null	The reference to the reward_id.



3.2.3.3 Relationships

- A revision is attached to only one survey.
- A revision has only one reference to a question entry
- A revision has only one reward table reference.

3.2.4 Automation Table

3.2.4.1 Description

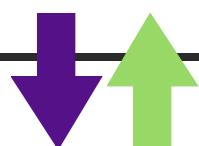
The automation id is used to provide separation from the revision and the command elements.

3.2.4.2 Table Info

Column Name	Type	Descriptive Name	Column Properties	Allow Nulls	Description
automation_id	int	Automation ID	Unique; Index; Primary_Key;	Not Null	The Automation ID is the unique id to identify the individual survey

3.2.4.3 Relationships

- Automation is part of only one question.
- Automation can have none to many automation_flows associated to it.
- Automation can have none or many command_elements references associated to it.



3.2.5 Command Elements Table

3.2.5.1 Description

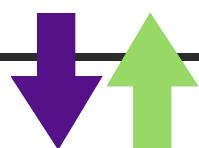
The command_elements table is the relational table that matches a command_element to an automation_id.

3.2.5.2 Table Info

Column Name	Type	Descriptive Name	Column Properties	Allow Nulls	Description
commands_id	int	Commands ID	Unique; Index; Primary_Key;	Not Null	The Commands ID is used to identify an individual reference
automation_id	int	Automation ID	Index; Foreign_Key;	Not Null	Reference to the Automation ID
command_element_id	int	Command Element ID	Foreign_Key;	Not Null	The reference to the command_element
weight	int	Weight		Not Null	Weight is the specific priority order of the element in the list

3.2.5.3 Relationships

- A command_elements entry will always have only one automation_id relation
- A command_elements entry will only have one command_element relation



3.2.6 Command Element Table

3.2.6.1 Description

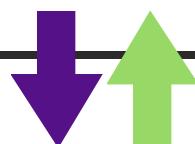
The command element table represents each individual commands that is listed in the automation aspect of a revision. The command_element represents all the different type of command_element, but is identified by the code_id.

3.2.6.2 Table Info

Column Name	Type	Descriptive Name	Column Properties	Allow Nulls	Description
command_element_id	int	Command Element ID	Unique; Index; Primary_Key;	Not Null	The Commands Element ID is used to identify an individual command element
code_id	int	Code ID	Foreign_Key;	Not Null	The code id is the identifying command_element_type
test_data	int	Test Data		Null	The input used in tests

3.2.6.3 Relationships

- The code_id references a code in the codes table and is the identifier for the type that the command element is.
- A command_element must have a least one association to an automation table
- A command_element may have more than one association to an automation table
- A command_element must have at least one reference to a command_identifier
- A command_element may have more than one reference to a command_identifier



3.2.7 Command Identifier Table

3.2.7.1 Description

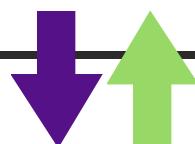
The command identifier table is used to store the identification type of a command element. This will include the ID Value, Name Value or CSS Value. The identifier is how the Web Driver will identify and look for the Web Element on the Satisfaction Survey Website.

3.2.7.2 Table Info

Column Name	Type	Descriptive Name	Column Properties	Allow Nulls	Description
command_identifier_id	int	Command Identifier ID	Unique; Index; Primary_Key;	Not Null	Used to identify an individual command identifier
command_element_id	int	Command Element ID	Index; Foreign Key;	Not Null	Used to associate this identifier to a command element
code_id	int	Code ID	Foreign Key;	Not Null	Used to identify the COMMAND IDENTIFIER TYPE
value	int	Value		Null	Contains the actual value that will be the identifier.

3.2.7.3 Relationships

- A code_identifier will have one and only one code_id associated. The Code Id will be of COMMAND_IDENTIFIER_TYPE value.
- A command_identifier will have one and only one command_element that it is associated to.



3.2.8 Form Table

3.2.8.1 Description

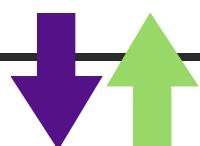
The form is used to provide separation from the question and the form elements.

3.2.8.2 Table Info

Column Name	Type	Descriptive Name	Column Properties	Allow Nulls	Description
form_id	int	Form ID	Unique; Index; Primary_Key;	Not Null	The Form ID is the unique id to identify the individual form

3.2.8.3 Relationships

- A form entry will always have only one question_id relation
- A form entry may have none or many form_elements relation associated to it.
- A form entry may have none or more form_flow entries associated to it.



3.2.9 Form Elements Table

3.2.9.1 Description

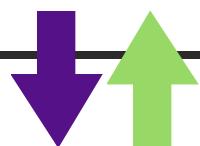
The form_elements table is the relational table that matches a form_element to a form_id

3.2.9.2 Table Info

Column Name	Type	Descriptive Name	Column Properties	Allow Nulls	Description
form_elements_id	int	Form Element ID	Unique; Index; Primary_Key;	Not Null	The Form Element ID is used to identify an individual reference
form_id	int	Automation ID	Index; Foreign_Key;	Not Null	Reference to the Form ID
form_element_id	int	Form Element ID	Foreign_Key;	Not Null	The reference to the form_element
weight	int	Weight		Not Null	Weight is the specific priority order of the element in the list

3.2.9.3 Relationships

- A form_element entry will only have one form_id relation
- A form_element entry will only have one form_element relation



3.2.10 Form Element Table

3.2.10.1 Description

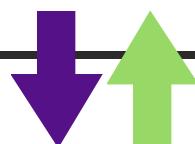
The form element table represents each individual commands that is listed in the automation aspect of a revision. The form_element represents all the different type of form_element, but is identified by the code_id

3.2.10.2 Table Info

Column Name	Type	Descriptive Name	Column Properties	Allow Nulls	Description
form_element_id	int	Form Element ID	Unique; Index; Primary_Key;	Not Null	The Form Element ID is used to identify an individual form element
code_id	int	Code ID	Foreign_Key;	Not Null	The code id is the identifying form_element_type

3.2.10.3 Relationships

- The code_id references a code in the codes table and is the identifier for the type that the command element is.
- A form_element must have a least one association to an form table
- A form_element may have more than one association to an form table
- A form_element may have at least one reference to a form_element_attribute
- A form_element may have more than one reference to a form_element_attribute
- A form_element may have at least one reference to a form_element_option
- A form_element may have more than one reference to a form_element_option



3.2.11 Form Element Attribute Table

3.2.11.1 Description

The form element attribute table contains the attribute fields that will be used for display purposes of that element. For instance, if the form element was a TEXTBOX, then the attribute may be the SIZE property of how long the textbox will display.

3.2.11.2 Table Info

Column Name	Type	Descriptive Name	Column Properties	Allow Nulls	Description
form_element_attribute_id	int	Form Element Attribute	Unique; Index; Primary_Key; ID	Not Null	Used to identify an individual reference
form_element_id	int	Form Element ID	Index; Foreign_Key;	Not Null	Reference to the Form Element ID
code_id	int	Code ID	Foreign_Key;	Not Null	Used to identify the FORM ATTRIBUTE TYPE
value	text	Value		Not Null	Contains the value of the attribute

3.2.11.3 Relationships

- A form_element_attribute will have one and only one code_id associated to it. The Code will be of type FORM_ELEMENT_ATTRIBUTE_TYPE.
- A form_element will have one and only one form_element associated to it.



3.2.12 Form Element Option Table

3.2.12.1 Description

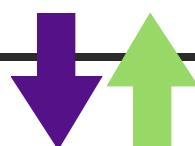
The Form Element Option Table will contain the options that are offered to the user to provide very easy of use. The options will be text values for a TEXTBOX and TEXTAREA field. The will be options for a RADIO, CHECKBOX, SELECTBOX elements.

3.2.12.2 Table Info

Column Name	Type	Descriptive Name	Column Properties	Allow Nulls	Description
form_element_option_id	int	Form Element Option ID	Unique; Index; Primary_Key;	Not Null	Used to identify an individual reference
form_element_id	int	Form Element ID	Index; Foreign_Key;	Not Null	Reference to the Form Element ID
display_text_id	int	Display Text ID	Foreign_Key;	Not Null	Reference the instance of the Display Text
value	text	Value		Not Null	Contains the value of the attribute

3.2.12.3 Relationships

- The form_element_option will have one and only one display_text associated to it.
- The form_element_option will have one and only one form_element associated to it.



3.2.13 Display Text Table

3.2.13.1 Description

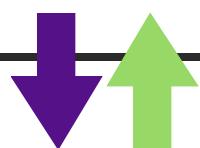
The display_text is a table that will hold the single representation of a text for displays.

3.2.13.2 Table Info

Column Name	Type	Descriptive Name	Column Properties	Allow Nulls	Description
display_text_id	int	Display Text ID	Unique; Index; Primary_Key;	Not Null	Used to identify an individual reference

3.2.13.3 Relationships

- display_text will have one or more display_text_translations.
- display_text can have one or more types of references where it is used.



3.2.14 Display Text Translation Table

3.2.14.1 Description

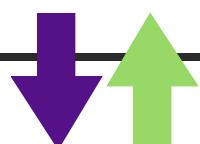
The Display Text Translation is simply a realization of a display_text for a specific language.

3.2.14.2 Table Info

Column Name	Type	Descriptive Name	Column Properties	Allow Nulls	Description
display_text_translation_id	int	Display Text Translation ID	Unique; Index; Primary_Key;	Not Null	Used to identify an individual reference
display_text_id	int	Display Text ID	Index; Foreign_Key;	Not Null	Used to identify the related Display Text Object
code_id	int	Code ID	Foreign_Key;	Not Null	Identifies the LANGUAGE TYPE
value	text	Value		Not Null	The value of the Display Text

3.2.14.3 Relationships

- display_text_translation will have one and only one display_text associated to it.
- display_text_translation will have one and only one code associated to it, used to identify the language. The code will be of type LANGUAGE_TYPE



3.2.15 Code Type

3.2.15.1 Description

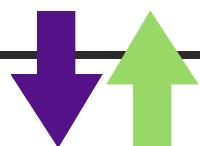
Code type is a type identifier for constants. It is what groups certain constants into a similar category.

3.2.15.2 Table Info

Column Name	Type	Descriptive Name	Column Properties	Allow Nulls	Description
code_type_id	int	Code Type ID	Unique; Index; Primary_Key;	Not Null	Used to identify an individual reference
type	text	Type		Not Null	Identifies the code_type but computer_ identifiable.
description	text	Description		Not Null	Identifiers the Code Type via human readable

3.2.15.3 Relationships

- code_type can have none or more codes associated to it.



3.2.16 Code

3.2.16.1 Description

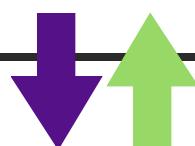
Code is simply a constant that is used in the system.

3.2.16.2 Table Info

Column Name	Type	Descriptive Name	Column Properties	Allow Nulls	Description
code_id	int	Code ID	Unique; Index; Primary_Key;	Not Null	Used to identify an individual reference
code_type_id	int	Code Type ID	Index; Foreign_Key;	Not Null	Used to identify the related code_type
code_value	text	Code Value	Unique; Index;	Not Null	A computer readable code value
name	text	Name		Not Null	Name of the code
description	text	Description		Not Null	Description of the code

3.2.16.3 Relationship

- code will have one and only one code_type associated to it
- code can be used by none or more areas of the system.



3.2.17 User Table

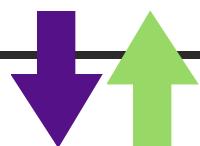
3.2.17.1 Description

This is the main table that holds the user information. Data in this table are User Objects. Currently the attributes of the user are the credentials of that user. However, in the future, there may be a user_attribute table in which additional attributes will be stored.

Note that this table currently not implemented as it is outside the scope of this project.

3.2.17.2 Table Info

Column Name	Type	Descriptive Name	Column Properties	Allow Nulls	Description
user_id	int	User ID	Unique; Index; Primary_Key;	Not Null	Used to identify an individual reference
username	text	Username	Unique; Index;	Not Null	The Username of the user
pin	text	Pin		Not Null	The password for the user



3.2.18 User Data Table

3.2.18.1 Description

User data table is the table that holds the Key information about a user. A key information example would be storing the persons GENDER, (male or female.)

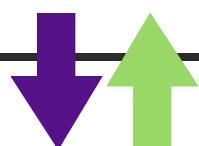
Note that this table currently not implemented as it is outside the scope of this project.

3.2.18.2 Table Info

Column Name	Type	Descriptive Name	Column Properties	Allow Nulls	Description
user_data_id	Int	User Data ID	Unique; Index; Primary_Key;	Not Null	Used to identify an individual reference
user_id	int	User ID	Index; Foreign_Key	Not Null	Used to identify the related User
code_id	int	Code ID	Index; Foreign_Key;	Not Null	Represents the USER DATA TYPE
value	text	Value		Not Null	The value of the data entry

3.2.18.3 Relationships

- user_data will have one and only one user_id associated to it.



3.2.19 User Setting Table

3.2.19.1 Description

User Setting is a separate table from the user_data table, whose purpose is to hold setting information. Such as when was the last time the user was logged in, or if the user is a premium user.

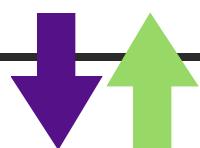
Note that this table currently not implemented as it is outside the scope of this project.

3.2.19.2 Table Info

Column Name	Type	Descriptive Name	Column Properties	Allow Nulls	Description
user_setting_id	Int	User Setting ID	Unique; Index; Primary_Key;	Not Null	Used to identify an individual reference
user_id	int	User ID	Index; Foreign_Key	Not Null	Used to identify the related User
code_id	int	Code ID	Index; Foreign_Key;	Not Null	Represents the USER SETTING TYPE
value	text	Value		Not Null	The value of the data entry

3.2.19.3 Relationships

- user_setting will have one and only one user_id associated to it.



3.2.20 Submission Table

3.2.20.1 Description

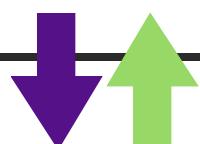
The submission table represents the Submission objects that the users submit their answers in. The submission object contains a list of the answers. In addition the actual reward that results from the submission of the submission object will be stored in an reward in the object.

3.2.20.2 Table Info

Column Name	Type	Description	Column Properties	Allow Nulls	Description
submission_id	int	Submission ID	Unique; Index; Primary_Key;	Not Null	Used to identify an individual reference
revision_id	int	Revision ID	Index; Foreign_Key;	Not Null	Identifies the related Revision entry
date_started	datetime	Date Started		Not Null	Is the time that the user started filling out the submission. This is used in case we need to save the answers as a partially filled out.
date_completed	datetime	Date Completed		Null	Is the time that the submission was fully completed and submitted
submission_reward_id	int	Submission Reward ID	Index; Foreign_Key;	Null	Is the reference to the associated reward. If the submission wasn't automated, then the reward would be null until automation is ran.

3.2.20.3 Relationships

- Submission has one or more submission_answers associated to it.
- A submission may or may not have one user_submission reference associated to it.
- A submission may have only one submission_reward associated to it.
- A submission must have one and only one reference to a Revision entry.



3.2.21 Submission Answer Table

3.2.21.1 Description

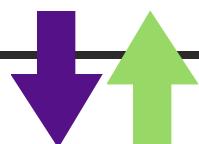
The submission answer table is the representation of the Answer object of the submission object. The answer may have a value, which will be what the user inputted. In addition the Answers contain all the question_ids and form_elements that the user went through on the form side.

3.2.21.2 Table Info

Column Name	Type	Descriptive Name	Column Properties	Allow Nulls	Description
submission_answer_id	int	Submission Answer ID	Unique; Index; Primary_Key;	Not Null	Used to identify an individual reference
submission_id	int	Submission ID	Index; Foreign_Key;	Not Null	Identifies the submission object the answer is associated to
form_element_id	int	Form Element ID	Index; Foreign_Key;	Not Null	Identifies which form_element the answer is submitting for.
value	text	Value		Null	The value of the answer that the user submitted
question_id	int	Question ID	Index; Foreign_Key;	Not Null	The question that the answer was answered in.

3.2.21.3 Relationships

- An answer is a part of only and only one submission entry
- An answer has one and only one question reference associated to it.
- An answer has one and only one form_element associated to it.



3.2.22 Submission Reward Table

3.2.22.1 Description

Submission Reward is the actual Reward that is resultant of a successful submission. The reward is created only when the web driver is ran and completed a submission. The reward has an id indicating what type of reward it is, and the value of the code or coupon.

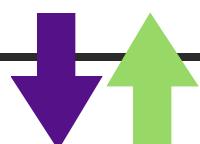
During the automation, if the form has receipt reward code, then the web driver will find the reward with that form.

3.2.22.2 Table Info

Column Name	Type	Descriptive Name	Column Properties	Allow Nulls	Description
submission_reward	int	Submission Reward	Unique; Index; Primary Key	Not Null	Used to identify an individual reference
reward_id	int	Reward ID	Index; Foreign_Key	Not Null	The actual realization of a Reward. The identifier for that reward that it is.
value	text	Value		Null	The value of the reward
redeemed	boolean	Redeemed		Not Null	Determines whether or not the reward has been redeemed by the user

3.2.22.3 Relationships

- A submission_reward has one and only one submission associated to it.
- A submission_reward has one and only one reward that it follows after.



3.2.23 Question Table

3.2.23.1 Description

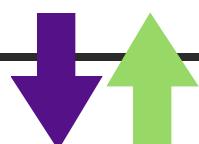
The question table is representation of the Question Objects. Questions are a group, made up of a form and automation. The question has no other properties other than providing the node like structure needed for the multiple flows.

3.2.23.2 Table Info

Column Name	Type	Descriptive Name	Column Properties	Allow Nulls	Description
question_id	int	Question ID	Unique; Index; Primary_Key;	Not Null	Used to identify an individual reference
name	text	Name		Null	This is a meaningful name of which to reference a specific question instance
form_id	int	Form ID	Index; Foreign_Key;	Not Null	The referenced form_id that the question is made up of
automation_id	int	Automation ID	Index; Foreign_Key;	Not Null	The referenced automation_id that the question is composed of

3.2.23.3 Relationships

- One and only one question will be referenced by a single revision entry.
- A question will have one and only one automation entry associated to it.
- A question will have one and only one form entry associated to it.
- A question may have none to many automation_flows which reference it.
- A question may have none to many form_flows which reference it.
- A question may have none to many submission_answers which reference it.



3.2.24 Reward Table

3.2.24.1 Description

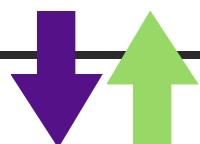
The Reward Table is composed of the Reward objects which are a template for what the reward for a certain revision will be. The submission_rewards will use the Reward entry as a guide to what will be stored and rewarded for that submission.

3.2.24.2 Table Info

Column Name	Type	Descriptive Name	Column Properties	Allow Nulls	Description
reward_id	int	Reward ID	Unique; Index; Primary_Key;	Not Null	Used to identify an individual reference
code_id	int	Code ID	Index; Foreign_Key;	Null	The code id is an indicator as to what type of reward that it is.
form_id	int	Form ID	Index; Foreign_Key;	Not Null	References a form entry

3.2.24.3 Relationships

- A reward will have one and only one revision entry to which it is associated with.
- A reward has one and only one form entry associated to it.
- A reward may be associated to none to many submission_rewards which reference it.
- A reward has one and only one code entry which identifies what type of reward it is.



3.2.25 Command Form Element Relation Table

3.2.25.1 Description

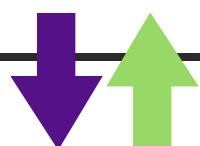
The Command Form Element Relation table is a relation table which only provides the associations between a command_element and a form_element. This association is important because when the web driver is going through the command_elements and a value is needed to be submitted, (such as a textbox,) the web driver will look at all the submission_answers and use the value with the corresponding form_element_id as is stored in the Command Form Element Relation Table.

3.2.25.2 Table Info

Column Name	Type	Description Name	Column Properties	Allow Nulls	Description
command_form_element_relation_id	int	Command Form Relation ID	Unique; Index; Primary_Key;	Not Null	Used to identify an individual reference
command_element_id	int	Command Element ID	Index; Foreign_Key;	Null	The referenced Command Element
form_element_id	int	Form Element ID	Index; Foreign_Key;	Not Null	The referenced Form Element

3.2.25.3 Relationships

- A command_form_element_relation has one and only one command_element associated.
- A command_form_element_relation has one and only one form_element associated.



3.2.26 Form Flow Table

3.2.26.1 Description

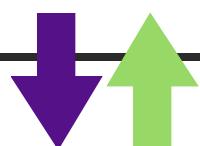
The Form Flow provides the ability for the node (question) to know where the possible next nodes are. Depending on what the user has submitted, it will look for the flow with that value first, otherwise it will use the default flow. This flow is on the form side of the question.

3.2.26.2 Table Info

Column Name	Type	Description	Column Properties	Allow Nulls	Description
form_flow_id	int	Form Flow ID	Unique; Index; Primary_Key;	Not Null	Used to identify an individual reference
form_id	int	Form ID	Index; Foreign_Key;	Null	The form that the flow is attached to
question_id	int	Question ID	Index; Foreign_Key;	Not Null	The question that the flow will go next to
value	text	Value		Null	The value that the flow will choose if it matches this value. The default flow will have value as empty

3.2.26.3 Relationships

- A form_flow will have one and only one form associated to it.
- A form_flow will have one and only one question entry to which it points next to.



3.2.27 Automation Flow Table

3.2.27.1 Description

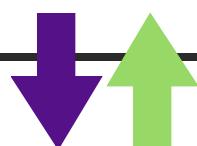
The Automation Flow provides the ability for the node (question) to know where the possible next nodes are. The Automation Flows are prioritized by numerical weight. It will go through the priority and look to see if the question_id is in the array of answers which all have a question_id associated to them. If that question is present, it means that the question was presented to the user and it should then go to that question in the flow. This flow is on the automation side of the question.

3.2.27.2 Table Info

Column Name	Type	Descriptive Name	Column Properties	Allow Nulls	Description
automation_flow_id	int	Automation Flow ID	Unique; Index; Primary_Key;	Not Null	Used to identify an individual reference
automation_id	int	Automation ID	Index; Foreign_Key;	Null	The automation that the flow is attached to
question_id	int	Question ID	Index; Foreign_Key;	Not Null	The question that the flow will go next to
weight	int	Weight		Null	The prioritization of the flows that the web driver will iterate through. Ascending prioritization

3.2.27.3 Relationships

- An automation_flow has one and only one automation entry associated to it.
- An automation_flow has one and only one question entry associated to it.



Appendix

Appendix A: Code Types

These are an example of the Code Types that are found in the system. The code types are essentially constants used in the system.

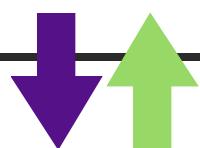
Code Type

FORM_ELEMENT_TYPE

COMMAND_ELEMENT_TYPE

CODE_TYPE

LANGUAGE_TYPE





WEB SERVICE SPECIFICATION

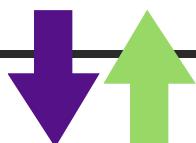
8/23/2013

Graduate Capstone

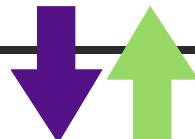


Table of Contents

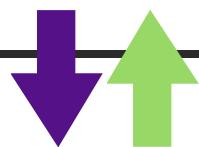
1 Introduction.....	5
1.1 Intended Audience.....	5
1.2 References.....	5
1.3 Revision History	5
2 Technical Documentation.....	6
2.1 Architecture Overview	6
2.2 Technical Requirements	6
2.3 HTTP Status Codes.....	7
2.4 Security	7
2.5 Testing	7
3 Web Service Definitions	8
3.1 Automation.....	9
3.1.1 Base Mappings	9
3.1.2 GET Mappings	9
3.1.3 POST Mappings.....	10
3.1.4 PUT Mappings	10
3.1.5 DELETE Mappings.....	11
3.1.6 Other Mappings	11
3.2 Automation Flow	12
3.2.1 Base Mappings	12
3.2.2 GET Mappings	12
3.2.3 POST Mappings.....	13
3.2.4 PUT Mappings	13
3.2.5 DELETE Mappings.....	13
3.3 Code	14
3.3.1 Base Mappings	14
3.3.2 GET Mappings	14
3.3.3 POST Mappings.....	15
3.3.4 PUT Mappings	15
3.3.5 DELETE Mappings.....	16
3.4 Code Type	17
3.4.1 Base Mappings	17
3.4.2 GET Mappings	17
3.4.3 POST Mappings.....	17
3.4.4 PUT Mappings	18
3.4.5 DELETE Mappings.....	18
3.5 Command Element.....	18
3.5.1 Base Mappings	19
3.5.2 GET Mappings	19



3.5.3 POST Mappings.....	19
3.5.4 PUT Mappings	20
3.5.5 DELETE Mappings.....	20
3.6 Command Form Element Relation	21
3.6.1 Base Mappings	21
3.6.2 GET Mappings	21
3.6.3 POST Mappings.....	21
3.6.4 PUT Mappings	22
3.6.5 DELETE Mappings.....	22
3.7 Command Identifier	24
3.7.1 Base Mappings	24
3.7.2 GET Mappings	24
3.7.3 POST Mappings.....	25
3.7.4 PUT Mappings	25
3.7.5 DELETE Mappings.....	26
3.8 Company	26
3.8.1 Base Mappings	27
3.8.2 GET Mappings	27
3.8.3 POST Mappings.....	27
3.8.4 PUT Mappings	28
3.8.5 DELETE Mappings.....	28
3.9 Form	28
3.9.1 Base Mappings	29
3.9.2 GET Mappings	29
3.9.3 POST Mappings.....	29
3.9.4 PUT Mappings	30
3.9.5 DELETE Mappings.....	30
3.10 Form Element Attribute.....	30
3.10.1 Base Mappings.....	30
3.10.2 GET Mappings.....	30
3.10.3 POST Mappings.....	31
3.10.4 PUT Mappings	31
3.10.5 DELETE Mappings	31
3.11 Form Element.....	32
3.11.1 Base Mappings	32
3.11.2 GET Mappings	32
3.11.3 POST Mappings.....	33
3.11.4 PUT Mappings	33
3.11.5 DELETE Mappings	33
3.12 Form Element Option.....	34
3.12.1 Base Mappings	34
3.12.2 GET Mappings.....	34



3.12.3 POST Mappings.....	35
3.12.4 PUT Mappings.....	35
3.12.5 DELETE Mappings	36
3.13 Form Flow	36
3.13.1 Base Mappings.....	36
3.13.2 GET Mappings.....	36
3.13.3 POST Mappings.....	37
3.13.4 PUT Mappings.....	37
3.13.5 DELETE Mappings	37
3.13 Question.....	38
3.13.1 Base Mappings	38
3.13.2 GET Mappings.....	38
3.13.3 POST Mappings.....	39
3.13.4 PUT Mappings.....	39
3.13.5 DELETE Mappings	39
3.14 Revision.....	40
3.14.1 Base Mappings.....	41
3.14.2 GET Mappings.....	41
3.14.3 POST Mappings.....	41
3.14.4 PUT Mappings.....	41
3.14.5 DELETE Mappings	42



1 Introduction

This document is created to provide technical information required in making web service calls from the client applications. The scope of the document includes specification of the web service and descriptions.

1.1 Intended Audience

This documented is intended for individuals of a technical background including but not limited to developers, testers, product teams or administrators.

1.2 References

- <http://www.healthit.gov/sites/default/files/nhin-web-services-registry-production-specification-v2.0-1.pdf>
- <http://www.fdsn.org/webservices/FDSN-WS-Specifications-1.0.pdf>
- http://heimshelp.deewr.gov.au/sites/heimshelp/Support/Documents/CHESSNTech_nicalSpecifications.pdf
- <http://www-sk.let.uu.nl/u/d2r-6b.pdf>
- http://www.civilservice.gov.uk/wp-content/uploads/2012/04/ePIMS_Benchmarking_Web_Service_Specification.pdf

1.3 Revision History

Name	Date	Reason For Change	Version
Andy Bottom	05/28/2013	Started the formatting and organizing of the document.	0.1
Andy Bottom	08/19/2013	Updated the web services	0.2
Andy Bottom	08/21/2013	Finished the Technical documentation of the web services	1.0



2 Technical Documentation

The following section contains a technical explanation and overview of the web services application.

2.1 Architecture Overview

The Web Services are used in the system in order to provide data and actions to the client application. The largest role that the Web Services acts as is the Data Access Layer for the system. This by doing this, all interaction with the Database is completely removed and only limited to the services.

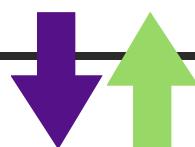
2.2 Technical Requirements

The Web Services are developed in Java and are implemented via utilizing the Spring MVC Framework.

The Web Services follows the RESTful API. This API follows the HTTP 1.1 principals. The accessing of the Web Services relies on protocols such as Transmission Control Protocol/Internet Protocol (TCP/IP) and Hypertext Transfer Protocol (HTTP).

The operations that are utilized in the Web Services are the following explained in the following table and how they are used in the system.

Operation	Description
GET	The operation is used for retrieving data from the database.
POST	Passes in an object in the request which is then added as a new entry in the desired table.
PUT	Performs an update of a preexisting entry in a table.
DELETE	Removes an entry from a table in the database.



2.3 HTTP Status Codes

With performing requests over HTTP, occasionally request fail due to an number of factors. When these failures occur, a status code is returned to the client application. Below is a list and description of the most common codes that are encountered when attempting connection with Web Services.

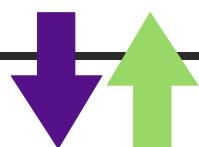
Code	Description
200	A Successful request and response.
400	Bad request due to missing data or parameter
401	Unauthorized authentication or access
500	Internal Server Error
503	The Service is temporarily unavailable or under maintenance

2.4 Security

Due to the open nature of the Web Services, security is very important. However, the extra security designs are outside the scope of the current project. However, currently future plans to implement security are created. The plan is to eventually implement HMAC Security as a design method that will secure the web services of the system.

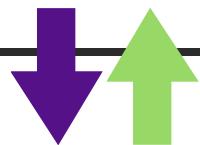
2.5 Testing

Testing for the web services are outside the scope of the current project. However, plans to perform automated testing of the web services are in the works. The plan is to use the program SoapUI to create automated tests on each of the modules of the Data Access Layer. By creating these tests, we can ensure that all the data entering the services will pass all validations and formatting. This will create an extra benefit of data quality in the database.



3 Web Service Definitions

The following section is an exhaustive list of all the definitions of the web services. Since this is the data access layer of the system, each database has a designated service to each table. In each service, there are mapping for being able to perform all CRUD actions on the database.



3.1 Automation

The Automation Object is part of the question which holds a list of command elements.

3.1.1 Base Mappings

/Automation

3.1.2 GET Mappings

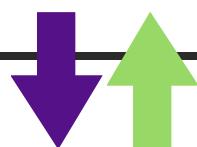
The following methods are used to simple get an Automation or List of Automation Objects.

3.1.2.1 Get by Automation Id

Category	Value
Mapping	/ById/Automation/{automationId}
URL Parameters	automationId: Integer. Represents the Id of the desired Automation. Should Return
Request	None
Response	Automation Object

3.1.2.2 Get by Revision Id

Category	Value
Mapping	/ById/Revision/{revisionId}
URL Parameters	revisionId: Integer. Represents the Revision ID of which the automations are associated with.
Request	None
Response	List of Automation Objects



3.1.3 POST Mappings

The following mappings are used to create a new Automation entry into the database.

3.1.3.1 Create

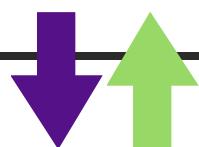
Category	Value
Mapping	/Create/
URL Parameters	None
Request	Automation Object
Response	Automation Object with new Automation Id Number

3.1.4 PUT Mappings

The following mappings are used to update an existing Automation entry in the database.

3.1.4.1 Update Automation's Command Order/Weight

Category	Value
Mapping	/Update/Weight/
URL Parameters	None
Request	Automation Object with all Command Elements Present in the correct order
Response	Automation Object with the Command Elements in the correct order.



3.1.5 DELETE Mappings

The following mappings are used to remove Automation entries from the database.

3.1.5.1 Remove by Automation Id

Category	Value
Mapping	/ById/Automation/{automationId}
URL Parameters	AutomationId: Integer; The Id of the Automation to delete
Request	None
Response	None

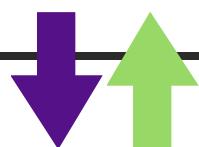
3.1.5.2 Remove by Automation Id

Category	Value
Mapping	/ById/Revision/{revisionId}
URL Parameters	revisionId: Integer; The Id of the Revision which all Automations to delete are associated with.
Request	None
Response	None

3.1.6 Other Mappings

3.1.6.1 Perform Automation Test

Category	Value
Mapping	/Action/Test/Revision/{revisionId}
URL Parameters	revisionId: Integer; The Id of the Revision for which the test automation should be run.
Request	None
Response	None



3.2 Automation Flow

3.2.1 Base Mappings

/AutomationFlow

3.2.2 GET Mappings

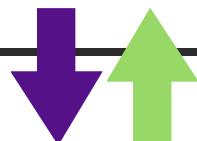
The following methods are used to simple get an Automation Flow or List of Automation Flow Objects.

3.2.2.1 Get by Automation Flow Id

Category	Value
Mapping	/ById/AutomationFlow/{automationFlowId}
URL Parameters	automationFlowId: Integer. Represents the Id of the desired Automation Flow.
Request	None
Response	Automation Flow Object

3.2.2.2 Get by Automation Id

Category	Value
Mapping	/ById/Automation/{automationId}
URL Parameters	automationId: Integer. Represents the Automation ID of which the automation Flows are associated with.
Request	None
Response	List of Automation Flow Objects



3.2.3 POST Mappings

The following mappings are used to create a new Automation Flow entry into the database.

3.2.3.1 Create

Category	Value
Mapping	/Create/
URL Parameters	None
Request	Automation Flow Object
Response	Automation Flow Object with new Automation Flow Id Number

3.2.4 PUT Mappings

None

3.2.5 DELETE Mappings

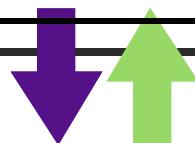
The following mappings are used to remove Automation Flow entries from the database.

3.2.5.1 Remove by Automation Flow Id

Category	Value
Mapping	/ById/AutomationFlow/{automationFlowId}
URL Parameters	automationFlowId: Integer; The Id of the Automation Flow to delete
Request	None
Response	None

3.2.5.2 Remove by Automation Id

Category	Value
Mapping	/ById/Automation/{automationId}
URL Parameters	automationId: Integer; The Id of the Automation which all Automation Flows are associated with.
Request	None
Response	None



3.2.5.3 Remove by Question Id

Category	Value
Mapping	/ById/Question/{questionId}
URL Parameters	questionId: Integer; The Id of the Question.
Request	None
Response	None

3.3 Code

3.3.1 Base Mappings

/Code

3.3.2 GET Mappings

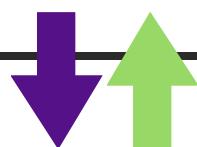
The following methods are used to simple get a Codes or List of Codes Objects.

3.3.2.1 Get All Codes

Category	Value
Mapping	/All/
URL Parameters	None
Request	None
Response	List of Codes

3.3.2.2 Get by Code Type ID

Category	Value
Mapping	/ById/CodeType/{codeTypeId}
URL Parameters	codeTypeId: Integer. Represents the Code Type ID that the code is associated with.
Request	None
Response	List of Code Objects



3.3.2.3 Get by Code Id

Category	Value
Mapping	/ById/Code/{codeId}
URL Parameters	codeId: Integer. Represents the Code ID of the Code
Request	None
Response	Code Object

3.3.2.4 Get by Code Id

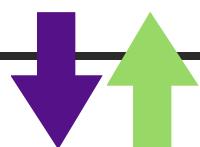
Category	Value
Mapping	/ByCode/Code/{code}
URL Parameters	code: String. Represents the Code of the Code Object
Request	None
Response	List of Code Objects

3.3.3 POST Mappings

The following mappings are used to create a new Code entry into the database.

3.3.3.1 Create

Category	Value
Mapping	/Create/
URL Parameters	None
Request	Code Object
Response	Code Object with new Code Id Number



3.3.4 PUT Mappings

The following mappings are used to update an existing Code entry in the database.

3.3.4.1 Update Code

Category	Value
Mapping	/Update/
URL Parameters	None
Request	Code Object with the Updated Values
Response	Code Object with the Updated Values

3.3.5 DELETE Mappings

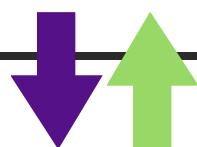
3.3.5.1 Remove by Code Id

The following mappings are used to remove Code entries from the database.

Category	Value
Mapping	/ById/Code/{codeId}
URL Parameters	codeId: Integer; The Id of the Code
Request	None
Response	None

3.3.5.2 Remove by Code

Category	Value
Mapping	/ByCode/Code/{code}
URL Parameters	code: String; The Code of the Code Object
Request	None
Response	None



3.4 Code Type

3.4.1 Base Mappings

/CodeType

3.4.2 GET Mappings

The following methods are used to simple get a Code Type or List of Code Type Objects.

3.4.2.1 Get All Codes

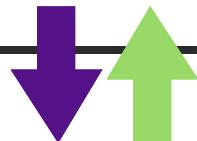
Category	Value
Mapping	/All/
URL Parameters	None
Request	None
Response	List of Code Types

3.4.2.2 Get by Code Type ID

Category	Value
Mapping	/ById/CodeType/{codeTypeId}
URL Parameters	codeTypeId: Integer; Represents the Code Type ID;
Request	None
Response	CodenType

3.4.2.3 Get by Type

Category	Value
Mapping	/ByType/CodeType/{type}
URL Parameters	type: String; Represents the Type of the Code Type;
Request	None
Response	Code Type Object



3.4.3 POST Mappings

The following mappings are used to create a new Code Type entry into the database.

3.4.3.1 Create

Category	Value
Mapping	/Create/
URL Parameters	None
Request	Code Type Object
Response	Code Type Object with new Code Type Id;

3.4.4 PUT Mappings

The following mappings are used to update an existing Code Type entry in the database.

3.4.4.1 Update

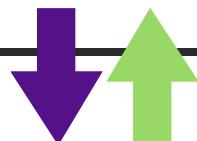
Category	Value
Mapping	/Update/
URL Parameters	None
Request	Code Type Object with the Updated Values
Response	Code Type Object with the Updated Values

3.4.5 DELETE Mappings

The following mappings are used to remove Code Type entries from the database.

3.4.5.1 Remove by Code Type Id

Category	Value
Mapping	/ById/CodeType/{codeTypeId}
URL Parameters	codeTypeId: Integer; The Id of the Code Type;
Request	None
Response	None



3.5 Command Element

3.5.1 Base Mappings

/CommandElement

3.5.2 GET Mappings

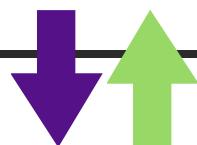
The following mappings are used to simple get a Command Element or List of Command Element Objects.

3.5.2.1 Get by Command Element Id

Category	Value
Mapping	/ById/CommandElement/{commandElementId}
URL Parameters	commandElementId: Integer; Represents the Id of the Command Element
Request	None
Response	Command Element

3.5.2.2 Get by Automation Id

Category	Value
Mapping	/ById/Automation/{automationId}
URL Parameters	automationId: Integer; Represents the Automation ID;
Request	None
Response	List of Command Elements



3.5.3 POST Mappings

The following mappings are used to create a new Command Element entry into the database.

3.5.3.1 Create

Category	Value
Mapping	/Create/{automationId}
URL Parameters	automationId: Integer; Represents the Id of the Automation that the Command Element will be associated with;
Request	Command Element Object
Response	Command Element Object with new Command Element Id;

3.5.4 PUT Mappings

3.5.4.1 Update

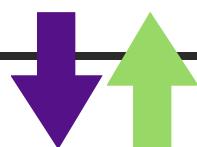
Category	Value
Mapping	/Update/
URL Parameters	None
Request	Command Element Object with the Updated Values
Response	Command Element Object with the Updated Values

3.5.5 DELETE Mappings

The following mappings are used to remove Command Element entries from the database.

3.5.5.1 Remove by Command Element Id

Category	Value
Mapping	/ById/CommandElement/{commandElementId}
URL Parameters	commandElementId: Integer; The Id of the Command Element
Request	None
Response	None



3.6 Command Form Element Relation

3.6.1 Base Mappings

/CommandFormElementRelation

3.6.2 GET Mappings

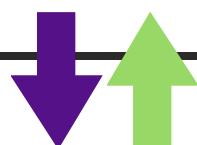
The following mappings are used to simple get a Command Form Element Relation or List of Command Form Element Relation Objects.

3.6.2.1 Get by Command Element Id

Category	Value
Mapping	/ById/CommandElement/{commandElementId}
URL Parameters	commandElementId: Integer; Represents the Id of the Command Element
Request	None
Response	List of Command Form Element Relation

3.6.2.2 Get by Form Element Id

Category	Value
Mapping	/ById/FormElement/{formElementId}
URL Parameters	formElementId: Integer; Represents the Form Element;
Request	None
Response	List of Command Form Element Relation



3.6.3 POST Mappings

The following mappings are used to create a new Command Form Element Relation entry into the database.

3.6.3.1 Create

Category	Value
Mapping	/Create/
URL Parameters	None
Request	CommandFormElementRelation Object
Response	CommandFormElementRelation Object with new Command Form Element Relation Id;

3.6.4 PUT Mappings

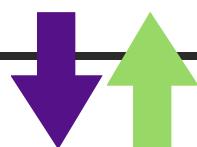
None;

3.6.5 DELETE Mappings

The following mappings are used to remove Command Form Element Relation entries from the database.

3.6.5.1 Remove by Command Form Element Relation Id

Category	Value
Mapping	/ById/CommandFormElementRelation/{ commandFormElementRelationId }
URL Parameters	commandFormElementRelationId: Integer; The Id of the Command Form Element Relation
Request	None
Response	None

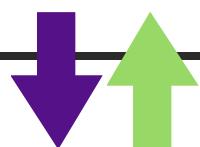


3.6.5.2 Remove by Command Element Id

Category	Value
Mapping	/ById/CommandElement/{commandElementId}
URL Parameters	commandElementId: Integer; The Id of the Command Element
Request	None
Response	None

3.6.5.3 Remove by Form Element Id

Category	Value
Mapping	/ById/FormElement/{formElementId}
URL Parameters	formElementId: Integer; The Id of the Form Element
Request	None
Response	None



3.7 Command Identifier

3.7.1 Base Mappings

/CommandIdentifier

3.7.2 GET Mappings

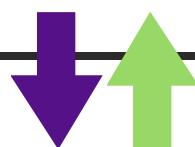
The following mappings are used to simple get a Command Identifier or List of Command Identifier Objects.

3.7.2.1 Get by Command Identifier Id

Category	Value
Mapping	/ById/CommandIdentifier/{commandIdentifierId}
URL Parameters	commandIdentifierId: Integer; Represents the Id of the Command Identifier
Request	None
Response	Command Identifier

3.7.2.2 Get by Command Element Id

Category	Value
Mapping	/ById/CommandElement/{commandElementId}
URL Parameters	commandElementId: Integer; Represents the Command Element;
Request	None
Response	List of Command Identifiers



3.7.3 POST Mappings

The following mappings are used to create a new Command Identifier entry into the database.

3.7.3.1 Create

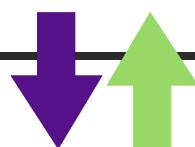
Category	Value
Mapping	/Create/{commandElementId}
URL Parameters	commandElementId: Integer; Represents the Id of the Command Element that the Command Identifier will be associated with;
Request	Command Identifier Object
Response	Command Identifier Object with new Command Identifier Id;

3.7.4 PUT Mappings

The following mappings are used to update an existing Command Identifier entry in the database.

3.7.4.1 Update

Category	Value
Mapping	/Update/
URL Parameters	None;
Request	Command Identifier Object
Response	Command Identifier Object;



3.7.5 DELETE Mappings

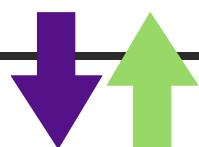
The following mappings are used to remove Command Identifiers entries from the database.

3.7.5.1 Remove by Command Identifier Id

Category	Value
Mapping	/ById/CommandIdentifier/{commandIdentifierId}
URL Parameters	commandIdentifierId: Integer; The Id of the Command Identifier;
Request	None
Response	None

3.7.5.2 Remove by Command Element Id

Category	Value
Mapping	/ById/CommandElement/{commandElementId}
URL Parameters	commandElementId: Integer; The Id of the Command Element
Request	None
Response	None



3.8 Company

3.8.1 Base Mappings

/Company

3.8.2 GET Mappings

The following mappings are used to simple get a Company or List of Company Objects.

3.8.2.1 Get All

Category	Value
Mapping	/All
URL Parameters	None;
Request	None;
Response	List of Company Objects

3.8.2.2 Get by Company ID

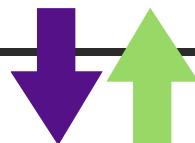
Category	Value
Mapping	/ById/Company/{companyId}
URL Parameters	companyId: Integer; Represents the Company;
Request	None;
Response	Company Object

3.8.3 POST Mappings

The following mappings are used to create a new Company entry into the database.

3.8.3.1 Create

Category	Value
Mapping	/Create/
URL Parameters	None;
Request	Company Object
Response	Company Object with new Company Id



3.8.4 PUT Mappings

The following mappings are used to update an existing Company entry in the database.

3.8.4.1 Update

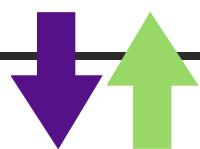
Category	Value
Mapping	/Update/
URL Parameters	None;
Request	Company Object
Response	Company Object;

3.8.5 DELETE Mappings

The following mappings are used to remove Company entries from the database.

3.8.5.1 Remove by Company ID

Category	Value
Mapping	/ById/Company/{companyId}
URL Parameters	companyId: Integer; The Id of the Company;
Request	None
Response	None



3.9 Form

3.9.1 Base Mappings

/Form

3.9.2 GET Mappings

The following mappings are used to simple get a Form or List of Form Objects.

3.9.2.1 Get by Form Id

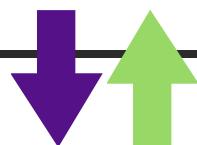
Category	Value
Mapping	/ById/Form/{formId}
URL Parameters	formId: Integer; Represents the Id of the Form;
Request	None;
Response	Form Objects;

3.9.2.2 Get by Revision ID

Category	Value
Mapping	/ById/Revision/{revisionId}
URL Parameters	revisionId: Integer; Represents the Id of the Revision;
Request	None;
Response	List of Form Objects;

3.9.3 POST Mappings

There are no creation mappings for inserting a Form. The Forms are automatically created upon a creation of a Question Object.



3.9.4 PUT Mappings

The following mappings are used to update an existing Form entry in the database.

3.9.4.1 Update

Category	Value
Mapping	/Update/Weight/
URL Parameters	None;
Request	Form Object with List of Form Elements in a correct order/weight
Response	Form Object with List of Form Elements in the new order/weight

3.9.5 DELETE Mappings

There are no mappings for deleting Form Objects. All removals are occurring automatically by the system.

3.10 Form Element Attribute

3.10.1 Base Mappings

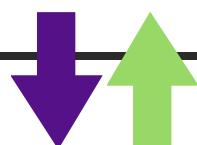
/FormElementAttribute

3.10.2 GET Mappings

The following mappings are used to simple get a Form Element Attribute or List of Form Element Attribute Objects.

3.10.2.1 Get by Form Element Attribute Id

Category	Value
Mapping	/ById/FormElementAttribute/{formElementAttributeId}
URL Parameters	formElementAttributeId: Integer; Represents the Id of the Form Element Attribute Object;
Request	None;
Response	Form Element Attribute Object;



3.10.2.2 Get by Form Element Id

Category	Value
Mapping	/ById/FormElement/{formElementId}
URL Parameters	formElementId: Integer; Represents the Id of the Form Element;
Request	None;
Response	List of Form Element Attribute Objects

3.10.3 POST Mappings

The following mappings are used to create a new Form Element Attribute entry into the database.

3.10.3.1 Create

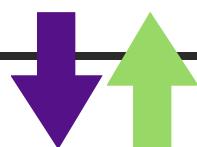
Category	Value
Mapping	/Create/{formElementId}
URL Parameters	formElementId: Integer; Represents the Id of the Form Element Object that the Form Element Attribute will be associated with;
Request	Form Element Attribute Object;
Response	Form Element Attribute Object with new Form Element Attribute Id

3.10.4 PUT Mappings

The following mappings are used to update an existing Form Element Attribute entry in the database.

3.10.4.1 Update

Category	Value
Mapping	/Update/
URL Parameters	None;
Request	Form Element Attribute Object;
Response	Form Element Attribute Object;



3.10.5 DELETE Mappings

The following mappings are used to remove Form Element Attribute entries from the database.

3.10.5.1 Remove by Form Element Attribute ID

Category	Value
Mapping	/ById/FormElementAttribute/{formElementAttributeId}
URL Parameters	formElementAttributeId: Integer; The Id of the Form Element Attribute Object;
Request	None
Response	None

3.11 Form Element

3.11.1 Base Mappings

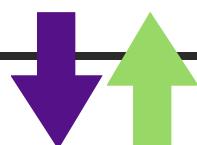
/FormElement

3.11.2 GET Mappings

The following mappings are used to simple get a Form Element or List of Form Element Objects.

3.11.2.1 Get by Form Element Id

Category	Value
Mapping	/ById/FormElement/{formElementId}
URL Parameters	formElementId: Integer; Represents the Id of the Form Element Object;
Request	None;
Response	Form Element Object;



3.11.2.2 Get by Form Id

Category	Value
Mapping	/ById/Form/{formId}
URL Parameters	formId: Integer; Represents the Id of the Form;
Request	None;
Response	List of Form Element Objects;

3.11.3 POST Mappings

The following mappings are used to create a new Form Element entry into the database.

3.11.3.1 Create

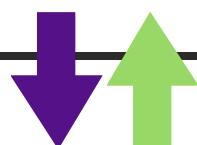
Category	Value
Mapping	/Create/{formId}
URL Parameters	formId: Integer; Represents the Id of the Form Object that the Form Element will be associated with;
Request	Form Element Object;
Response	Form Element Object with new Form Element Id;

3.11.4 PUT Mappings

The following mappings are used to update an existing Form Element entry in the database.

3.11.4.1 Update

Category	Value
Mapping	/Update/
URL	None;
Parameters	
Request	Form Element Object;
Response	Form Element Object;



3.11.5 DELETE Mappings

The following mappings are used to remove Form Element entries from the database.

3.11.5.1 Remove by Form Element ID

Category	Value
Mapping	/ById/FormElement/{formElementId}
URL Parameters	formElementId: Integer; The Id of the Form Element Object;
Request	None;
Response	None;

3.12 Form Element Option

3.12.1 Base Mappings

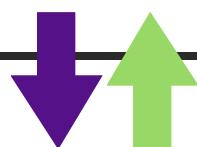
/FormElementOption

3.12.2 GET Mappings

The following mappings are used to simple get a Form Element Option or List of Form Element Options Objects.

3.12.2.1 Get by Form Element Option Id

Category	Value
Mapping	/ById/FormElementOption/{formElementOptionId}
URL Parameters	formElementOptionId: Integer; Represents the Id of the Form Element Option Object;
Request	None;
Response	Form Element Option Object;



3.12.2.2 Get by Form Element Id

Category	Value
Mapping	/ById/FormElement/{formElementId}
URL Parameters	formElementId: Integer; Represents the Id of the Form Element;
Request	None;
Response	List of Form Element Option Objects

3.12.3 POST Mappings

The following mappings are used to create a new Form Element Option entry into the database.

3.12.3.1 Create

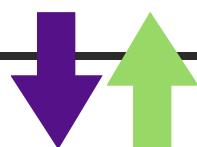
Category	Value
Mapping	/Create/{formElementId}
URL Parameters	formElementId: Integer; Represents the Id of the Form Element Object that the Form Element Option will be associated with;
Request	Form Element Option Object;
Response	Form Element Option Object with new Form Element Option Id;

3.12.4 PUT Mappings

The following mappings are used to update an existing Form Element Option entry in the database.

3.12.4.1 Update

Category	Value
Mapping	/Update/
URL Parameters	None;
Request	Form Element Option Object;
Response	Form Element Option Object;



3.12.5 DELETE Mappings

The following mappings are used to remove Form Element Option entries from the database.

3.12.5.1 Remove by Form Element Option ID

Category	Value
Mapping	/ById/FormElementOption/{formElementOptionId}
URL Parameters	formElementOptionId: Integer; The Id of the Form Element Option Object;
Request	None
Response	None

3.13 Form Flow

3.13.1 Base Mappings

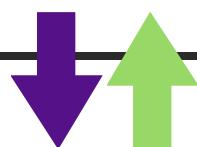
/FormFlow

3.13.2 GET Mappings

The following mappings are used to simple get a Form Flow or List of Form Flow Objects.

3.13.2.1 Get by Form Flow Id

Category	Value
Mapping	/ById/FormFlow/{formFlowId}
URL Parameters	formFlow: Integer; Represents the Id of the Form Flow Object;
Request	None
Response	Form Flow Object;



3.13.2.2 Get by Form Id

Category	Value
Mapping	/ById/Form/{formId}
URL Parameters	formId: Integer; Represents the Id of the Form;
Request	None
Response	List of Form Flow Objects

3.13.3 POST Mappings

The following mappings are used to create a new Form Flow entry into the database.

3.13.3.1 Create

Category	Value
Mapping	/Create/
URL Parameters	None;
Request	Form Flow Object;
Response	Form Flow Object with new Form Flow Id;

3.13.4 PUT Mappings

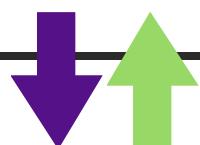
None;

3.13.5 DELETE Mappings

The following mappings are used to remove Form Flow entries from the database.

3.13.5.1 Remove by Form Flow ID

Category	Value
Mapping	/ById/FormFlow/{formFlowId}
URL Parameters	formFlowId: Integer; The Id of the Form Flow Object;
Request	None
Response	None



3.13.5.2 Remove by Form ID

Category	Value
Mapping	/ById/Form/{formId}
URL Parameters	formId: Integer; The Id of the Form Object;
Request	None
Response	None

3.13 Question

3.13.1 Base Mappings

/Question

3.13.2 GET Mappings

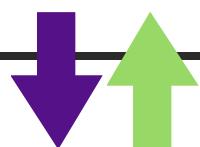
The following mappings are used to simple get a Question or List of Question Objects.

3.13.2.1 Get by Question Id

Category	Value
Mapping	/ById/Question/{questionId}
URL Parameters	questionId: Integer; Represents the Id of the Question Object;
Request	None;
Response	Question Object;

3.13.2.2 Get by Form Id

Category	Value
Mapping	/ById/Form/{formId}
URL Parameters	formId: Integer; Represents the Id of the Form;
Request	None;
Response	List of Question Objects;



3.13.2.3 Get by Automation Id

Category	Value
Mapping	/ById/Automation/{automationId}
URL Parameters	automationId: Integer; Represents the Id of the Automation;
Request	None;
Response	List of Question Objects;

3.13.3 POST Mappings

The following mappings are used to create a new Question entry into the database.

3.13.3.1 Create

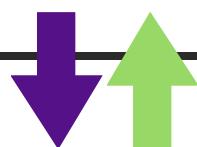
Category	Value
Mapping	/Create/
URL Parameters	None;
Request	Question Object;
Response	Question Object with new Question Id;

3.13.4 PUT Mappings

The following mappings are used to update an existing Question entry in the database.

3.13.4.1 Update

Category	Value
Mapping	/Update/
URL Parameters	None;
Request	Question Object;
Response	Question Object;



3.13.5 DELETE Mappings

The following mappings are used to remove Question entries from the database.

3.13.5.1 Remove by Question ID

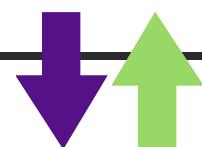
Category	Value
Mapping	/ById/Question/{questionId}
URL Parameters	questionId: Integer; The Id of the Question Object;
Request	None
Response	None

3.13.5.2 Remove by Form ID

Category	Value
Mapping	/ById/Form/{formId}
URL Parameters	formId: Integer; The Id of the Form Object;
Request	None
Response	None

3.13.5.3 Remove by Automation ID

Category	Value
Mapping	/ById/Automation/{automationId}
URL Parameters	automationId: Integer; The Id of the Automation Object;
Request	None
Response	None



3.14 Revision

3.14.1 Base Mappings

/Revision

3.14.2 GET Mappings

The following mappings are used to simple get a Revision or List of Revision Objects.

3.14.2.1 Get by Revision Id

Category	Value
Mapping	/ById/Revision/{revisionId}
URL Parameters	revisionId: Integer; Represents the Id of the Revision;
Request	None;
Response	Revision Objects;

3.14.2.3 Get by Survey Id

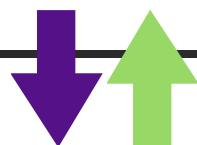
Category	Value
Mapping	/ById/Survey/{surveyId}
URL Parameters	surveyId: Integer; Represents the Id of the Survey;
Request	None;
Response	List of Revision Objects

3.14.3 POST Mappings

The following mappings are used to create a new Revision entry into the database.

3.14.3.1 Create

Category	Value
Mapping	/Create/
URL Parameters	None;
Request	Revision Object;
Response	Revision Object with new Revision Id;



3.14.4 PUT Mappings

The following mappings are used to update an existing Revision entry in the database.

3.14.4.1 Update

Category	Value
Mapping	/Update/
URL Parameters	None;
Request	Revision Object;
Response	Revision Object;

3.14.5 DELETE Mappings

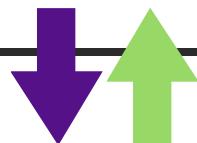
The following mappings are used to remove Revision entries from the database.

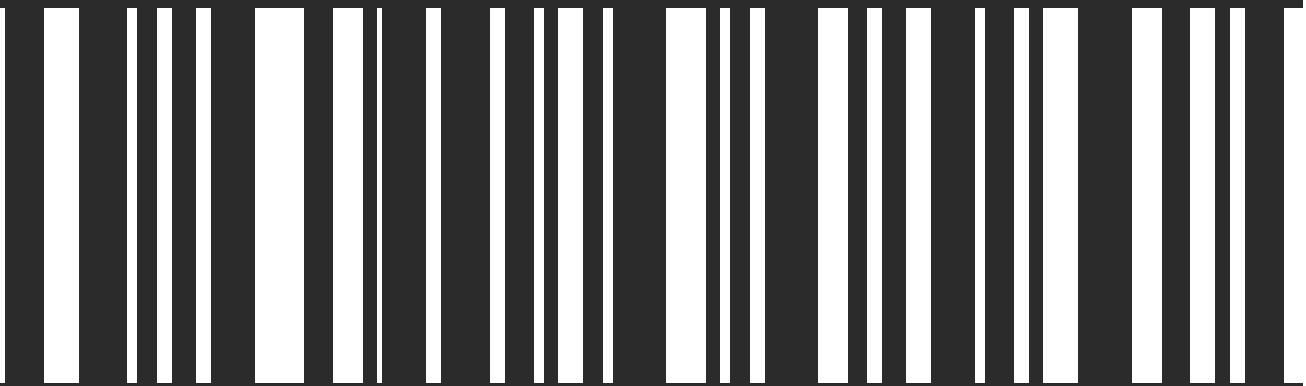
3.14.5.1 Remove by Revision ID

Category	Value
Mapping	/ById/Question/{revisionId}
URL Parameters	revisionId: Integer; The Id of the Revision Object;
Parameters	
Request	None
Response	None

3.14.5.2 Remove by Survey ID

Category	Value
Mapping	/ById/Form/{surveyId}
URL Parameters	surveyId: Integer; The Id of the Survey Object;
Request	None
Response	None





RISK MANAGEMENT PLAN

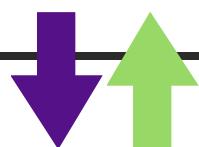
8/23/2013

Graduate Capstone



Table of Contents

1 Introduction.....	3
1.1 Intended Audience.....	3
1.2 References.....	3
1.3 Revision History	3
2 Approach to Risk Management.....	4
2.1 Risk Categories	4
2.1.1 Cost Risk	4
2.1.2 Schedule Risk.....	4
2.1.3 User Risk	4
2.1.4 Operational Risk.....	4
2.1.5 Technical Risks	4
2.1.6 Legal Risks	4
2.2 Risk Rating System	5
2.2.1 Likelihood of Occurrence	5
2.2.2 Risk Impact.....	5
2.2.3 Risk Score.....	6
2.3 Response Strategy.....	7
2.3.1 Mitigation	7
2.3.2 Acceptance.....	7
2.3.3 Avoidance	7
2.3.4 Transference	7
3 Risk Assessment.....	8
3.1 Definitions	8
3.2 Form Template.....	8
3.3 Table Template	8
Appendix.....	9
Appendix A: Risk Form Template	9
Appendix B: Risk Assessment Table Template	10



1 Introduction

This document's purpose is to give instructions and guidance regarding the identifying and handling of risks that may occur in the system.

1.1 Intended Audience

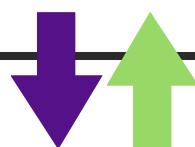
This document is written and is geared toward individuals of a medium level understanding of technical background and who have a medium level understanding of the project scope itself.

1.2 References

- <http://www.cs.odu.edu/~cpi/411/orangs11/web/plans/Risk%20Management%20Plan.doc>
- http://www.oceanleadership.org/wp-content/uploads/2009/04/OOI_CDR_RiskMgtPlan.pdf
- <http://www.nd.gov/itd/files/services/pm/risk-management-plan-sample.pdf>
- http://international.fhwa.dot.gov/riskassess/risk_hcm06_05.cfm

1.3 Revision History

Name	Date	Reason For Change	Version
Andy Bottom	04/29/2013	Started compiling the plan into the document	0.1
Andy Bottom	05/06/2013	Finished the first version of the risk document	1.0



2 Approach to Risk Management

2.1 Risk Categories

2.1.1 Cost Risk

This risk type is used to identify risks involving finances and costs.

2.1.2 Schedule Risk

This risk type identifies those occurrences that could negatively impact the schedule and push back deadlines. This risk may show a cascading effect on features after the occurrence. Risk is very time oriented focus.

2.1.3 User Risk

This risk involves the occurrences directly related to external users using the system. Includes security from users, both handling sensitive data and potential hackers attempting to break into the system.

2.1.4 Operational Risk

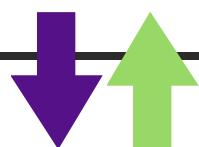
Directly results from Operational changes such as a change in ability to manage systems.

2.1.5 Technical Risks

Involve risks regarding technical systems, such as servers being overloaded.

2.1.6 Legal Risks

This involves legal ramifications that may impact the project and bring in outside requirements. In other words, identifying and ensuring that laws are met.



2.2 Risk Rating System

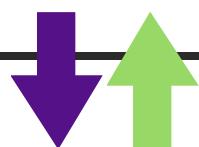
In order to effectively measure the risks, the following section explains in a meaningful way what the impact of a risk may indicate.

2.2.1 Likelihood of Occurrence

Numeric Score	Natural Language	Probability Value of Calculations	Probability Range
1	Remote	5%	1% - 10%
2	Unlikely	26%	11% - 40%
3	Possible	51%	41% - 60%
4	Probable	76%	61% - 90%
5	Near Certainty	95%	91% - 99%

2.2.2 Risk Impact

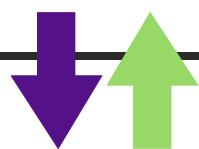
Numeric Score	Impact	Description
1	Negligible	If occurred, would have no impact on the project; Requirements would be unaffected;
3	Minor	If occurred, would cause small change in schedule; Requirements would remain intact;
5	Moderate	If occurred, would cause moderate increase in schedule; Important requirements would be unaffected.
8	Serious	If occurred, would cause large increase in schedule; Secondary requirements may not be achieved;
10	Critical	If occurred, it may cause potential project failure; Minimal requirements may not be achieved;



2.2.3 Risk Score

The matrix below is an example of the overall probability of occurrence and its related impact. The table is used to compare risks and gather a more observable prioritization between risks.

	Negligible (1)	Minor (3)	Moderate (5)	Serious (8)	Critical (10)
Remote Occurrence (1)	1	3	5	8	10
Unlikely Occurrence (2)	2	6	10	16	20
Possible Occurrence (3)	3	9	15	24	30
Probably Occurrence (4)	4	12	20	32	40
Near Certainty of Occurring (5)	5	15	25	40	50



2.3 Response Strategy

2.3.1 Mitigation

This technique is geared to minimize as much of the risk as possible or completely resolving it. Recommended if the risk cannot be avoided.

2.3.2 Acceptance

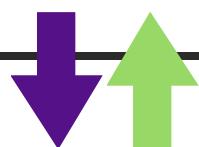
Essentially the team must accept the risk or completely ignore it till it occurs.

2.3.3 Avoidance

In this strategy, the action is to change the plan in hopes of completely avoiding the risk and prevent it from occurring.

2.3.4 Transference

This is the ability to transfer the risk to a third party or someone outside the scope of the project.



3 Risk Assessment

3.1 Definitions

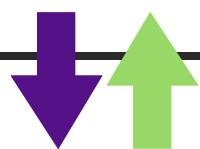
- Risk ID – is the unique id to identify the risk
- Risk Summary – is a description about the risk
- Response Action Summary – description about the action that is needs to be done

3.2 Form Template

Can be found in Appendix A

3.3 Table Template

Can be found in Appendix B



Appendix

Appendix A: Risk Form Template

<NAME>

Information

Risk Information

Risk ID	<Risk ID>
Risk Name	<Name>
Category	<Category>

Submitter Information

Name of Submitter	<User Name>
Date of Submission	<Date>

Risk Summary

<Description>

Risk Specifics

Likelihood	<Likelihood>
Impact	<Impact>
Response Action	<Response>

Response Action Summary

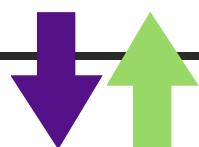
<Summary>

Risk Review

Review Name	<Name>
Date of Review	<Date>
Revised Impact	<Impact>

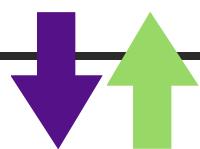
Review Summary

<Summary>



Appendix B: Risk Assessment Table Template

ID	Category	Likelihood	Impact	Response
<Risk ID>	<Category>	<Occurrence>	<Impact>	<Response>
<Risk ID>	<Category>	<Occurrence>	<Impact>	<Response>
<Risk ID>	<Category>	<Occurrence>	<Impact>	<Response>
<Risk ID>	<Category>	<Occurrence>	<Impact>	<Response>
<Risk ID>	<Category>	<Occurrence>	<Impact>	<Response>



RISK ASSESSMENT

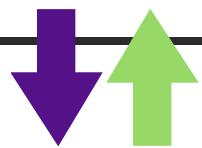
8/23/2013

Graduate Capstone



Table of Contents

1 Introduction.....	3
1.1 Intended Audience.....	3
1.2 References.....	3
1.3 Revision History	3
2.1 Design Complexity.....	5
2.2 Automation Work Load.....	6
2.3 Time Management.....	7
2.4 Development Team Diversity	8
2.5 Code Readability.....	9
2.6 Late User Prototyping.....	10
3 Risk Table.....	11



1 Introduction

The purpose of this document is to give a tutorial and walkthrough on how to navigate and use the Receipt Reward Application.

1.1 Intended Audience

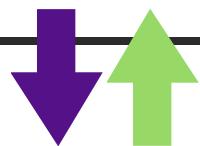
This document is intended for any persons who wish to utilize the Phone Application and gain a better understanding of how the screens work. This document is written for people with any degree of familiarity with the application.

1.2 References

No references were used.

1.3 Revision History

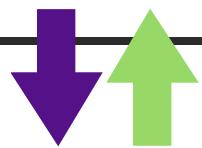
Name	Date	Reason For Change	Version
Andy Bottom	08/13/2013	Created the introduction and parts of the document.	0.1
Andy Bottom	08/20/2013	Finished the risk assessment.	1.0



2 Risk Forms

Below is a list of the risks that are involved with this project. The risks are in no particular order besides the date that they were logged.

For more information about what the risks are, please see the Risk Management Plan document for more information about the risks.



2.1 Design Complexity

Information

Risk ID	RISK-01	Risk Owner	Andy Bottom
Risk Name	Design Complexity	Date Submitted	04/29/2013
Category	TECHNICAL		
Score			

Risk Summary

May cause the system to be hard to maintain and make improvements for future iterations

Risk Specifics

Likelihood	Remote
Impact	Moderate
Response	Mitigation

Response Plan:

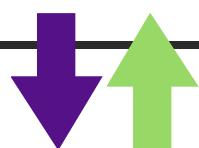
The plan is to document thoroughly the structure of the system.

Risk Review

Review Name	Andy Bottom
Date of Review	04/29/2013
Revised Impact	Mitigation

Review Summary

The plan is to document thoroughly the structure of the system.



2.2 Automation Work Load

Information

Risk ID	RISK-02	Risk Owner	Andy Bottom
Risk Name	Automation Work Load	Date Submitted	05/04/2013
Category	OPERATIOAL		
Score	MEDIUM-HIGH		

Risk Summary

The amount of load on the automation server may cause it to be bogged down and show very long waits and slow response times.

Risk Specifics

Likelihood	4
Impact	8
Response	Mitigation

Response Plan:

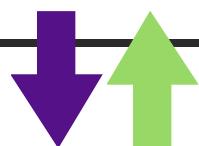
The plan to mitigate the risk is to either optimize the automation to be very fast. Or the more plausible plan is to add additional automation servers to split up the load per server and thus have faster responses.

Risk Review

Review Name	
Date of Review	
Revised Impact	

Review Summary

(Empty box for review summary)



2.3 Time Management

Information

Risk ID	RISK-03	Risk Owner	Andy Bottom
Risk Name	Time Management	Date Submitted	04/016/2013
Category	Schedule		
Score	LOW-MEDIUM		

Risk Summary

Basically this is identifying that my schedule can get very busy at times with balancing internship work and family, so my time as a resource could be a potential risk.

Risk Specifics

Likelihood	2
Impact	8
Response	ACCEPTANCE

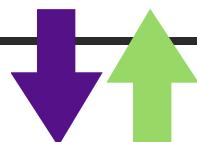
Response Plan:

This is something that cannot be prevented, thus the only action is to accept that this may occur. To reduce the impact, working to keep ahead of schedule can help with the impact.

Risk Review

Review Name	
Date of Review	
Revised Impact	

Review Summary



2.4 Development Team Diversity

Information

Risk ID	RISK-04	Risk Owner	Andy Bottom
Risk Name	Development Team Diversity	Date Submitted	05/07/2013
Category	OPERATIONAL		
Score	6		

Risk Summary

Essential, this risk is identifying that since I am the only resource that is involved with this project, that it may be a risk that there is not other point of view or opinion involved with design choices.

Risk Specifics

Likelihood	5
Impact	4
Response	ACCEPTANCE

Response Plan:

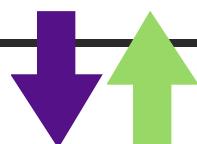
There is nothing that I can do about being the only one involved with the project. To lessen the impact, I will constantly run my ideas by with fellow classmates, co-workers, friends and professors to get outside opinion and advice involved with the project.

Risk Review

Review Name	
Date of Review	
Revised Impact	

Review Summary

--



2.5 Code Readability

Information

Risk ID	RISK-05
Risk Name	Code Readability
Category	TECHNICAL
Score	7

Risk Owner	Andy Bottom
Date Submitted	05/07/2013

Risk Summary

Essentially, there is always a risk involved with iterative development that the code can get very messy and hard to maintain

Risk Specifics

Likelihood	2
Impact	8
Response	Mitigation

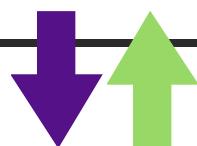
Response Plan:

I created a document specifying the code design specifications. Also, a need for the developers to have a discipline to keep code clean and add comments when necessary is vital.

Risk Review

Review Name	
Date of Review	
Revised Impact	

Review Summary



2.6 Late User Prototyping

Information

Risk ID	RISK-06	Risk Owner	Andy Bottom
Risk Name	Late User Prototyping	Date Submitted	04/29/2013
Category	OPERATIONAL		
Score	7		

Risk Summary

Because I will be doing a bottom-up approach to the development, one of the major disadvantages is that the phone application and the User Experience aspect won't be developed to the very end. This is a risk because I will have limited time to perform User Acceptance Testing and also may not have time to Clean up, fine-tune or fix small bugs on the phone side of things

Risk Specifics

Likelihood	8
Impact	4
Response	Acceptance

Response Plan:

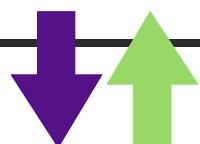
This is just the nature of the beast. Try to start some form of phone prototyping earlier so that integration on that end will be somewhat easier than starting from scratch late in the game.

Risk Review

Review Name	Andy Bottom
Date of Review	08/15/2013
Revised Impact	

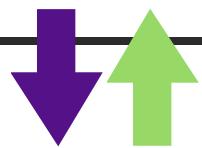
Review Summary

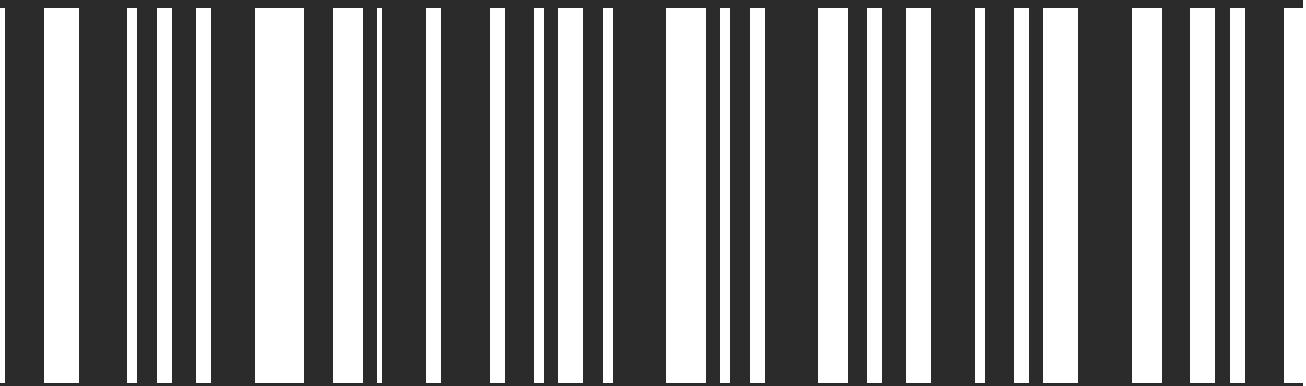
As expected, this is what happened. I did start earlier on initial phone development. But did run out of time to tidy-up and make a very clean Phone UX. Something that will need to be improved upon.



3 Risk Table

ID	Category	Likelihood	Impact	Response
RISK-01	TECHNICAL	REMOTE	MODERATE	MITIGATE
RISK-02	OPERATIONAL	4	8	MITIGATE
RISK-03	SCHEDULE	2	8	ACCEPTANCE
RISK-04	OPERATIONAL	5	4	ACCEPTANCE
RISK-05	TECHNICAL	2	8	MITIGATE
RISK-06	OPERATIONAL	4	2	ACCEPTANCE





TEST PLAN INSTRUCTIONS DOCUMENT

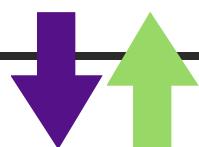
8/23/2013

Graduate Capstone



Table of Contents

1 Introduction.....	3
1.1 Intended Audience	3
1.2 References.....	3
1.3 Revision History	3
2 Test Methodology.....	4
2.1 Elements of Testing	4
2.2 Items to Be Tested	4
2.3 Items Not To Be Tested.....	4
3 Types of Testing	5
3.1 Functional Testing	5
3.2 Data Testing	6
3.3 User Testing (User Acceptance Testing)	7
3.4 Non-Functional Testing.....	7
3.5 Performance Testing	8
3.6 Integration Testing	9
3.6.1 Bottom Up Approach.....	9
3.6.2 Integration Plan.....	10
3.7 Unit Testing	11
Appendix.....	12
Appendix A: Functional Test Log Example	13
Appendix B: Integration Test Result Template	14
Appendix C: Integration Test Results	15
INT-01	15
INT-02	16
INT-03	17
INT-04	18
INT-05	19
INT-06	20
INT-07	21
INT-08	22
INT-09	23



1 Introduction

The purpose of this document is to create a clear description about the approach to testing that will be done on the system. Testing is a crucial part to the development process, and must be understood and followed to ensure quality of the code and validation of the requirements.

1.1 Intended Audience

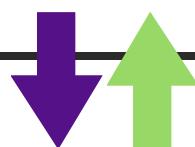
This document is intended for users of a moderate technical background. It is ideal for users to be very familiar with the system. In general, this document should be used by testers as a reference of all the testing involved in the system.

1.2 References

- http://www.dir.texas.gov/SiteCollectionDocuments/IT%20Leadership/Framework/Framework%20Extensions/SDLC/SDLC_testPlan_instructions.pdf
- <http://www.engr.sjsu.edu/gaojerry/course/287/TestPlan.doc>
- <http://www.jiludwig.com/templates/ATTemplate.doc>
- http://ltodi.est.ips.pt/es/index_files/pdf/TestPlanOutline.doc
- http://www.smithmusic123.com/application/views/download/download_file/Testing.pdf
- <http://sucs.org/~tobeon/testing.pdf>

1.3 Revision History

Name	Date	Reason For Change	Version
Andy Bottom	06/24/2013	Created the formatting of the document and added references	0.1
Andy Bottom	08/16/2013	Updated Testing Descriptions and added Integration Test Ret	0.2
Andy Bottom	08/22/2013	Finished Test Plan document.	1.0



2 Test Methodology

Below is a generic summary about the testing methodology.

2.1 Elements of Testing

Testing requires very thorough and dedication. By testing, the hopes is to improve code quality, reduce the amount of defects and validate the fulfillment of the requirements.

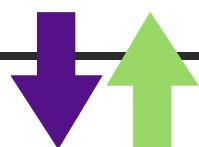
2.2 Items to Be Tested

- Integration Testing

2.3 Items Not To Be Tested

Due to time and scope constraints, formal Data Testing will not be done during the Capstone. Informal checks of the database may be done to ensure that data quality is at a passable level as to not cause critical bugs.

For more information about items that will not be developed, please refer to the [Vision and Scope Document](#).



3 Types of Testing

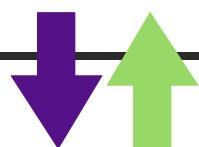
3.1 Functional Testing

Functional testing will be used to test all the functional requirements of the system. This type of test will ensure that the product is meeting the criteria of what the system was designed to do.

All the functional requirements can be found in the **Software Requirement Specification** document. Along with a list of the functional requirements, there are FIT criteria of which the tests are supposed to meet in order to pass.

If a functional requirement does not pass, and development related to those requirements has been completed, then a bug must be created to resolve the problem. Also, at this point, it may ultimately be decided that the requirement had changed from when it was last updated, and a new change to the requirement may be needed to be made to allow for the change. This change may then cause the functional pass to now pass and a bug may not need to be filed.

A log of the functional testing must be made in order to identify when in the testing process it was passed. Once a functional requirement was verified and passed functional testing, further testing of that requirement may not be needed to be done anymore. In this case, a passed requirement will be only tested in a larger smoke test of the entire system. The log can be found in this document under Appendix A.

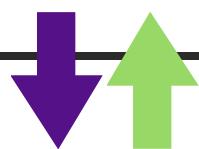


3.2 Data Testing

Data testing are tests that are done on the data in the database. The purpose of Data Testing is to ensure data quality of the system.

To determine the success criteria of a system, the data quality requirements will be found as a non-functional requirement of the database and system.

The types of quality standards that can be done with data testing may include but not limited to, relations in the database between entries. Ensuring no data is or become orphaned in the database; that all primary ids are unique and not duplicated. Other types of data testing may involve proper formatting of certain types of data. Such as all Date Fields will be formatted YYYY-MM-DD, or something similar.



3.3 User Testing (User Acceptance Testing)

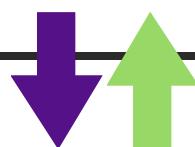
User Acceptance Testing will be a large aspect of the testing for this project. User Acceptance testing will be done to determine whether or not the product or prototype is on the correct path and meets the needs of individual users of the system. The importance of this type of testing cannot be understated. Because users will ultimately be the ones to use the product, it is vital that the application pass the tests from User Testing.

In addition to testing Non-Functional Requirements, you can also obtain very valuable information from User Acceptance Tests. Some of the data may determine if the product is meting the user needs, if the interface is helpful or inhibiting, if the overall experience of the application.

Whenever User Acceptance Testing is performed, the results will be logged so that the data can be used and acted upon the findings. The results of the User Acceptance Testing can be found in this document under Appendix B.

3.4 Non-Functional Testing

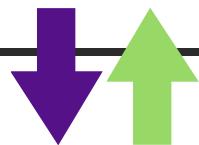
The non-functional tests in this project are spread out between the other forms of testing. Performance sorts of tests will be done in the Performance Testing. Other generic non-functional requirements will be tested in the Functional Tests. See section 3.1



3.5 Performance Testing

Performance Testing is done in respects to time and duration. The goal of performance testing is to test efficiency of the system. Performance Tests typically will come from Non-Functional Requirements. The success of the requirements will come from the FIT Criteria.

As mentioned before, performance testing can test thing such as the amount of wait time a call to the web server takes, or the size of the json file that is returned from the web services to the phone. The performance tests goal is to make the application faster for the user not to have to wait, and that the size of data is efficient to prevent sluggishness, memory leaks or database timeouts.



3.6 Integration Testing

Integration testing is performed to ensure parts, modules or code chunks were together with other code sets, when they are being integrated together, hence the name. The importance of these tests is to provide a set of tests during integration as a check that pieces are fitting together and unexpected bugs that occur are caught immediately.

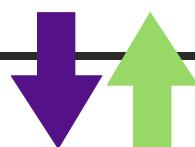
Due to the nature of my project, having 4 large standalone modules, the integration testing will be the most vital of the tests for this system. By doing integration tests, it will be the most efficient at identifying bugs during this stage.

Also, integration tests are great in doing black box testing which allows for very broad tests to be done to catch other inconsistencies and get these fix.

With the iterative type of software development, the integration test is also perfect fit. At the end of iterations, an integration test will be performed to ensure all changes work well with the existing product and functional prototype is resulted.

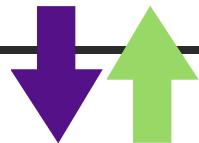
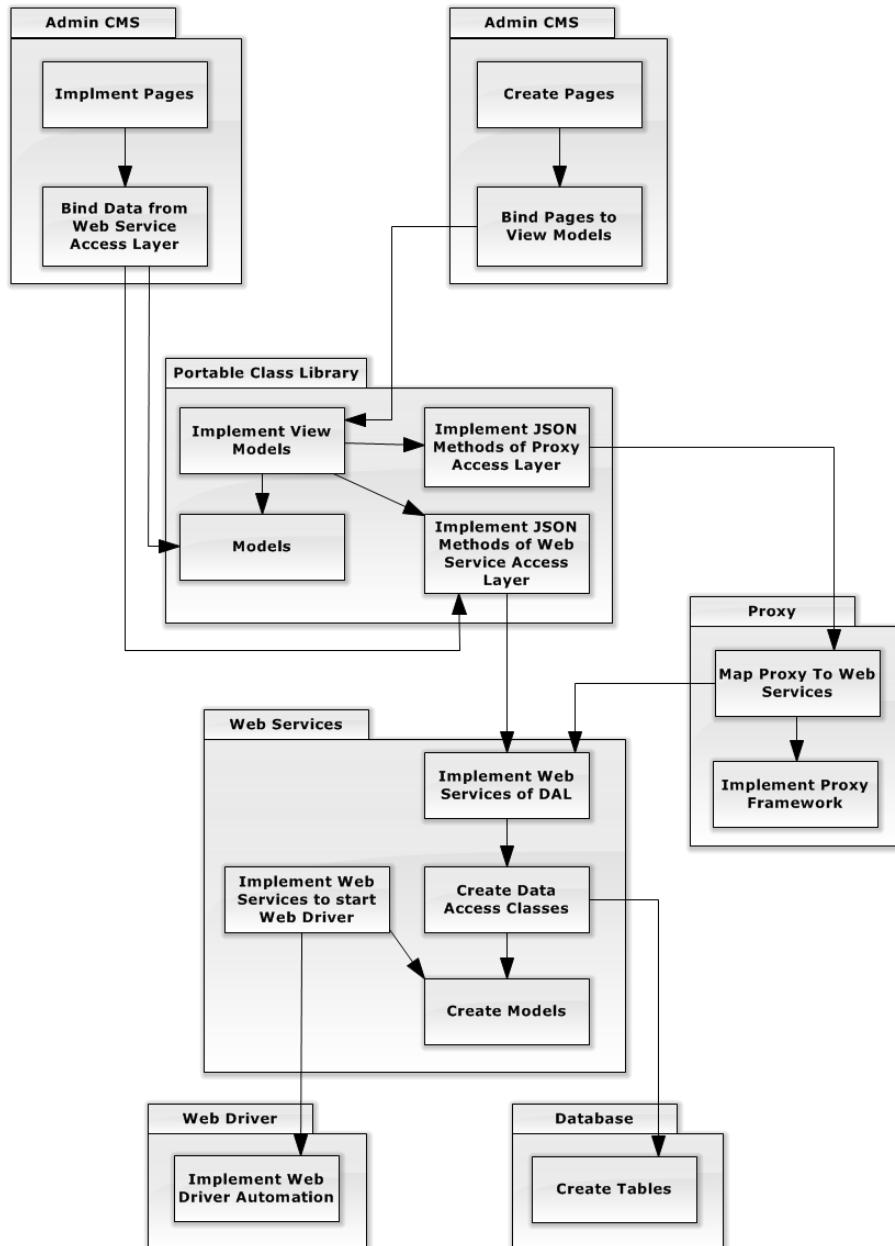
3.6.1 Bottom Up Approach

As specified in the **Iteration Document**, I will be developing with a Bottom-Up Approach. As with all things, everything has its benefits and disadvantages. With the bottom-up approach, the advantage is that the backend functionality will be tested first and built upon one another. This is advantageous because testing on database, web services and admin can all be done early. This will ensure the quality of the backend functionality. However, the disadvantage is that the phone application will be implemented and integrated last. This is problematic because functioning prototype won't be created to the very end, thus testing and quality of the phone app will be rushed thus opening the possibility of more bugs.



3.6.2 Integration Plan

The following is a diagram representing all the high level modules and sub-modules of the project. The lines represent the dependencies of each component.

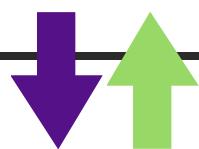


3.7 Unit Testing

Unit testing is essentially the use of automation to perform testing. Often times, a test case(s) should be created with a piece of functionality is created. This test case will be added to the test suite for later use. Often times the Unit test can be ran to ensure that no bugs are created. This is the advantage to catch a wide variety of bugs so that they don't occur and are prevented.

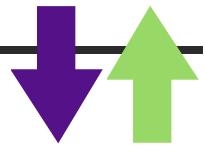
The disadvantage of unit testing is the development time will be needed to create the test cases, but this pays off in the long run as they provide solid testing down the line and no additional resource is needed to test since it is automation.

Even though the benefit of Unit Testing is evident, due to the limited amount of developer resources and time, Unit Testing will not be done for the scope of this project as valuable time will be needed to develop the entire system.



Appendix

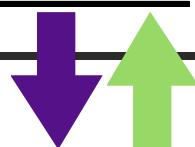
The following is the appendix of the document.



Appendix A: Functional Test Log Example

The functional testing will be referencing the Functional and Non-Functional Requirements (that can be found in the Software Requirement Specification.) All success criteria will ensure that it passes the FIT criteria.

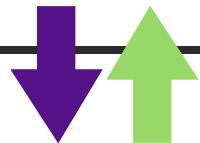
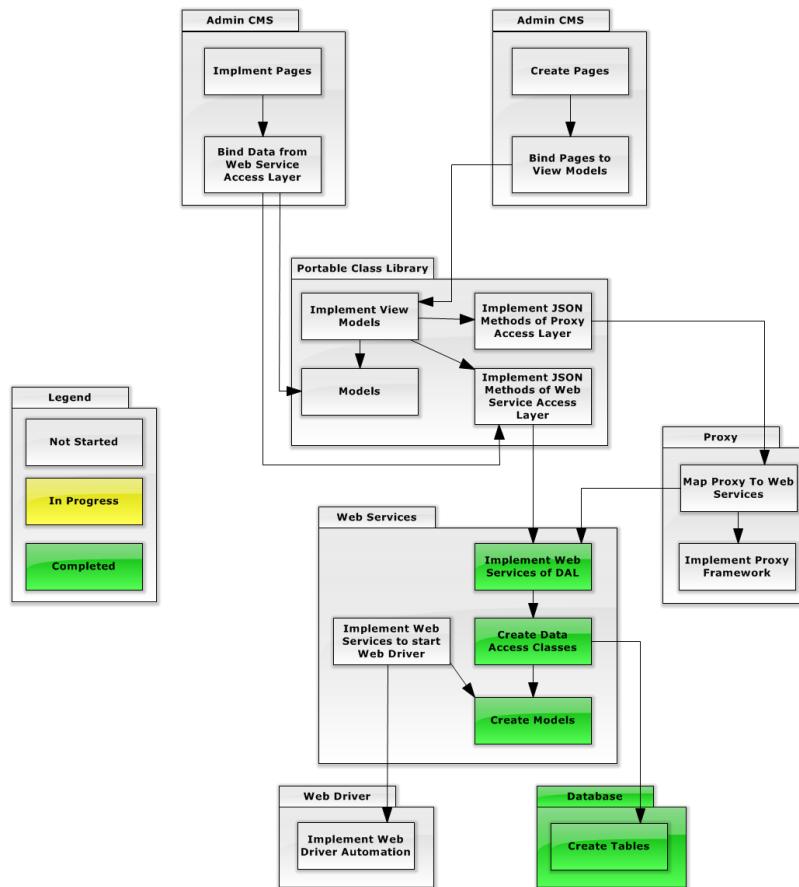
Req ID	Test Case Description	Date Tested	Pass / Fail	Test Result Description
REQ-01	Provide a brief description of the functionality the case will test	mm/dd/yy	Pass	
REQ-02	Add more lines as needed and remove blue text prior to use			
REQ-03				
REQ-04				
REQ-05				
REQ-06				
REQ-07				
REQ-08				
REQ-09				
REQ-10				
REQ-11				
REQ-12				
REQ-13				
REQ-14				
REQ-15				



Appendix B: Integration Test Result Template

The following is an example of the form used to log the results of the Integration Tests.

Integration ID	Test Author
Test Name	Date Tested
Modules Involved	
Test Summary	



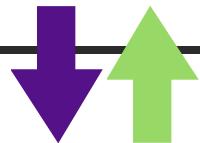
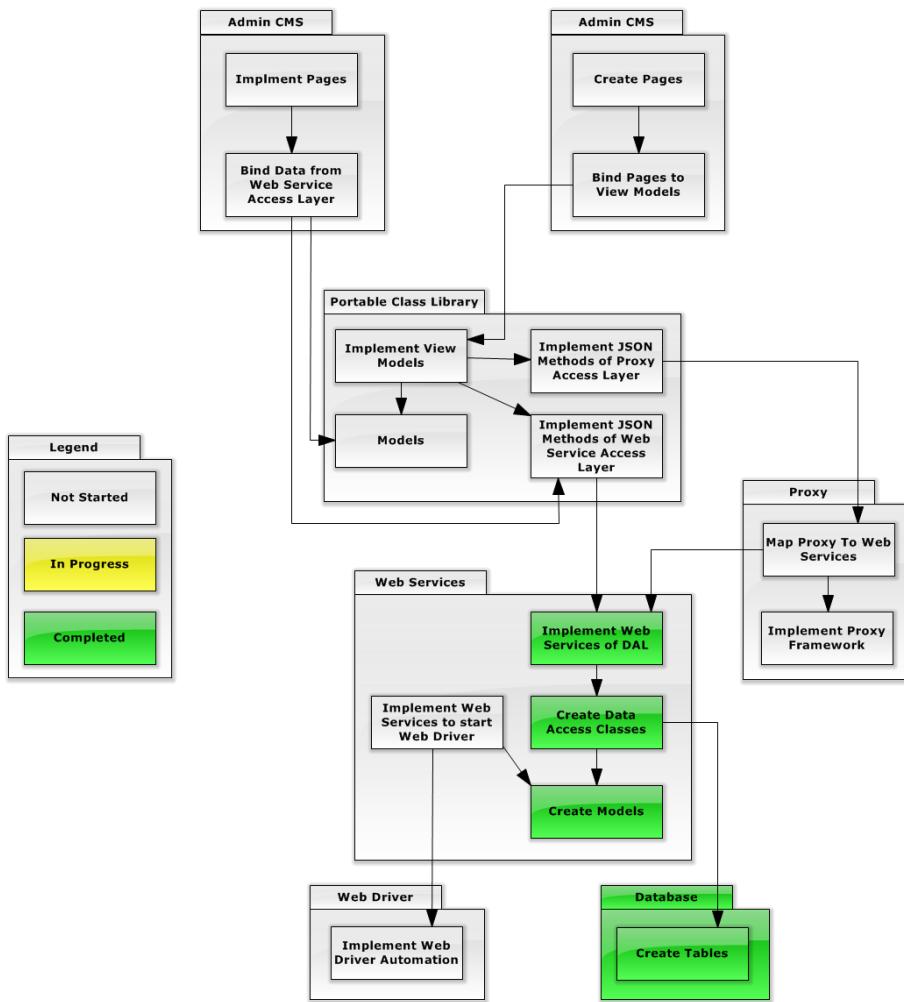
Appendix C: Integration Test Results

INT-01

Integration ID	INT-01	Test Author	Andy Bottom
Test Name	DAL Test	Date Tested	
Modules Involved	Web Services; Data Access Layer; Database;		

Test Summary

Tested Web Services and the interaction with the database.

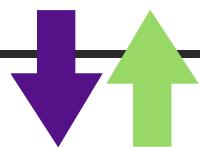
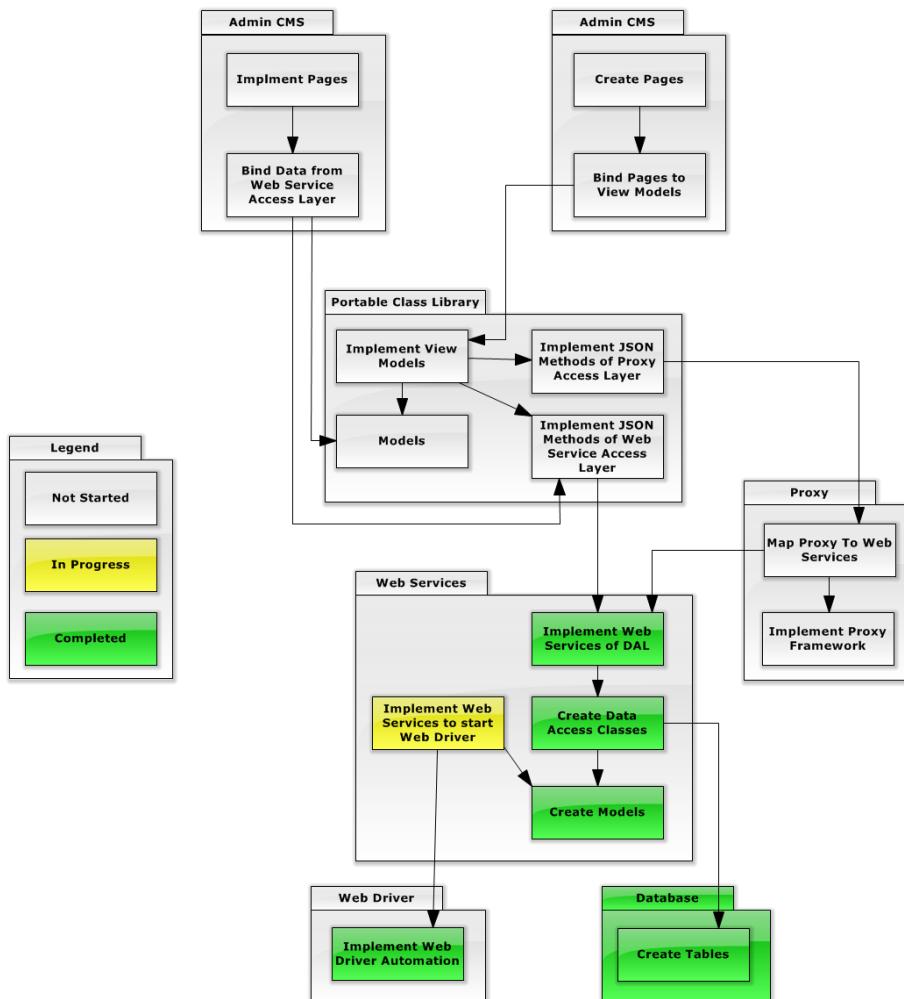


INT-02

Integration ID	INT-02	Test Author	Andy Bottom
Test Name	Web Driver Test	Date Tested	
Modules Involved	Web Services; Web Service Automation Layer;		

Test Summary

Tested the interaction with the web driver automation and being called and activated through the web services.

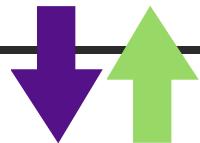
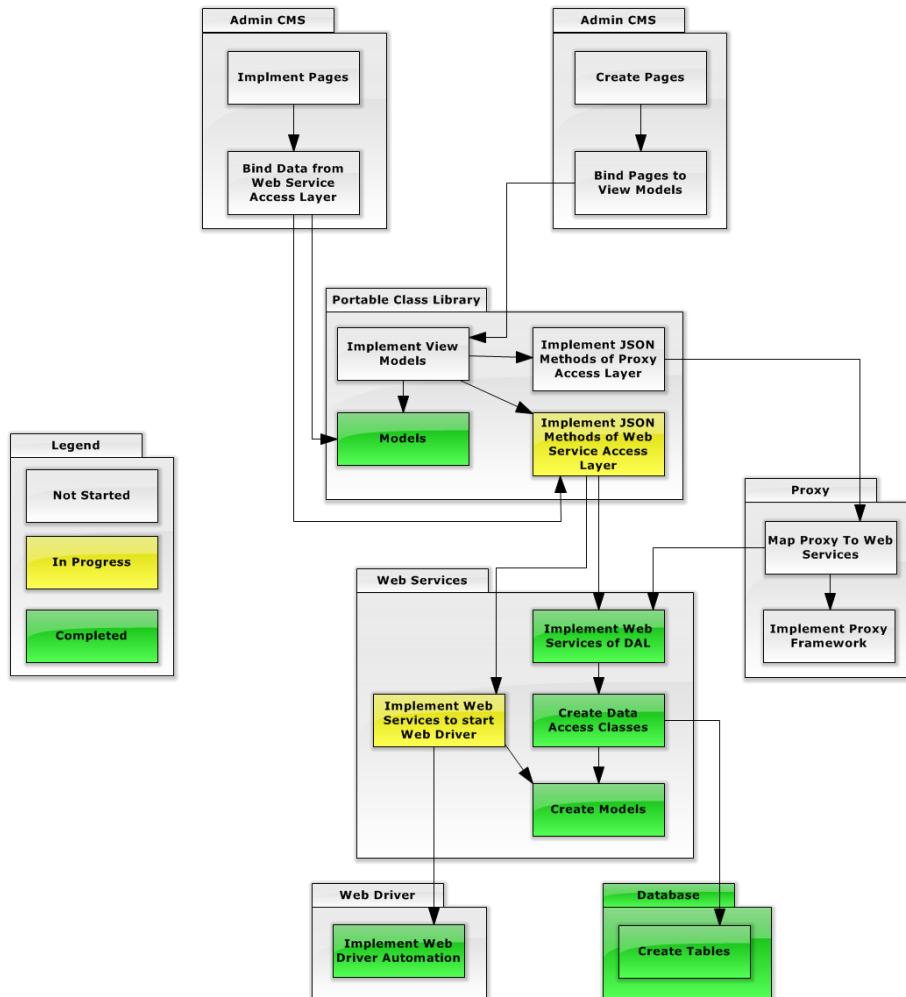


INT-03

Integration ID	INT-03	Test Author	Andy Bottom
Test Name	Automation Test	Date Tested	
Modules Involved	Web Service Automation Access Layer; PCL;		

Test Summary

Tested that the PCL can send calls to the Web Services and receive responses.

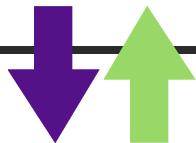
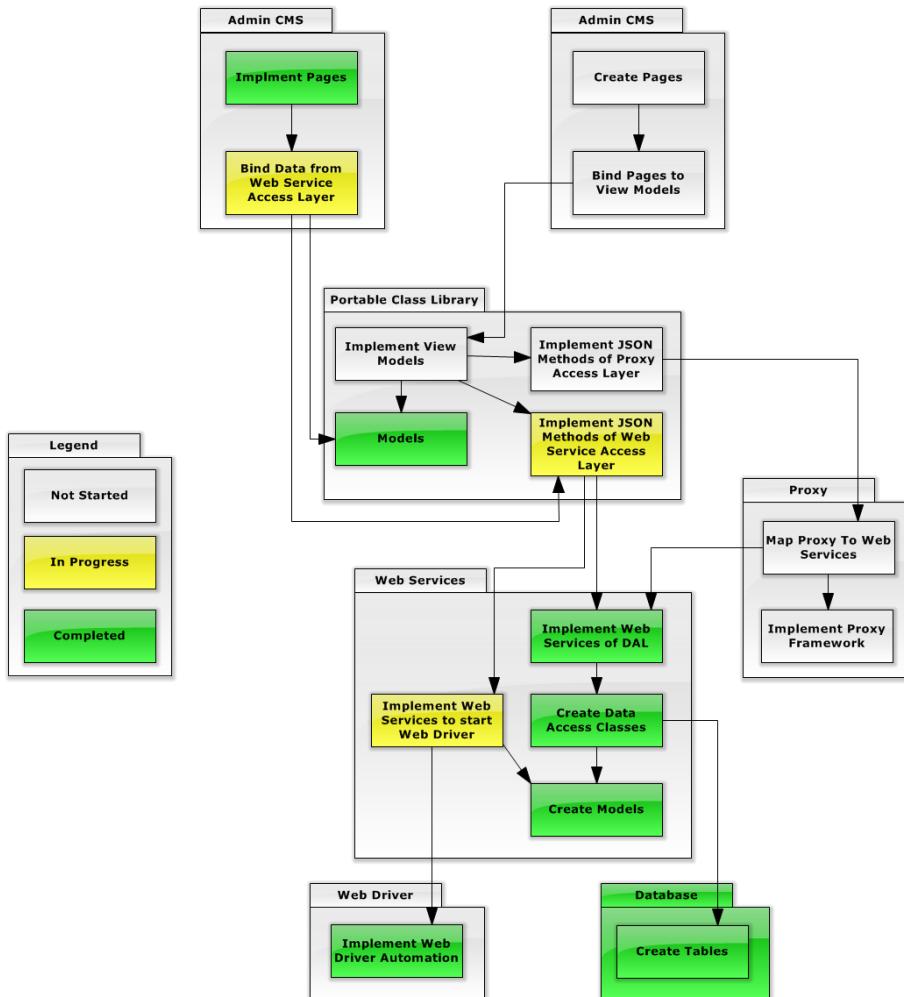


INT-04

Integration ID	INT-04	Test Author	Andy Bottom
Test Name	CMS Binding Test	Date Tested	
Modules Involved	Web Admin CMS; PCL		

Test Summary

Tested that the PCL can get accessed by the Web Admin CMS. This way we can get reuse from the PCL library.

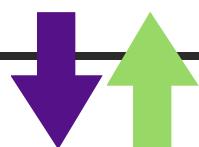
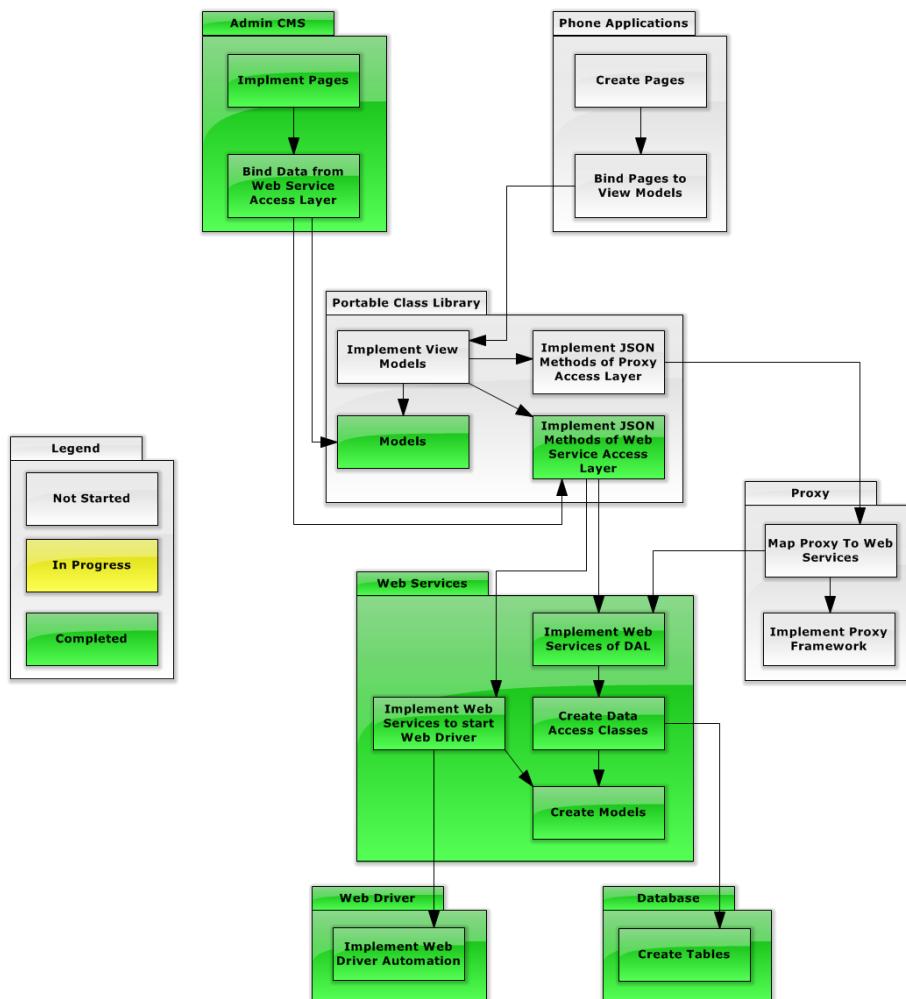


INT-05

Integration ID	INT-05	Test Author	Andy Bottom
Test Name	Complete Web Service Test	Date Tested	
Modules Involved	Admin CMS; PCL; Web Services; Web Driver;		

Test Summary

Test all aspects of the Web Services from all the client applications.

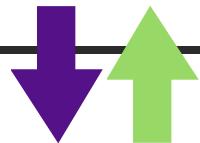
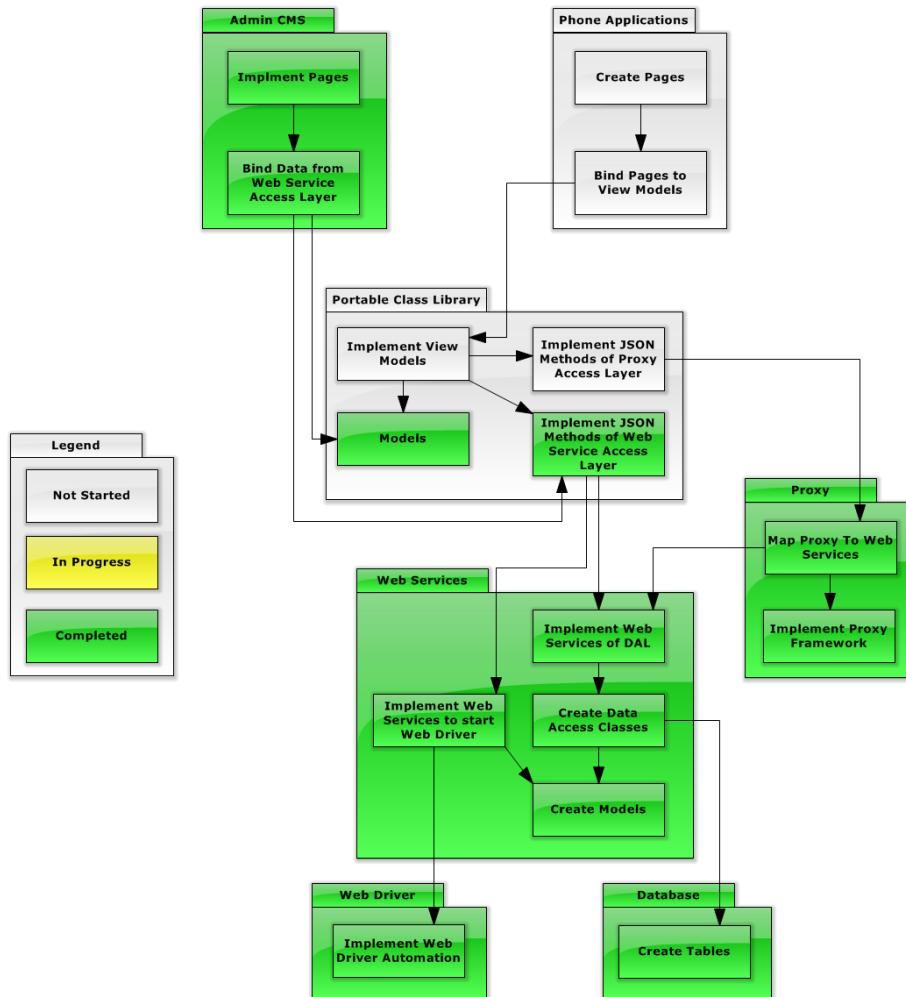


INT-06

Integration ID	INT-06	Test Author	Andy Bottom
Test Name	Proxy Test	Date Tested	
Modules Involved	Proxy; Web Services;		

Test Summary

Tested that reverse proxy actively hits the Web Services the first time and then caches the response so that the second time, it will immediately return the cached response and not hit the web services.

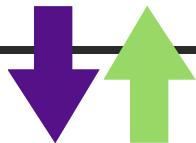
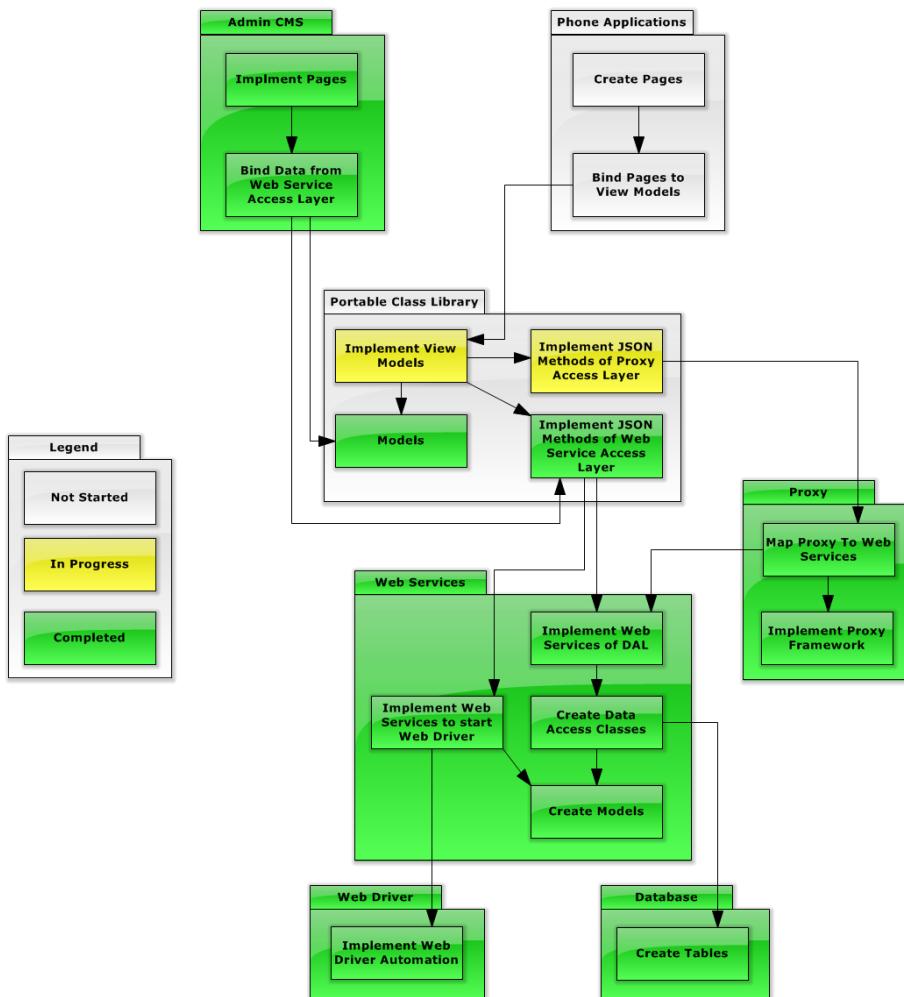


INT-07

Integration ID	INT-07	Test Author	Andy Bottom
Test Name	Initial View-Model Test	Date Tested	
Modules Involved	PCL;		

Test Summary

Tested the bindings of the phone application that it can hook up the UI of the phone to the view-models in the PCL.

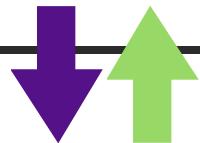
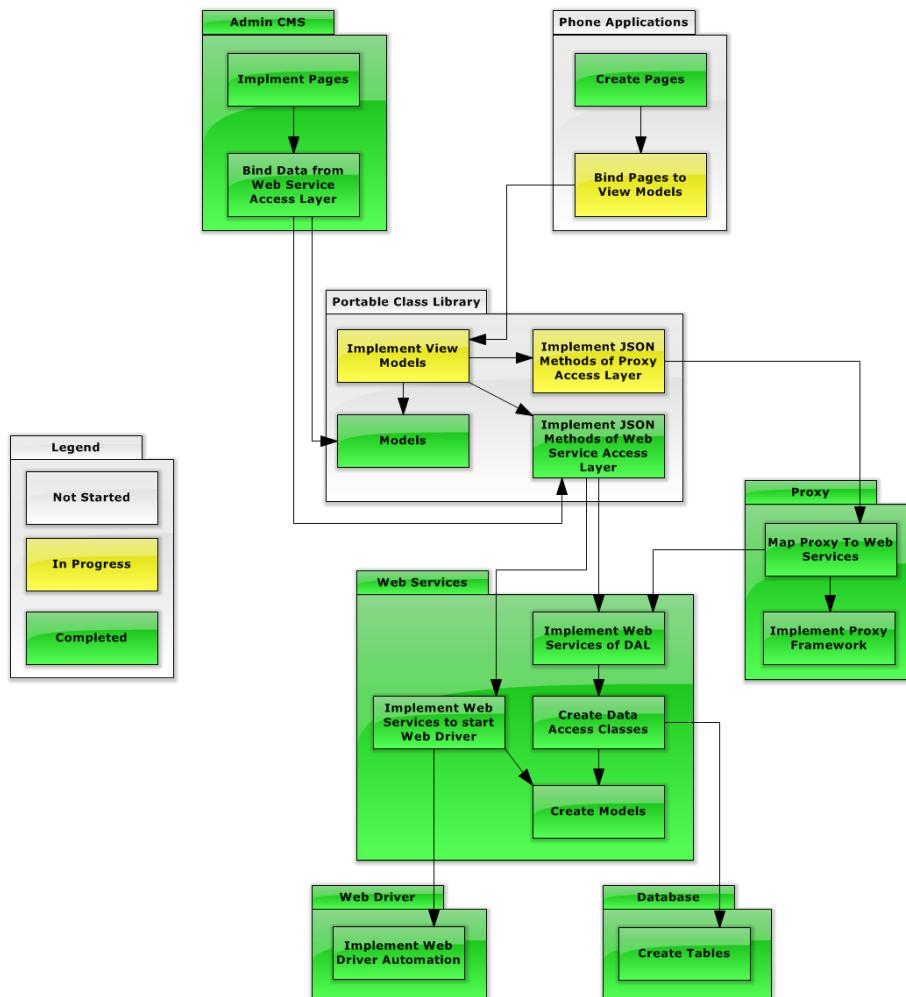


INT-08

Integration ID	INT-08	Test Author	Andy Bottom
Test Name	Phone Binding Test	Date Tested	
Modules Involved	Phone Apps; PCL;		

Test Summary

Complete Test of the Phone Binding with the entire PCL.

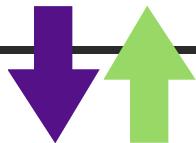
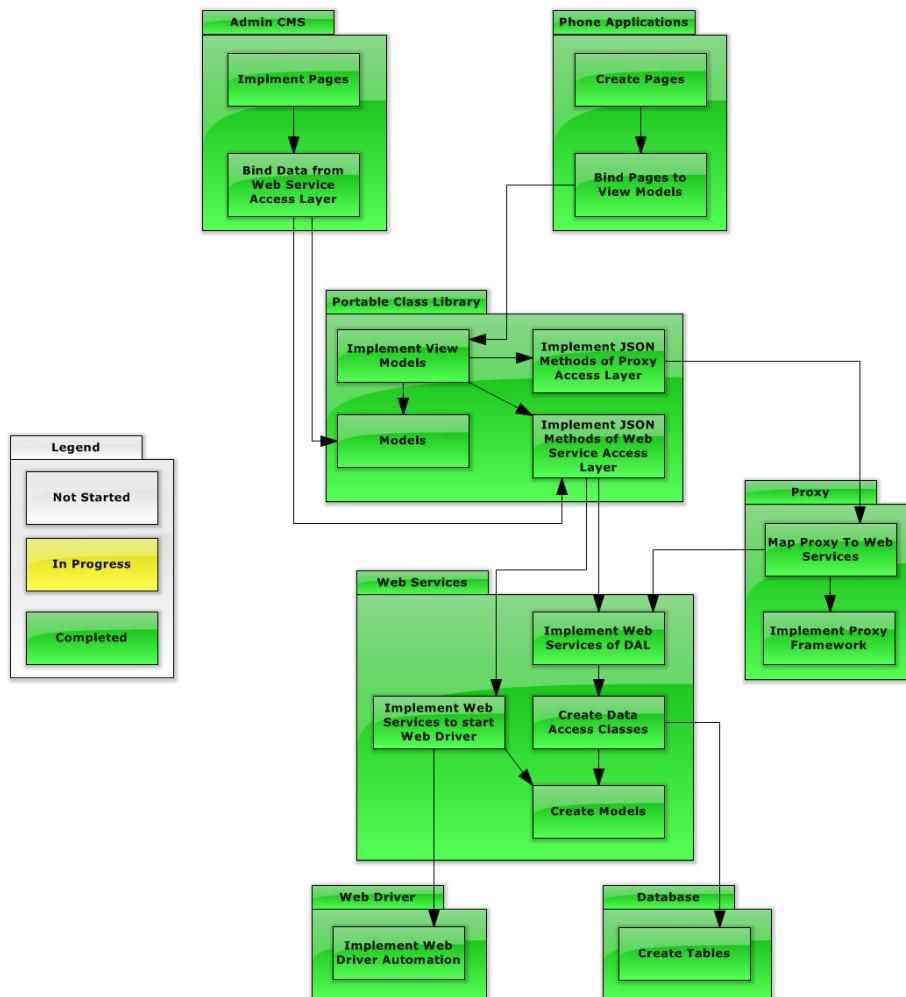


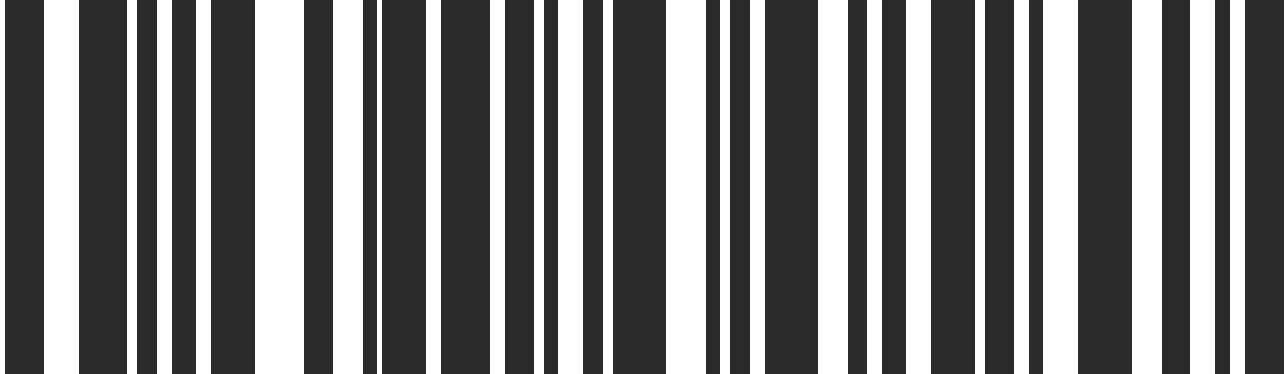
INT-09

Integration ID	INT-09	Test Author	Andy Bottom
Test Name	Complete Integration	Date Tested	
Modules Involved	Phone Application; PCL; Proxy; Web Services;		

Test Summary

Entire system integration test that all the parts work together and data travels properly.





USER EXPERIENCE GUIDE

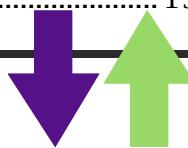
8/23/2013

Graduate Capstone

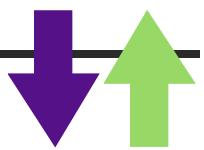


Table of Contents

1 Introduction.....	4
1.1 Intended Audience.....	4
1.1 References.....	4
1.2 Revision History	4
2 Logo	5
2.1 The Logo.....	5
2.1.1 Black and White	5
2.1.2 Color.....	5
2.1 Elements of Logo.....	5
2.1.1 Barcode.....	5
2.1.2 Arrows	5
2.2 Prototypes.....	6
3 Fonts.....	7
3.1 Typefaces	7
3.1.1 Lucidia Console (Sans Serif).....	7
3.1.2 Lucidia Bright (Serif).....	7
4 Colors.....	8
4.1 Purples	8
4.2 Greens	9
4.3 Neutrals	10
5 Design Elements	11
5.1 Windows Phone.....	11
5.1.1 Panorama View	11
5.1.2 Bottom Bar	12
5.2 Android Phone	13
5.2.1 Drawer	13
5.2.2 Action Bar	14
6 Prototypes and Wireframes.....	14
6.1 Phone Application.....	15
6.1.1 Dashboard / Main Menu	15
6.1.2 Register Page	15
6.1.3 Login Page.....	16
6.1.4 Login Welcome.....	16
6.1.5 Search.....	17
6.1.6 Search Results.....	17
6.1.7 Company List	18
6.1.8 Company Page	18
6.1.9 Featured Company Page	19
6.1.10 Survey Page	19



6.1.11 Survey	20
6.1.12 Survey Complete	20



1 Introduction

The purpose of this document is to describe the user interface standards to follow in the development of the front-end aspects of the applications. By doing this, the goal is that there will be a sense of consistency between all the applications.

1.1 Intended Audience

This document is intended for any persons who wish to utilize the Phone Application and gain a better understanding of how the screens work. This document is written for people with any degree of familiarity with the application.

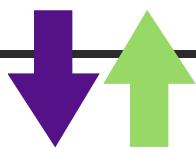
1.1 References

- <http://www.globway.eu/files/100511%203API%20User%20Experience%20Guideline.pdf>
- http://www.ixda.org/sites/default/files/UX_Kit_Aug09.pdf
- <http://creativetest2008.files.wordpress.com/2008/05/blogs-v121doc.pdf>
- <http://www.mass.gov/eohhs/docs/eohhs/eohhs-user-experience-and-style-guide-v3.pdf>
- <http://www.wsdot.wa.gov/eesc/bridge/software/files/user%20interface%20specific%20action.pdf>
- http://web.fe.up.pt/~jlopes/lib/exe/fetch.php/teach/lbaw/lectures/ibm_uia.pdf

1.2 Revision History

Managing the change history of this document will occur in this table.

Name	Date	Reason For Change	Version
Andy Bottom	07/29/2013	Added the initial categories for the document. Also added references.	0.1



2 Logo

The logo is a very important part of a product. By having the logo, we begin to create the entire look for the product. From here we add colors and define elements that can identify with the application.

2.1 The Logo

This is the logo that will be used on for the phone application.

2.1.1 Black and White



2.1.2 Color



2.1 Elements of Logo

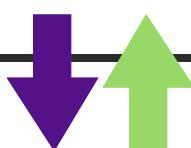
The following section contains the design elements involved in the logo.

2.1.1 Barcode

The barcode is very symbolic of the receipts. The barcode can also be an element reused throughout the app

2.1.2 Arrows

I wanted to use arrows to represent the transactional relation between the Receipt and the Rewards. The arrows are also elements used throughout the app.



2.2 Prototypes

The following section contains the prototypes of the logos.



3 Fonts

The following section contains all the information about the fonts used in the project.

3.1 Typefaces

The Typefaces used throughout the applications and the documentation is very important. The text needs to be consistent throughout the app. Below is a list of both Serif and San Serif font faces. Typically, for Headings, the Sans Serif will be used. Otherwise for body text, the serif font face will work best.

3.1.1 Lucidia Console (Sans Serif)

We decided to go with the Lucidia Sans because the typeface on receipts is Lucidia Console. This is the typeface used in the Receipt Rewards Logo.

3.1.2 Lucidia Bright (Serif)

For a serif font, we will be going with the sister font Lucidia Bright for the serif option.



4 Colors

This section contains all the colors used in the project. The colors are a schema. The first colors in each row are the main ones. There others are simply a shade to help show of the other colors and have a contrast.

4.1 Purples

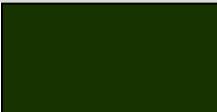
The color purple was chosen because it has a royal connotation to it.

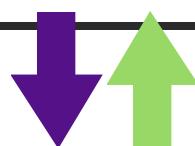
Color	RGB	HSB	Hexadecimal
	R: 85 G: 17 B: 136	H: 273 S: 87 B: 53	#551188
	R: 99 G: 63 B: 127	H: 272 S: 50 B: 49	#633F7F
	R: 189 G: 121 B: 242	H: 272 S: 49 B: 94	#BD79F2



4.2 Greens

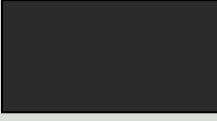
The color green was chose to represent the rewards, due to the money connotation green has.

Color	RGB	HSB	Hexadecimal
	R: 152 G: 216 B: 102	H: 93 S: 52 B: 84	#98D866
	R: 22 G: 51 B: 0	H: 93 S: 100 B: 20	#163300
	R: 67 G: 76 B: 61	H: 94 S: 19 B: 29	#434C3D
	R: 44 G: 102 B: 0	H: 93 S: 100 B: 40	#2C6600
	R: 73 G: 168 B: 0	H: 93 S: 100 B: 65	#49A800



4.3 Neutrals

The few neutral colors help support and enhance the colors that are used.

Color	RGB	HSB	Hexadecimal
	R: 43 G: 43 B: 43	H: 93 S: 0 B: 16	#2B2B2B
	R: 218 G: 221 B: 215	H: 88 S: 22 B: 86	#DADDD7
	R: 17 G: 16 B: 17	H: 299 S: 5 B: 6	#111011



5 Design Elements

This section is used to explain and describe UI patterns that will be used in the phone applications. It is important for the applications to have a UI that is geared specifically for that OS, so that the experience feels natural and comfortable to the user.

5.1 Windows Phone

The Windows Phone OS has a very distinctive user experience.. To keep up to date and utilize these elements, a list below will describe in more detail what to look for and use in the front-end of the Windows Phone Application.

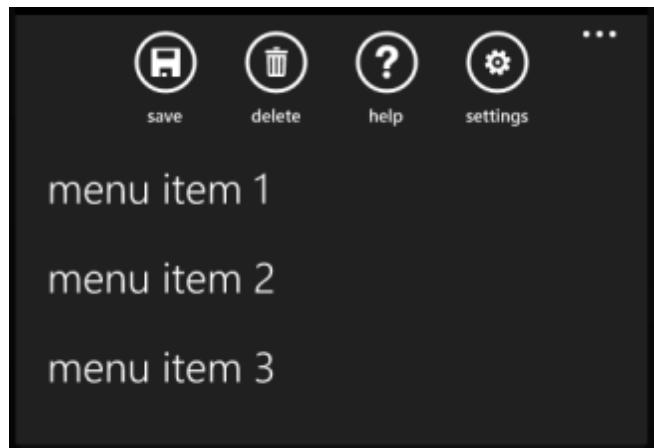
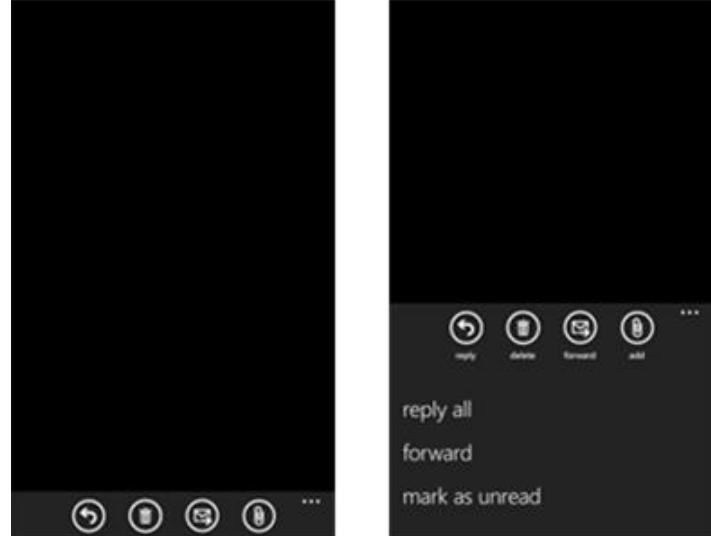
5.1.1 Panorama View

The first main UX element that will be utilized is the Panorama View. The panorama view is unique as it can be considered a very long page of which the phone can slide to see more content. The pages work by adding Panorama Items to the Panorama view. These items are the actually things that display to the user. The function of the user is in contain and hold the sever Panorama Items to create a very distinctive view.



5.1.2 Bottom Bar

The bottom bar is another Windows Phone element. It is essentially a menu system that allows the user to access additional functionality on the page, without having to trade off valuable visible space. By have users to easily perform actions in this manner, it allows to create an interface that is more approachable, organized and intuitive to use.

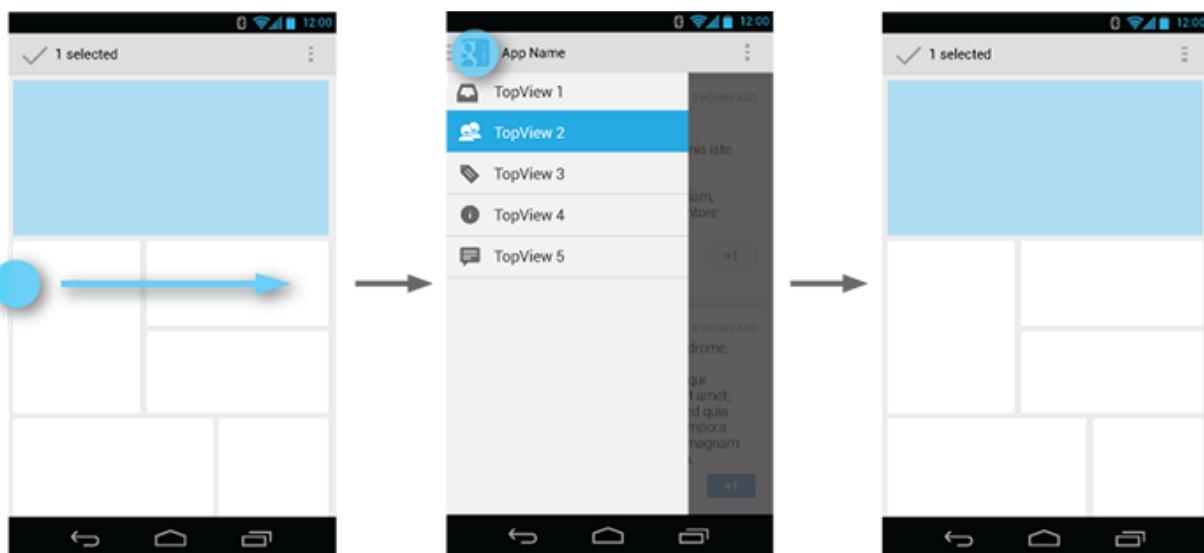


5.2 Android Phone

The Android Phone has a more common user experience. The OS is open sourced and has a unique style in that it adapts UI elements into the API as patterns begin to get widely used in apps. A good example of this is the Drawer Element.

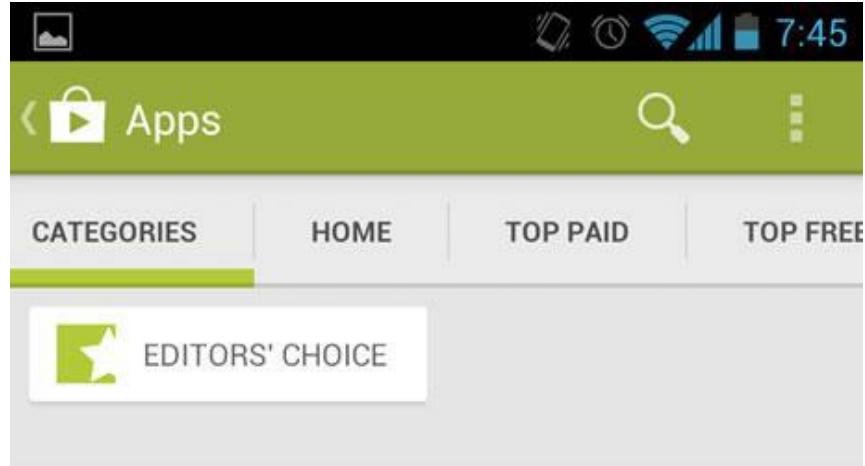
5.2.1 Drawer

The drawer is the newest official way to provide easy to use navigation for the user. The drawer pattern has been widely used in adopted by apps already, which is why Android has now officially added the Drawer element to its API due to its wide success. The drawer is essentially a hidden vertical menu that the user can access by sliding from the side. In the drawer is the navigation for the application. By being able to navigate in this way, the user experience can save valuable space by having this core functionality hidden off screen, but able to be used at any time.



5.2.2 Action Bar

The action bar is another very popular UI pattern that helps in navigation. By having this bar it allows the user to understand exactly where they are in the current flow of the application. In addition, it assists in navigation in views where pages are next to each other in a page. The user can slide to move to the next “page” or click the link in the action bar to navigate.



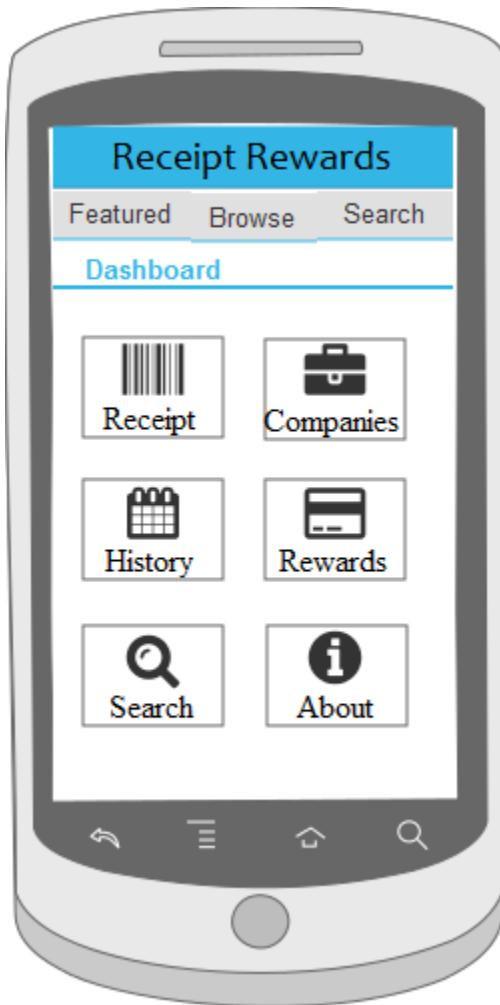
6 Prototypes and Wireframes

A wireframe and mock of the interface of the phone application during the initial design phase of the application. By doing this, it could be planned how the user would interface with the system and utilize the functionality. This allowed use cases to be created specifically aimed at the phone user.



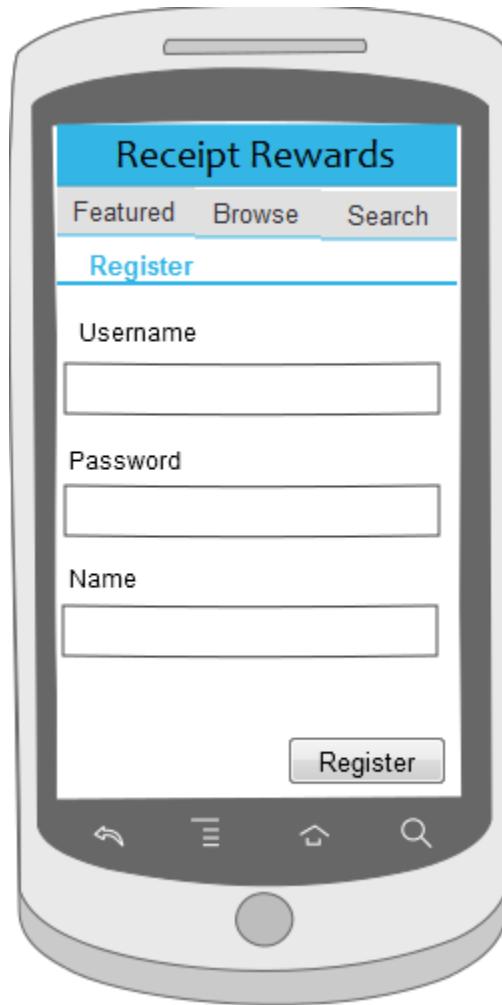
6.1 Phone Application

6.1.1 Dashboard / Main Menu

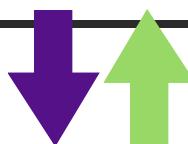


This is the dashboard or main menu page. From here, the user will be able to get to all the functionality of the system.

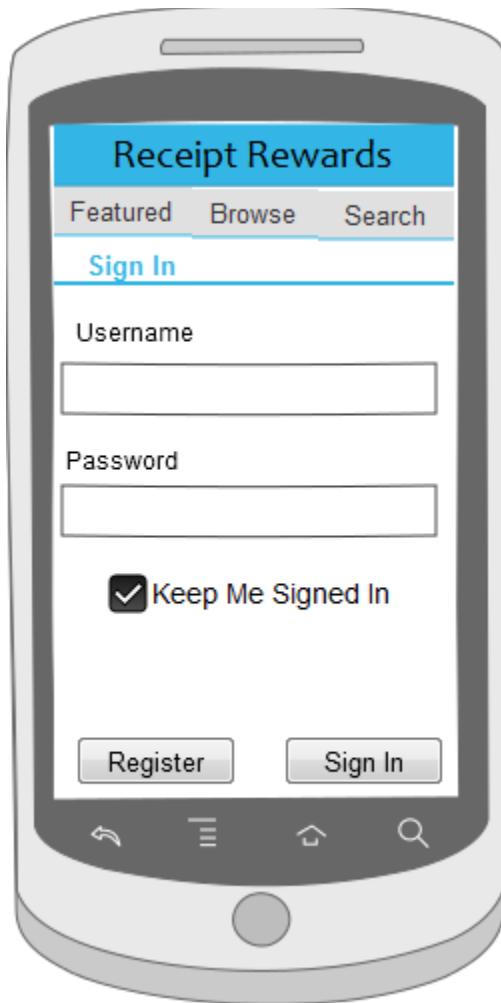
6.1.2 Register Page



This is a register page so that an anonymous user can register and create a user account for them.

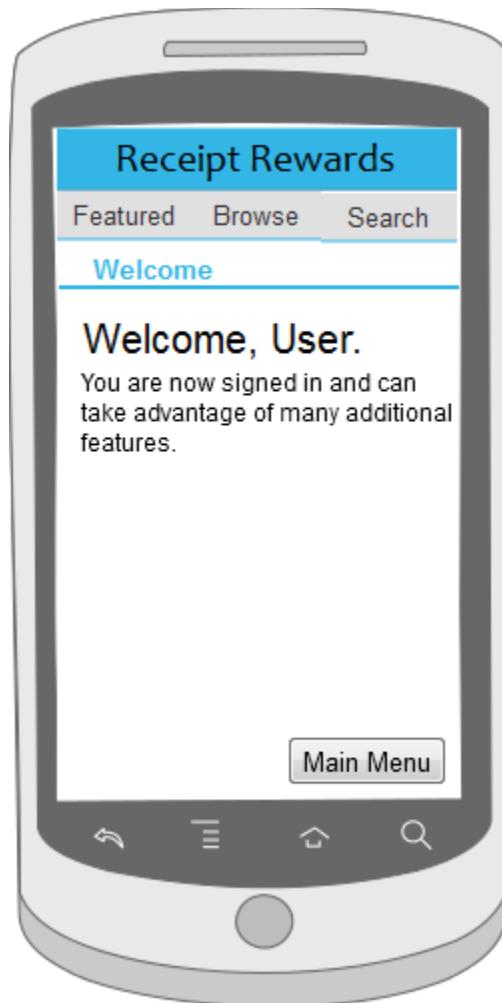


6.1.3 Login Page



The login page provides the functionality for the user to be able to login to their user account.

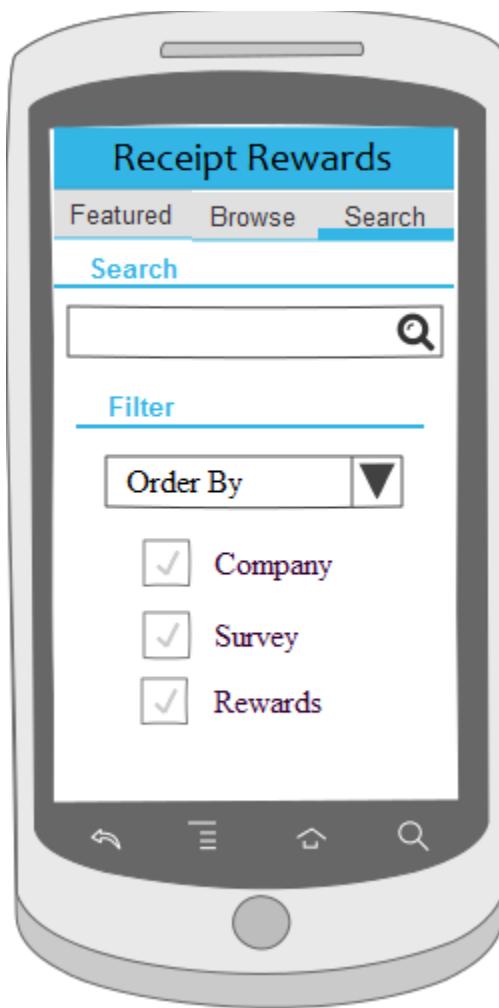
6.1.4 Login Welcome



This is the welcome page for the user.



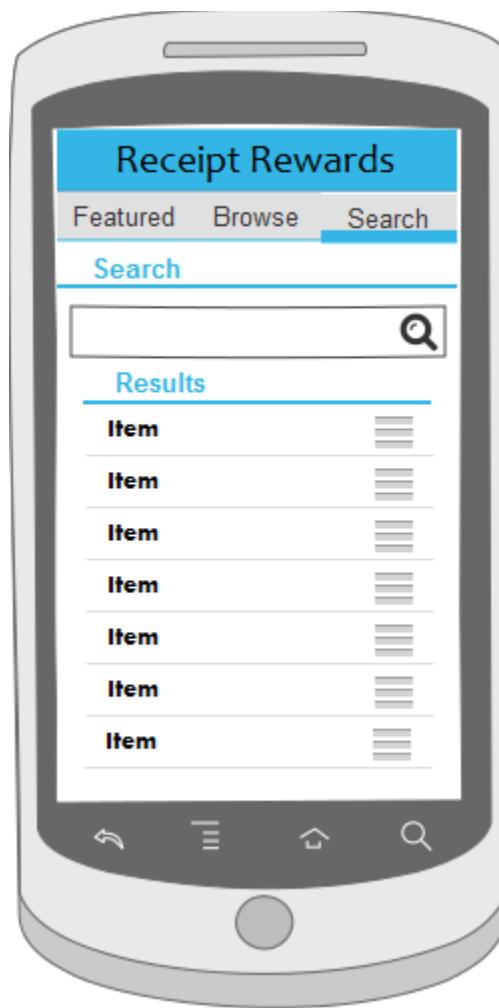
6.1.5 Search



The search gives the user a search box and advanced options to be able to search for companies and surveys.

Upon searching, it will display the search result page.

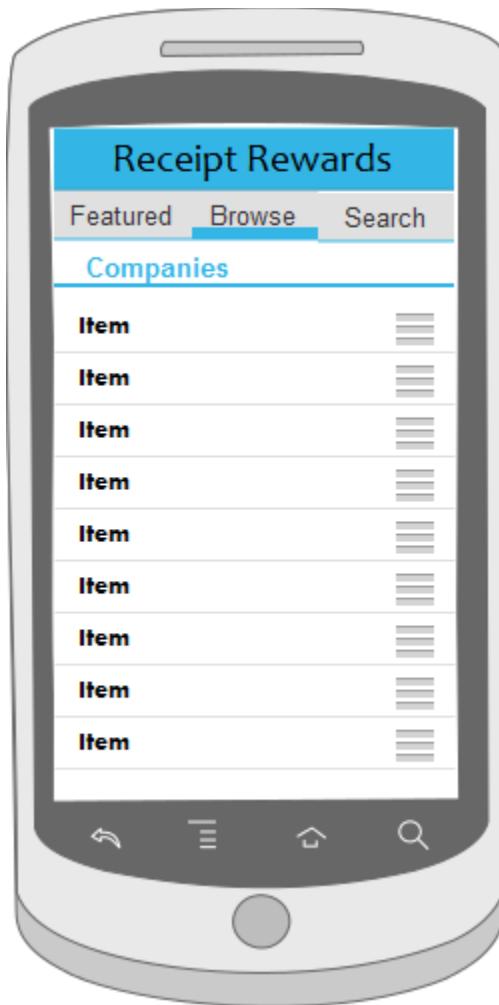
6.1.6 Search Results



This is the search results page. The result will be of companies and surveys. They can select an item and go to either the Company Page or the Survey Page.



6.1.7 Company List



The company list page is used to display a list of all the companies to the user.

From here, a user can select a company to view which will display the company page to the user.

6.1.8 Company Page

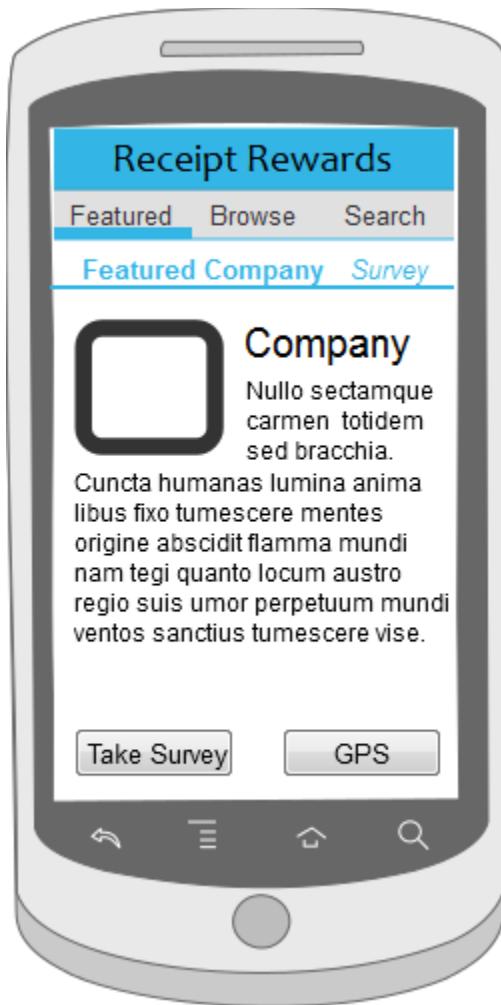


The company page is the page which displays all the information about a company to the user.

From this page, a user can also navigate the survey corresponding to that company.

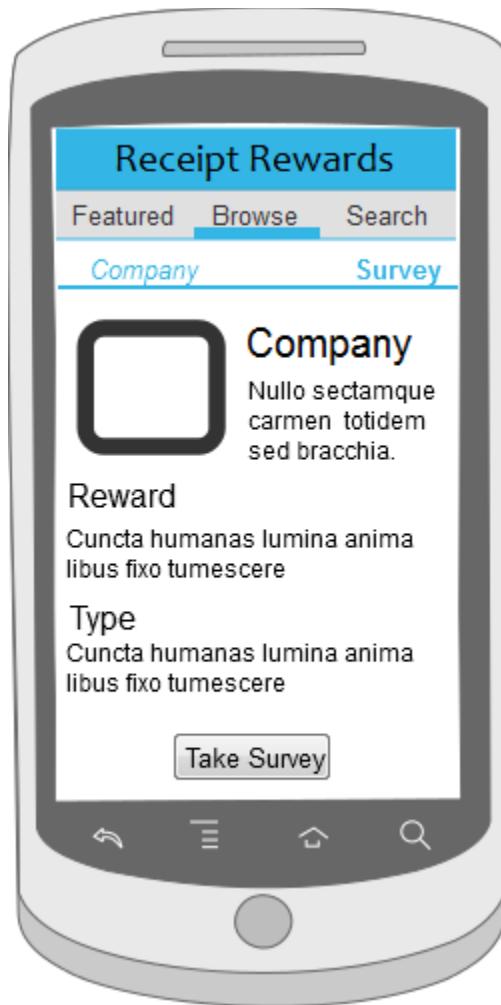


6.1.9 Featured Company Page



The featured company page is a place in the application where a company can pay to be listed in a special section. Kind of like a advertising spot for companies.

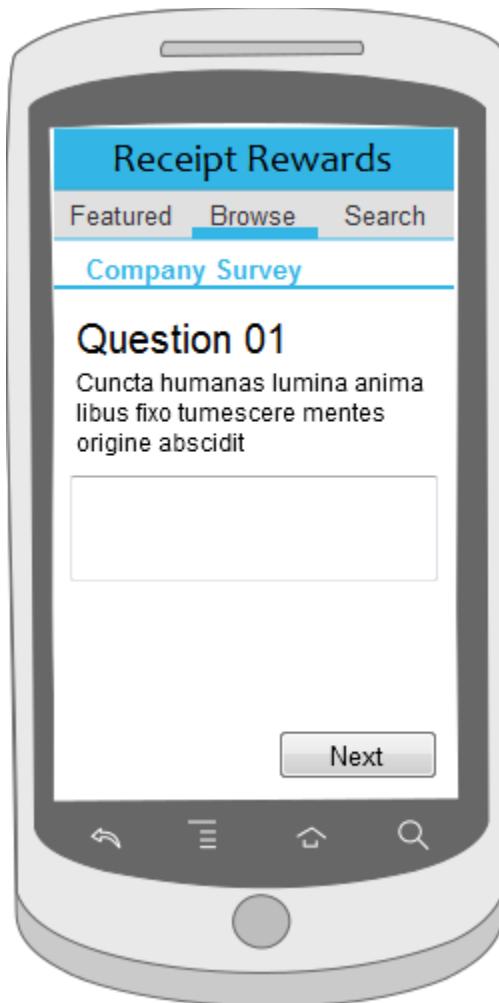
6.1.10 Survey Page



Survey page provides information regarding the survey and what the reward will be. From here, they can click take survey to move on to filling out the survey.

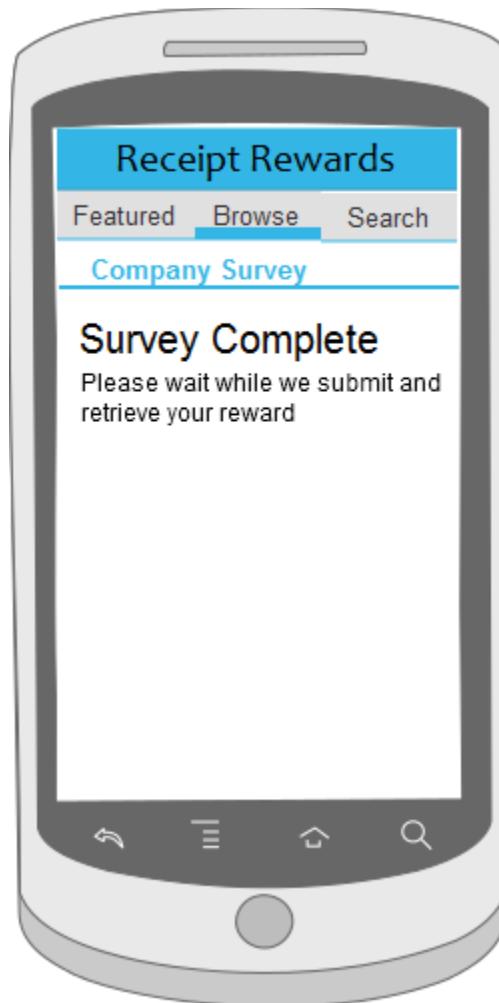


6.1.11 Survey



This is the page where they are actually taking the survey and the first question is displayed. They go through these screens until they finish.

6.1.12 Survey Complete



Once they finish taking the survey they will see the complete page. It is also on this page where they will receive information about how to get their reward.





DEVELOPER GUIDE

8/23/2013

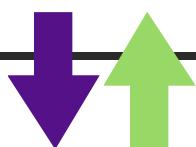
Graduate Capstone

RECEIPT

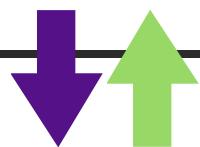
REWARDS

Table of Contents

1	Introduction.....	4
1.1	Intended Audience.....	4
1.2	References.....	4
1.3	Revision History	4
2	Before You Start.....	5
2.1	Visual Studios 2012	5
2.1.1	PCLs for Android and iOS	6
2.1.2	Xamarin Plug-In	8
2.2	NetBeans IDE	9
2.2.1	Glassfish with SQL Server Support	9
2.3	Virtual Machines and Emulators.....	11
3	PCL.....	12
3.1	References Package.....	12
3.2	Aliases Package.....	13
3.3	Business Layer Package.....	13
3.4	Models Package.....	14
3.5	Service Access Layer Package	14
3.6	View Models.....	14
4	Windows Phone.....	15
4.1	Properties Folder	15
4.2	References Folder.....	15
4.3	Panorama Model Package	16
4.4	Resources Folder	16
4.5	Pages	16
5	Android.....	17
5.1	Properties	17
5.2	References.....	17
5.3	Fragment Models Package	18
5.4	Resources Package	18
5.5	UI Models Package.....	18
5.6	Pages	19
6	Web Admin CMS	20
6.1	App Code.....	20
6.2	Bin.....	21
6.3	Includes	21
6.4	Scripts	21
6.5	Site Files	21
6.6	Styles	22
6.7	Pages	22



7 Web Services	23
7.1 Alias Models Package	23
7.2 Automation Package.....	24
7.3 Common Package.....	24
7.4 Controller Package.....	24
7.5 Models Package.....	25
7.6 Libraries.....	25



1 Introduction

The purpose of this document is to provide a guide for developers as how to get started with the project. In addition, it aims to give a detailed explanation of the structure and organization of the project structure and files.

1.1 Intended Audience

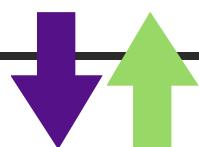
This document is intended mainly for developers who will be writing code for the project. The goal is for those individuals to learn about the organizations, conventions and purposes of each file and object and what they do in the project.

1.2 References

- <http://developer-support-handbook.appspot.com/documentation.html>

1.3 Revision History

Name	Date	Reason For Change	Version
Andy Bottom	06/24/2013	Created the formatting of the document and added references	0.1
Andy Bottom	08/20/2013	Completed the developer guide and all the content.	1.0



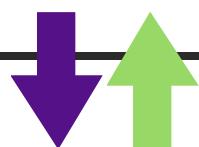
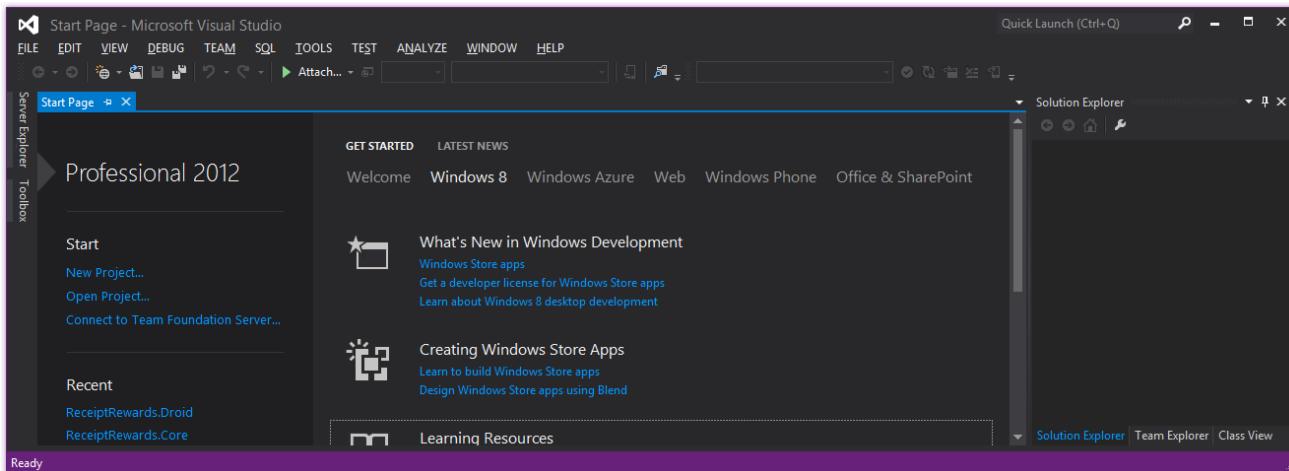
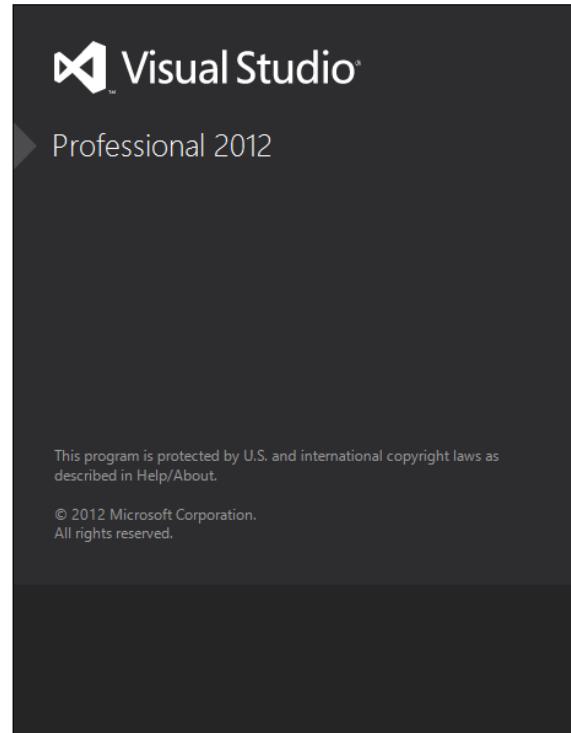
2 Before You Start

Before you can start, there are several steps that need to be done, such as configuring your environment

All software dependencies can be found in the [Software Requirement Specification](#), but the following section explains the tricky setup steps that were needed in order to prepare the environment and have things running smoothly.

2.1 Visual Studios 2012

You will need to be running Visual Studios 2012 as it is designed to function properly with phone development and Xamarin plugins.



2.1.1 PCLs for Android and iOS

By default, PCL libraries only work for the Windows Platforms (Windows Phone, Windows Store, X-Box.) However, the project relies on the ability to put all the back-end logic into the PCL to be used by all Phone OS Platforms. Thus additional configuration must be done so that VS can recognize the PCL for all platforms.

To configure Visual Studios 2012 to create and use PCLs for Android and iOS, first navigate to the following directory:

C:\Program Files (x86)\Reference
Assemblies\Microsoft\Framework\.NETPortable\v4.0\Profile\Profile104\SupportedFrameworks

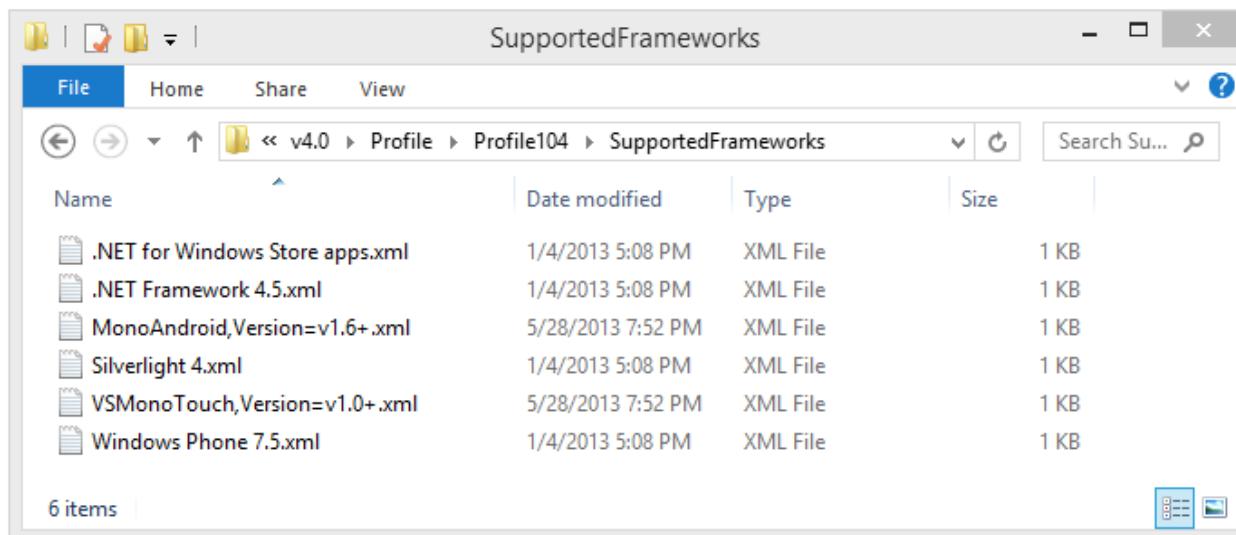
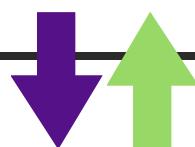
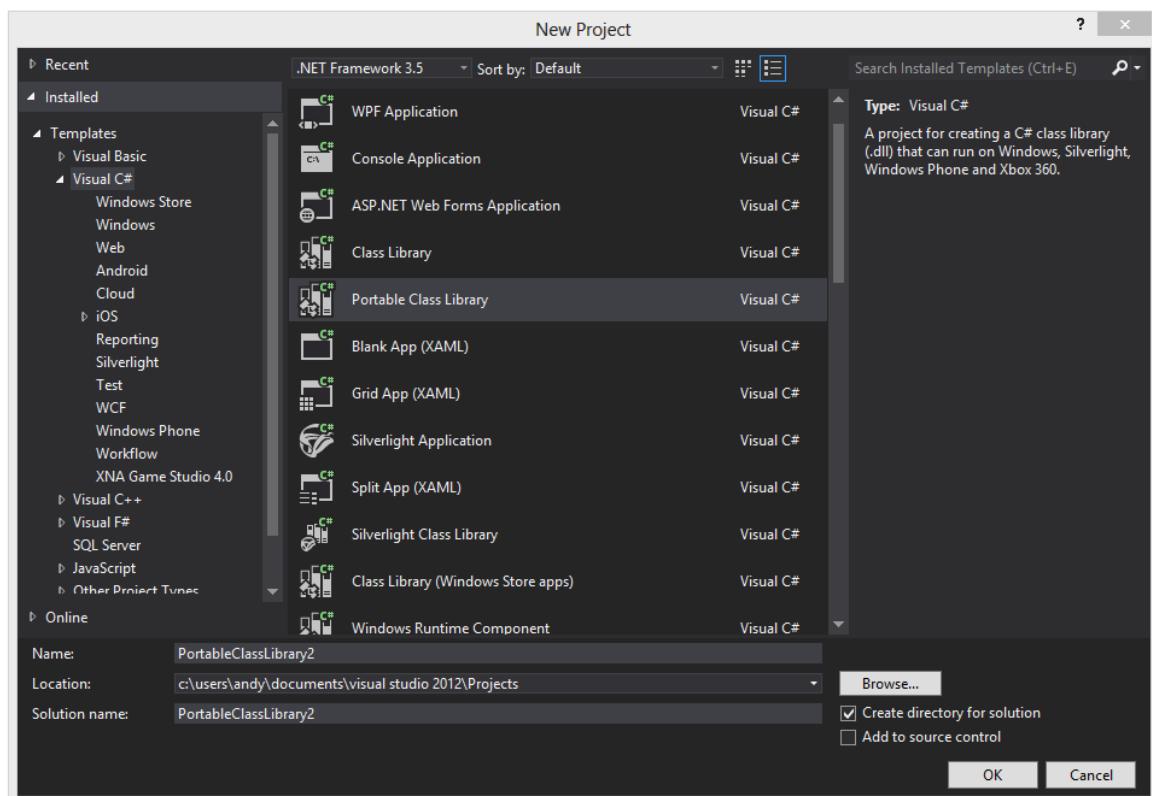


figure 2.1.1.1-1

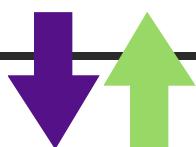
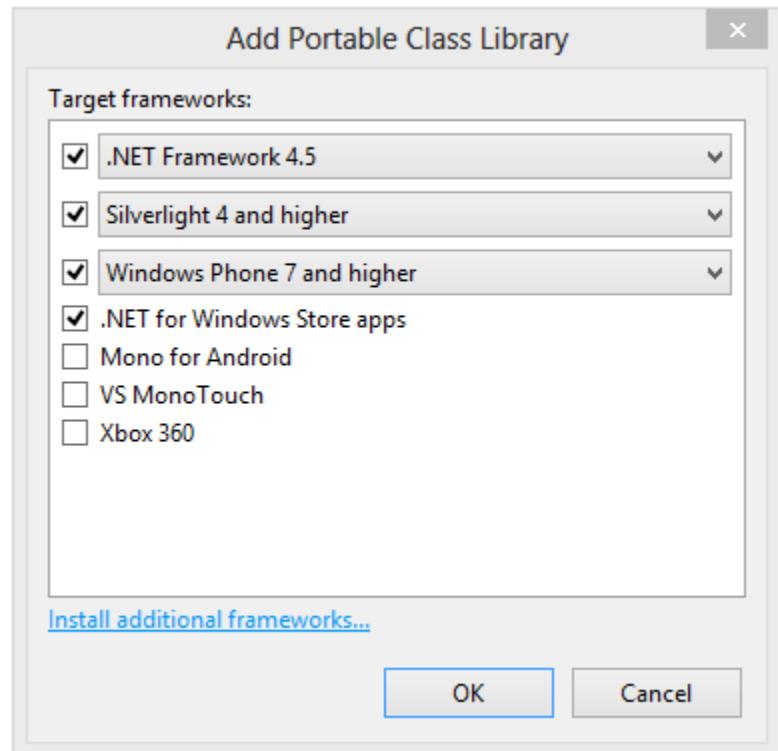
In the developer guide resources, (that is provide in the code base,) you find files that will need to be added to this directory. Copy the files into this folder so that your directory should appear similar to **figure 2.1.1.1-1**:

- .NET for Windows Store apps.xml
- .NET Framework 4.5.xml
- MonoAndroid,Version=1.6_.xml
- Silverlight 4.xml
- VSMonoTouch,Version=v1.0+.xml
- Windows Phone 7.5.xml



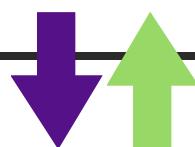
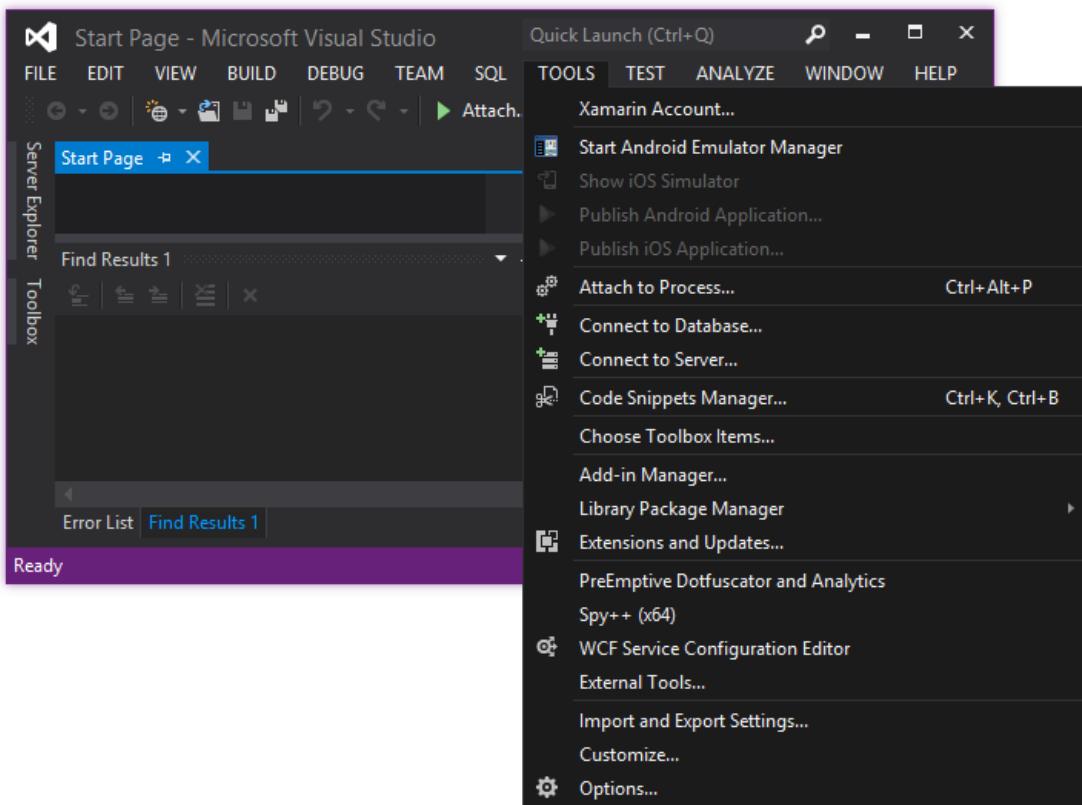
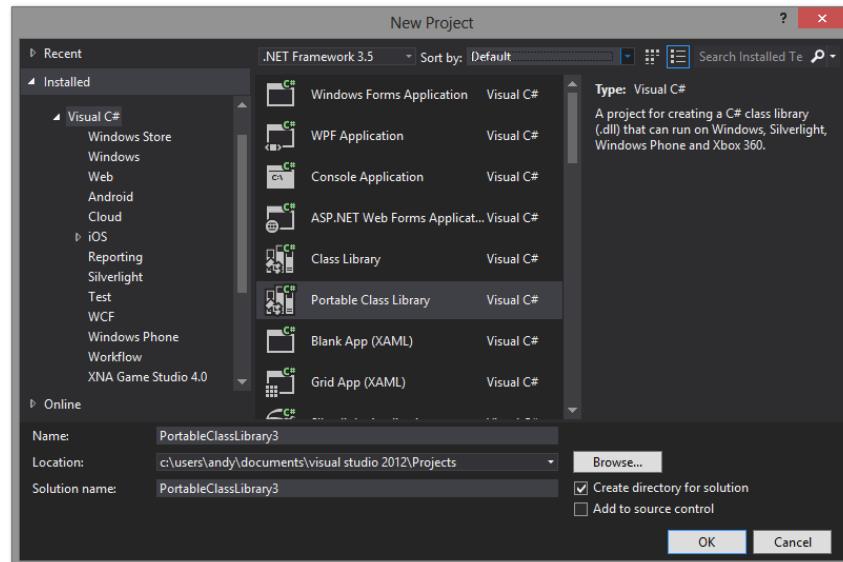


Once these files are in the folder system, open up Visual Studios 2012. When you create a new PCL library, a dialog will appear asking for what versions you want the PCL to target. If you are successful, all the platforms including Windows Phone, Android and iOS should appear.



2.1.2 Xamarin Plug-In

The project utilizes the Xamarin Platform, so the plugins must be installed. Below is a screenshot as to how the Xamarin Plugin appears in Visual Studios 2012.





2.2 NetBeans IDE

The NetBeans IDE is used to create the Web Services. Other Java IDEs could be used, but I use this one, thus instructions about how to set it up will be given.

2.2.1 Glassfish with SQL Server Support

Since the Database is on a Microsoft SQL Server, the glassfish server must contain the jar resource so that it can successfully make this connection. To configure this, follow the following steps.

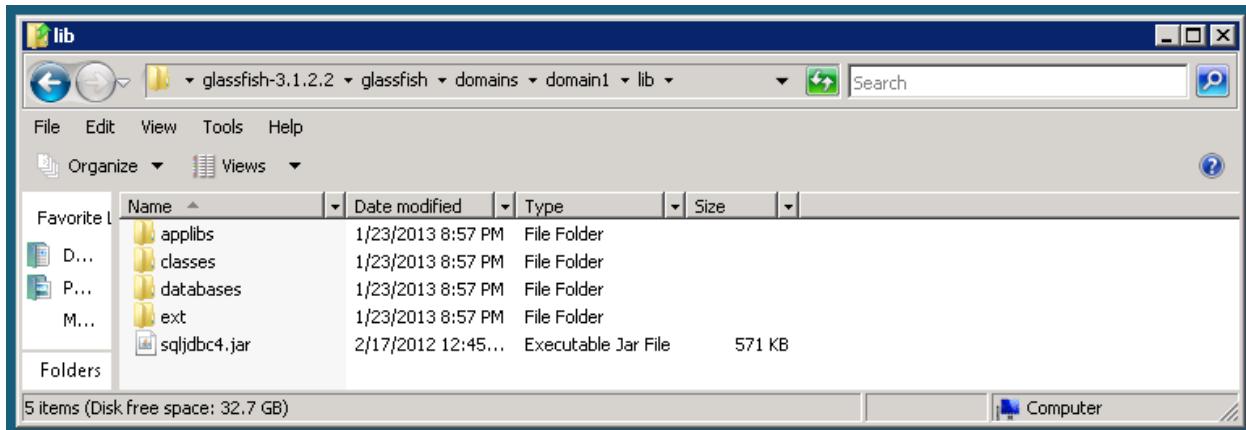
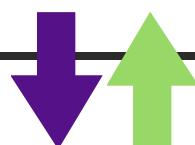


figure 2.2.1-1

First, find the location of the glassfish folder. Then, navigate to the relative URI, `glassfish/domains/domainX/lib/` (where x is the domain number, default is 1). Once located in the libs folder, put the file `sqljdbc4.jar` into this folder, (the file can be found in the Developer Guide Resources Help Package.) The structure should be similar as seen in **figure 2.2.1-1**.



This will then fix allow for the server to have the jdbc connection loaded so that the web services can utilize the jar for the connection. To verify that the jar was properly loaded, navigate to the admin console windows of the glassfish server. Next, navigate to the JDBC Resources sections. In here will be the list of connection resource. If you create a new one, you will see several dropdown boxes. In the Driver field, find the option that says Microsoft SQL Server. If this field is present, then the JDBC Driver was successfully loaded.

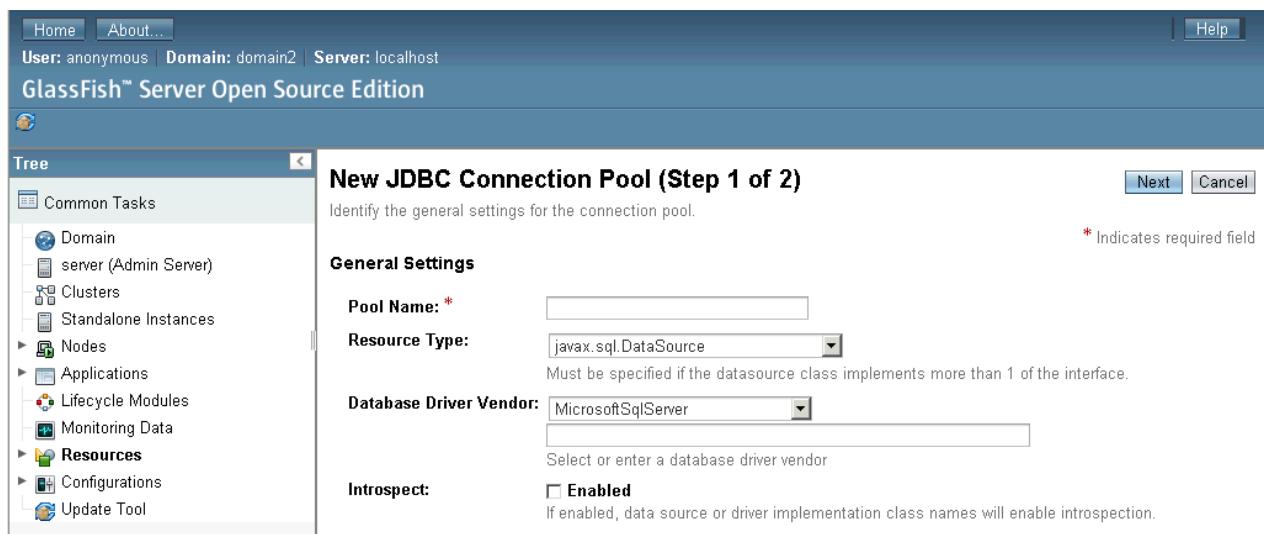
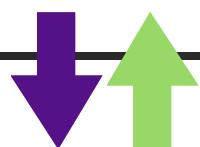


figure 2.2.1-2



2.3 Virtual Machines and Emulators

During the testing of the phone app, you will be using the Phone Emulators. However, you may run into a problem where you receive an error message, as seen in **figure 2.3-1**, and the Emulators fail to start.

The cause of this is that the PC is not set up to have multiple operating systems co-running on the system. This is a configuration found in the BIOS of the computer.

To fix this, restart computer. Prior to the OS starting, initialize the BIOS configuration screen of the PC. From here navigate to advanced PC settings. In this area, there are several properties. The following properties must be enabled to allow for virtual machines and emulators to be ran.

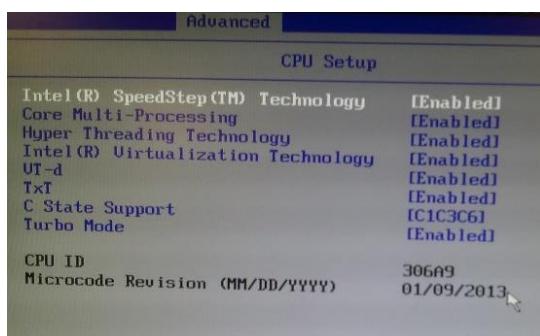


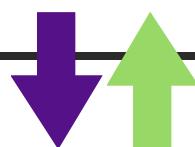
figure 2.3-2

- Hyper Threading Technology - Enabled
- Intel ® Virtualization Technology - Enabled
- VT-d - Enabled
- TxT - Enabled

Once these configurations are done, save the changes and restart the computer. Next time that you attempt to use a virtual machine or emulators, you will find that the error is gone and that they should be working normally.



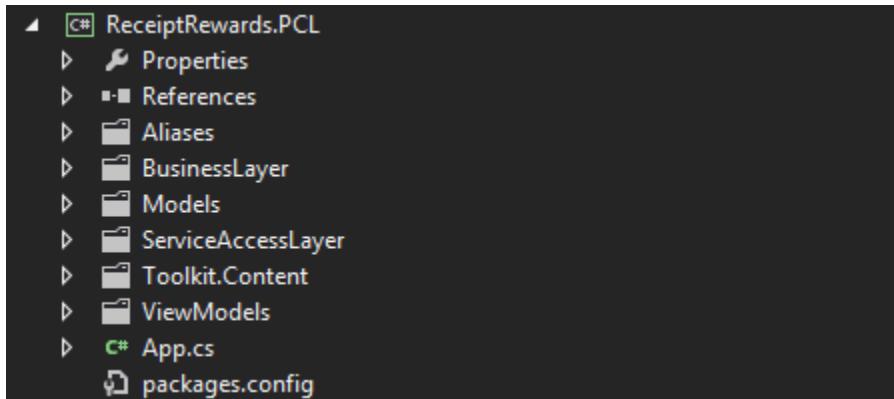
figure 2.3-1



3 PCL

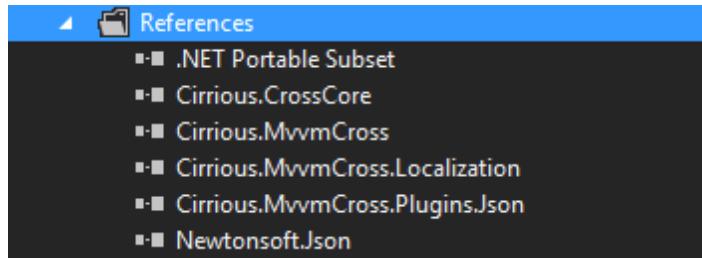
The portable class library is the common functionality needed for all version of the phone application. Think of the PCL as containing everything needed for the phone app, just without the user interface and handlers. The phone applications will utilize this special library to make the app work and function.

The PCL contains certain packages which will be described in greater detail below.

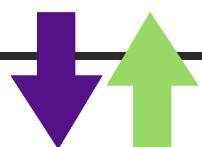


3.1 References Package

The references folder contains all the libraries that the PCL code is dependent on. As you can see in

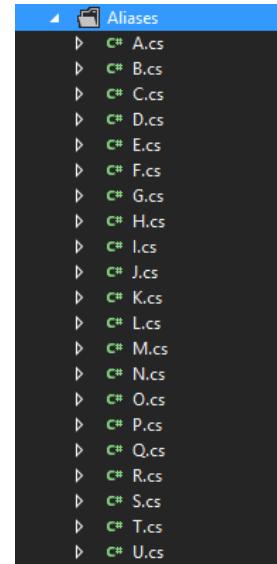


the structure below, the MCCM Cross Platform is used mainly for the ability to perform the Web Services JSON Requests, and the Newtonsoft is used to easily parse the JSON into objects.



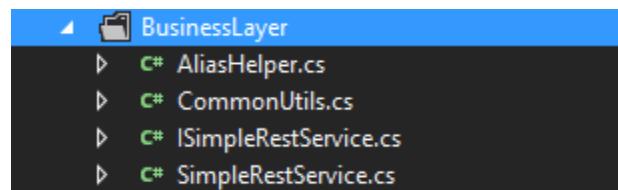
3.2 Aliases Package

The alias classes are the objects used to receive and transfer the standard objects to and from the Restful Services via an anonymous structure. This allows the request to be substantially smaller, thus improving the performance.

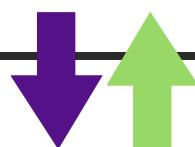


3.3 Business Layer Package

The business layer of the PCL contains static classes of methods that are commonly used throughout the rest of the PCL classes, and even for other code bases to use. By having this package, instills an emphasis on code reuse.

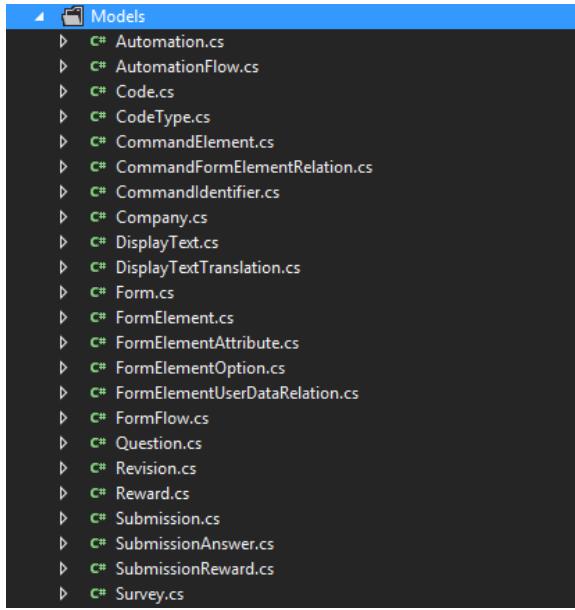


- *Alias Helper*: Contains the logic to convert all the alias objects to and from the standard objects.
- *Simple Rest Service*: Contains the logic to perform an asynchronous call to the Web Service Layer. The requests are based on JSON.
- *Common Utils*: Contains very miscellaneous functions that were used. Simply a code reuse of most common functions.



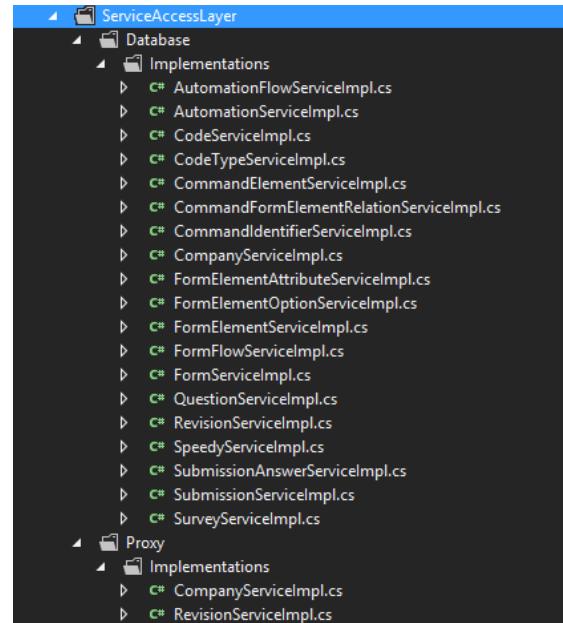
3.4 Models Package

The models contain all the objects used in the system for the C# code base.



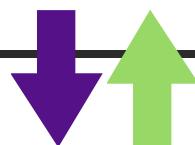
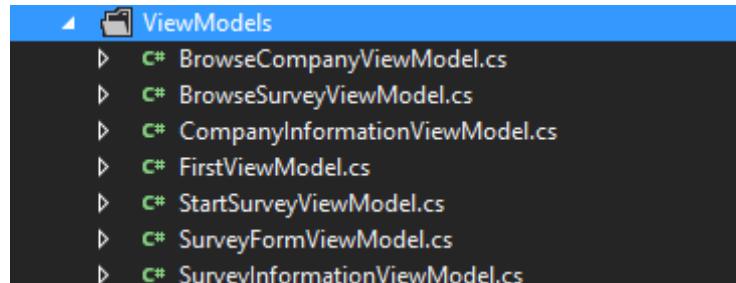
3.5 Service Access Layer Package

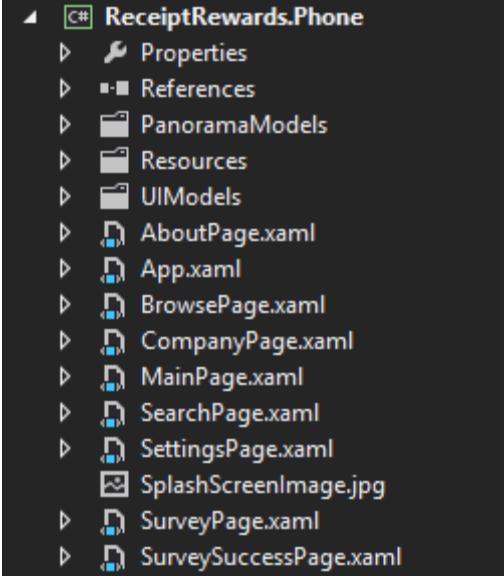
The Service Access Layer contains all the definitions to both the Web Services and the Reverse Proxy URL Locations.



3.6 View Models

The View Models is the most important aspect of the PCL in regards to the phone applications. These classes are logical representations of all the screens that will be in all the phone applications. In other words, these are only the business logic and functionality of a page. These are everything but the UI and event handlers. The actual applications will utilize a view model and hook up the UI and handlers to the logic located in the view models.



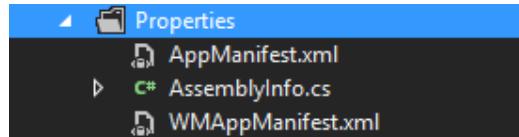


4 Windows Phone

The windows phone project is the source code of creating the phone application. The structure to the project follows the standard structure of the windows phone applications.

4.1 Properties Folder

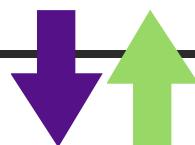
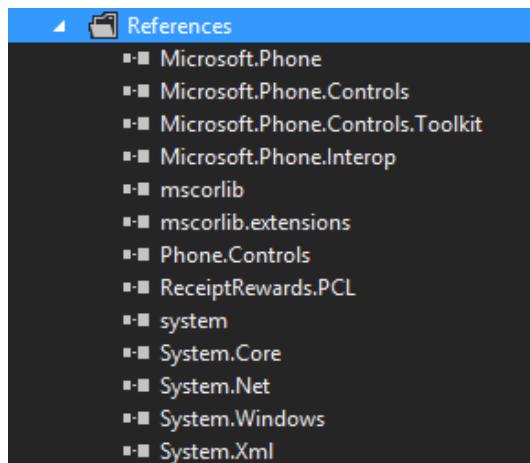
The properties folder contains the configuration files involved with how the app functions on the operating system.



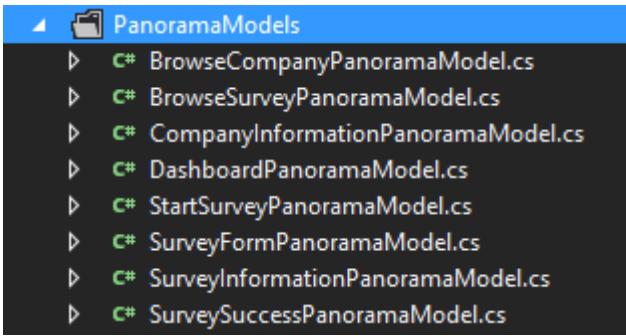
4.2 References Folder

The references folder contains all the libraries and jar files that get used in the phone application.

Specifically the libraries are the default jar files that are by default loaded into phone applications. Also, there are some open source controls that are installed too.



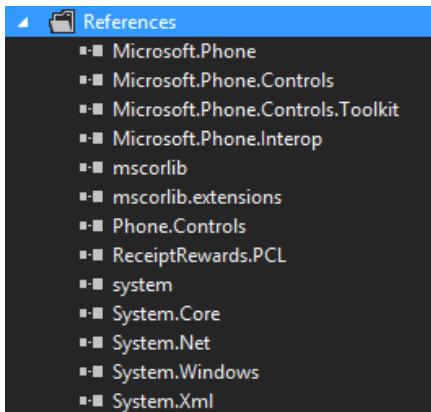
4.3 Panorama Model Package



The panorama package contains all the view objects that are the actual pages. These are the objects that get bind to the view models located inside the PCL library.

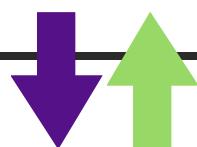
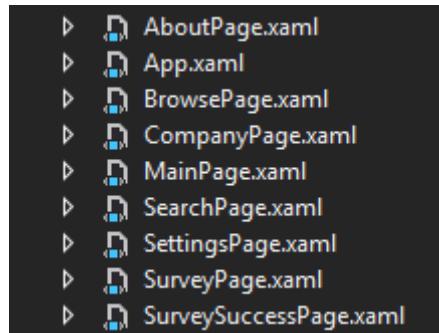
4.4 Resources Folder

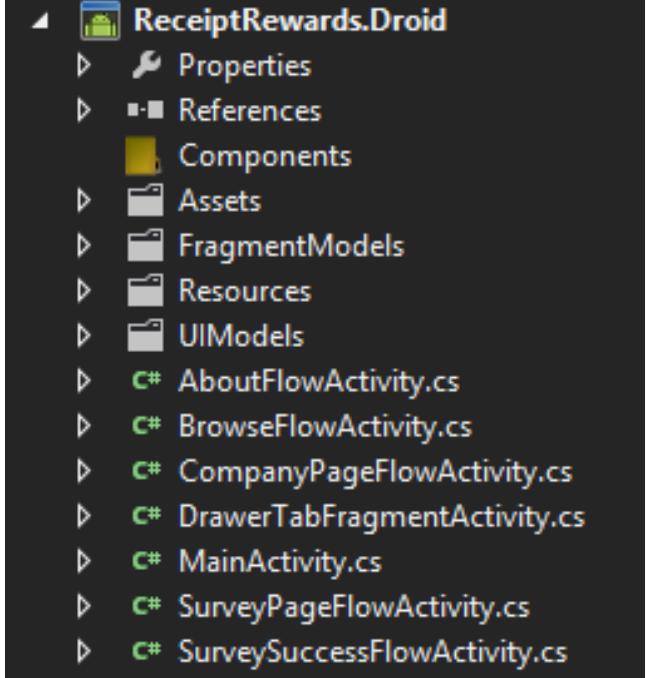
The resources folder contains all of the extra files that the phone will need. Specifically this contains images that are displayed to the user.



4.5 Pages

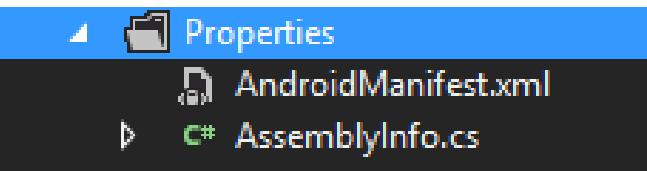
These are the XAML pages are essentially an empty wrapper that is referenced to a backend .net instance. That instance is what creates programmatically a panorama view or single page view. It then calls the views from the Panorama Model package to fill with dynamic content. The content is the actual implementation of the view models.





5 Android

The android files system also follows a very standard structure. This file structure also follows the MVVM Cross suggested structure.



5.1 Properties

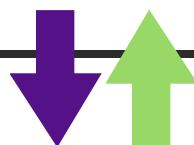
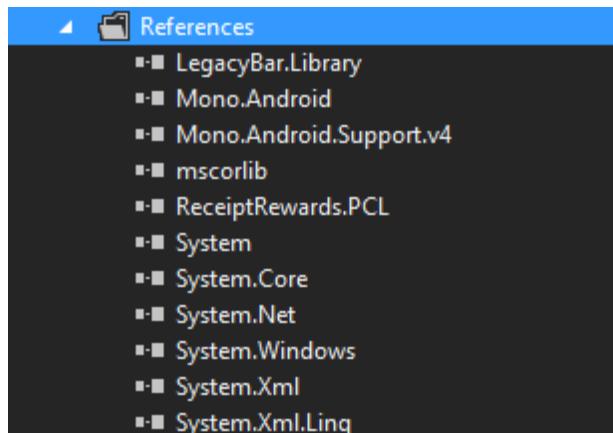
The properties folder contains the configuration of how the app will work on the Android OS. Some of these properties may include:

- Internet Connectivity
- App Name
- Logo
- Author Information

5.2 References

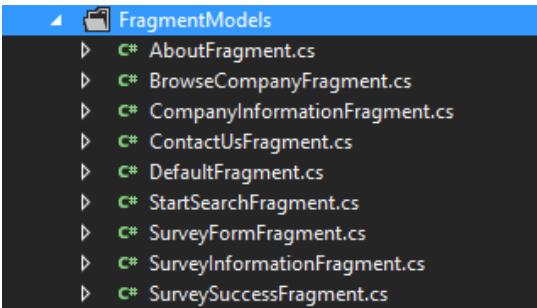
The references files structure contains all the dependencies of the Android Application. Most of these dependencies will include:

- Android OS Libraries (Mono.Android)
- .Net Libraries
- Android OS Component Jars (for display elements)



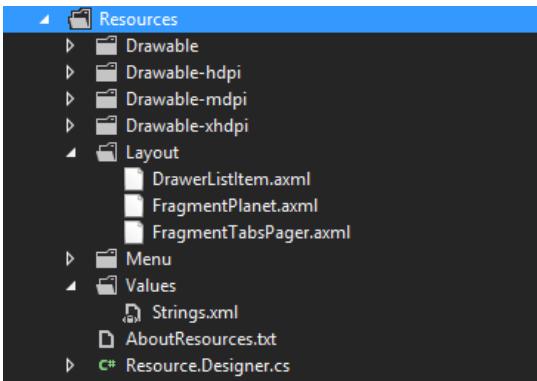
5.3 Fragment Models Package

The Fragment Models Package is the realization of the View Models (of the PCL.) These create programmatically a User Interface and event handlers, which it then hooks up to the View Model logic so that event handlers can automatically call the backend logic and update the UI when appropriate.



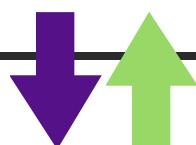
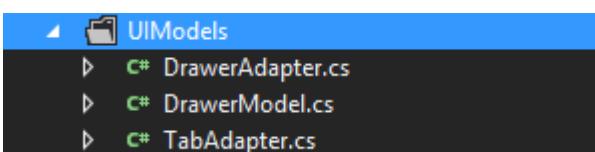
5.4 Resources Package

The resources folder contains a simple file system. These files include Pictures, logos and graphics that are displayed out to the user. It may also contain text files and lists which are used as static text to be display to the user, and create menu systems.



5.5 UI Models Package

UI Models is a package which contains static methods which are used to generate a User Interface object. The reason to put these components in here is to get reusability for elements that are recreated throughout the application. The structures in here are the Drawer and the Tab system. UI components in this package typically larger amount of complicated code and logic which also makes it advantageous to separate out.

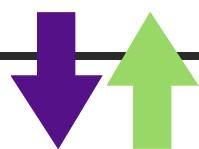


5.6 Pages

The main Android pages as you can see are located in the root view of the application.

These pages are actually containers to hold the actual content, (the Fragments.) The pages are used when linking between pages. When a page is loaded, the onLoad function programmatically adds all the fragments into the main view.

- ▷ `C# AboutFlowActivity.cs`
- ▷ `C# BrowseFlowActivity.cs`
- ▷ `C# CompanyPageFlowActivity.cs`
- ▷ `C# DrawerTabFragmentActivity.cs`
- ▷ `C# MainActivity.cs`
- ▷ `C# SurveyPageFlowActivity.cs`
- ▷ `C# SurveySuccessFlowActivity.cs`



6 Web Admin CMS

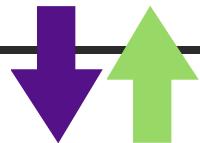
This System contains the code base which creates the Web Admin CMS application. This application is used to manage all the data in the database. Since this is a typical .Net Web Application, it follows a very similar file structure utilizing folders such as App_Code. The structure also contains standard website organization such as a Styles and Scripts folder.

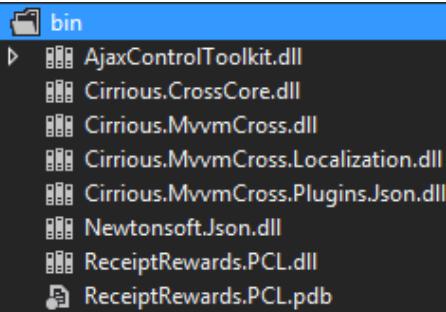
ReceiptRewards.WebCMS
App_Code
App_Data
bin
Includes
Scripts
Site_Files
Styles
Code.aspx
Code_Type.aspx
Code_Type_Manager.aspx
CommandElement.aspx
CommandIdentifier.aspx
Company.aspx
Company_Manager.aspx
FormElement.aspx
FormElementAttribute.aspx
FormElementOption.aspx
Index.aspx
Question.aspx
Revision.aspx
Survey.aspx
Web.config

6.1 App Code

The App_Code package contains all the C# static classes which contain logic that is utilized by the code_behind files. These Implementations are broken down by the models and contain the logic for restful services. They also include logic to generate the breadcrumbs for their given page.

App_Code
Impls
AutomationFlowImpl.cs
AutomationImpl.cs
CodeImpl.cs
CodeTypeImpl.cs
CommandElementImpl.cs
CommandFormElementRelation.cs
CommandIdentifierImpl.cs
CompanyImpl.cs
FormElementAttributeImpl.cs
FormElementImpl.cs
FormElementOptionImpl.cs
FormFlowImpl.cs
FormImpl.cs
QuestionImpl.cs
RevisionImpl.cs
SurveyImpl.cs
Global.cs



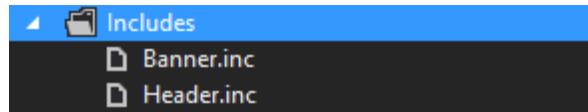


6.2 Bin

The bin folder contains all the jars and dll files which are the libraries that are used in the application. Some of these dependencies include .Net Controls.

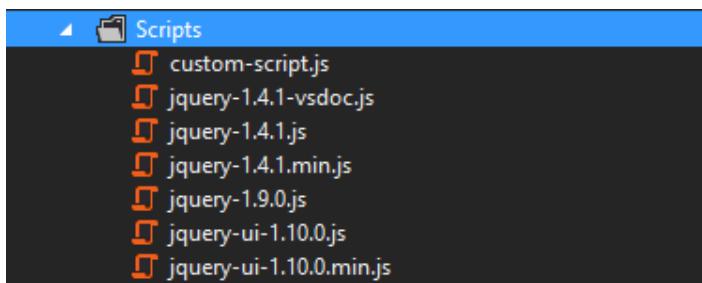
6.3 Includes

The Includes folder contains HTML / JavaScript that it used on every page. The Banner file is the logic for the actual generation of the banner and breadcrumbs. The Header contains the logic and includes for all the pages. This package demonstrates an emphasis of code reusability.



6.4 Scripts

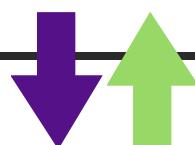
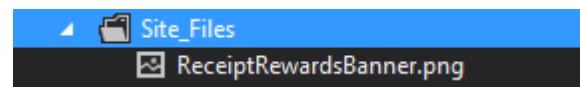
The script folder contains all the JavaScript libraries that are used in the application. As you can see the only libraries that are used are jQuery libs. The custom-script.js



contains custom written JavaScript for the Admin Site that is reused throughout the pages. An example of the code in here is the logic to collapse and expand the buttons on pages.

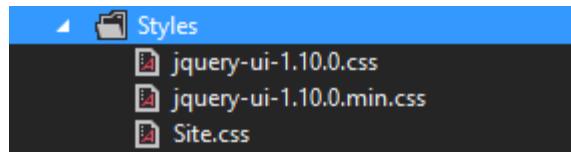
6.5 Site Files

The site files folder contains files that are used in the website. Types of files would include images, pdfs, txts, etc. As you can see, the application is pretty streamlined with only the banner graphic being the only image used in the website.



6.6 Styles

The style directory is the folder in which all the CSS Styling files will be placed. This provides easy organization.



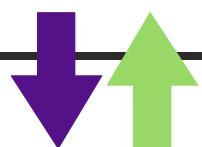
6.7 Pages

All the pages are located in the Root directory for the admin site. As you can see, there

- ▷ Code.aspx
- ▷ Code_Type.aspx
- ▷ Code_Type_Manager.aspx
- ▷ CommandElement.aspx
- ▷ CommandIdentifier.aspx
- ▷ Company.aspx
- ▷ Company_Manager.aspx
- ▷ FormElement.aspx
- ▷ FormElementAttribute.aspx
- ▷ FormElementOption.aspx
- ▷ Index.aspx
- ▷ Question.aspx
- ▷ Revision.aspx
- ▷ Survey.aspx
- ▷ Web.config

is no sub-navigation or nesting of the pages into subdirectories because the pages were self-evident and work in a very linear fashion.

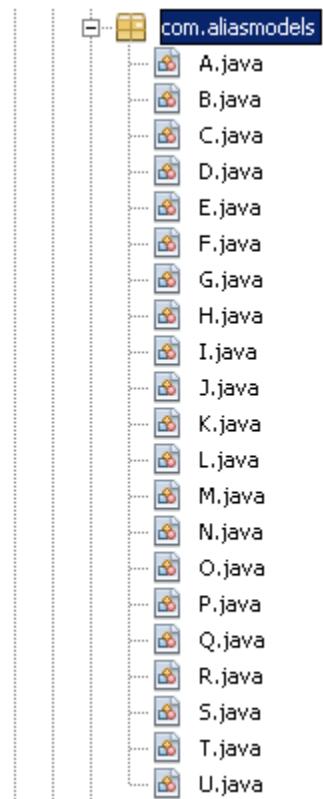
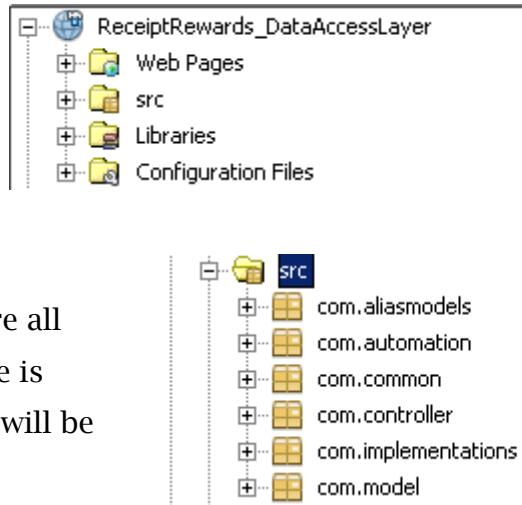
All the web pages are ASPX files, which is a special extension for the .Net Framework. Every page has an associated “code-behind” file, which contains C# code that contains the logic of event handlers. The .Net framework automatically binds the ASP controls with these functions.



7 Web Services

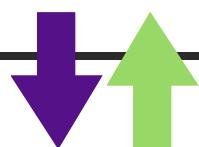
The web services are a Java Project that you will need NetBeans to manage.

The structure for the web services are all primarily in the src folder. The source is broken down into packages of which will be discussed more in-depth next.



7.1 Alias Models Package

The Alias Models package contains the list of objects that are alias versions of the models.



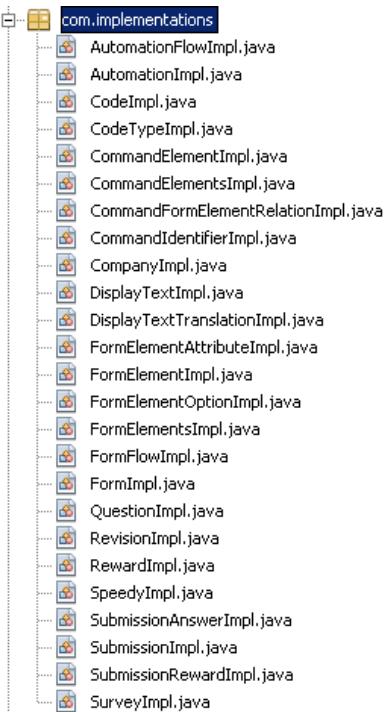
7.2 Automation Package

The automation package contains the classes that are involved with performing the Web Driver Automation logic. The methods are kicked off by a web service.



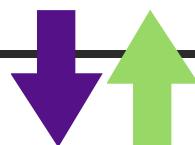
7.3 Common Package

The common package contains static methods that are used throughout the application. You can think of them as helper files. The AliasHelper contains all the logic to convert models to and from their alias model counterparts. The DBConnection is used with all the data access methods. It creates the connection to the database.



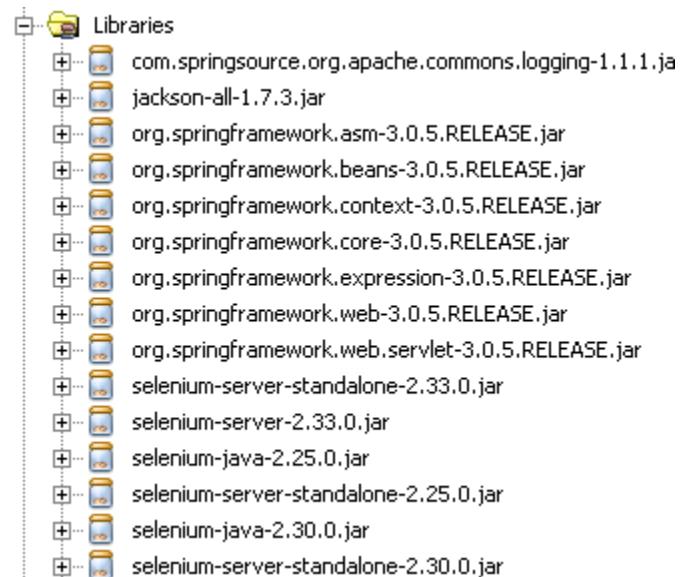
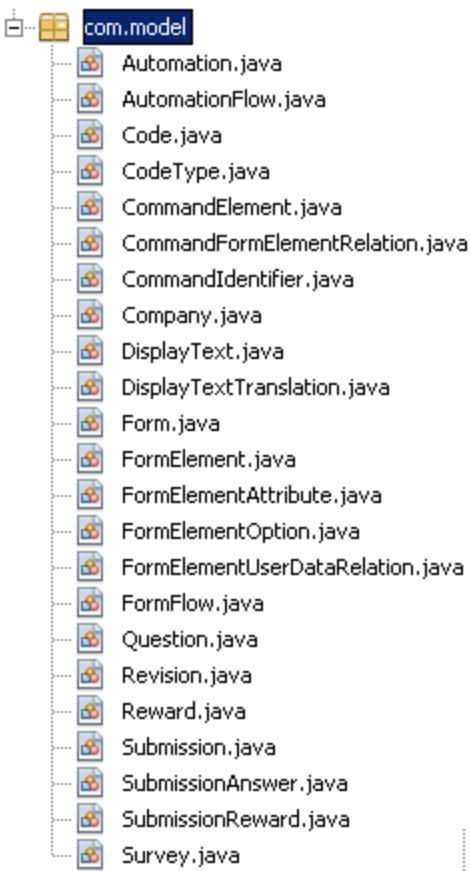
7.4 Controller Package

The Controller Packages contains classes that map the web services to a specific URL. These methods are divided to contain all the CRUD actions for a particular model object. The methods contain the definitions for what the requests and responses of the Web Services.



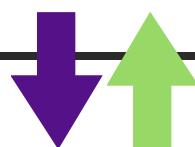
7.5 Models Package

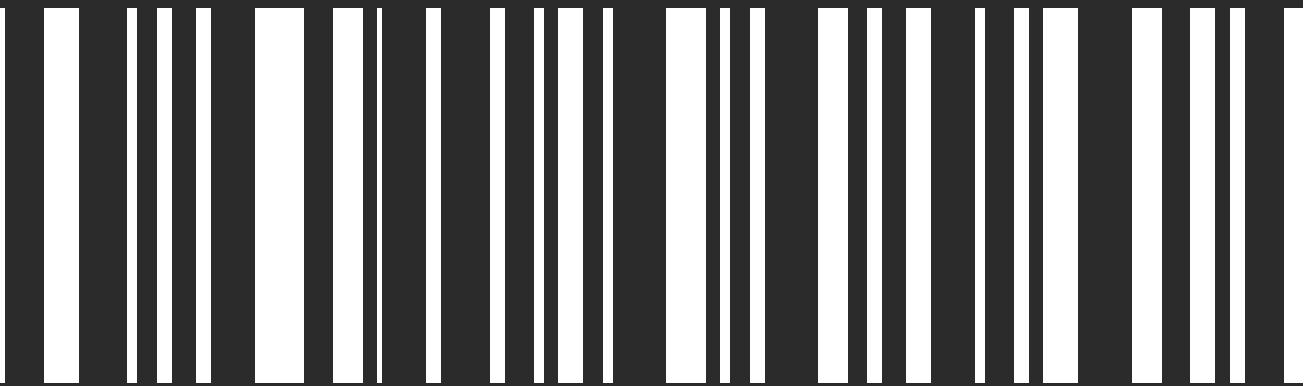
The Models Package contains the Models for this layer. Each object is a representation of the Object Orientated design that the database and the system follows.



7.6 Libraries

The libraries section contains a list of all the external libraries and dependencies that are used in the web services layer. As seen below, there is a strong dependency to the spring framework.





WEB ADMIN CMS USER MANUAL

8/23/2013

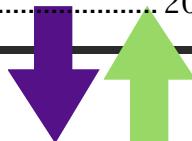
Graduate Capstone

RECEIPT

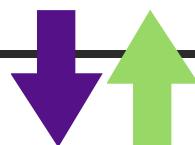
REWARDS

Table of Contents

1 Introduction.....	4
1.1 Intended Audience.....	4
1.2 References.....	4
1.3 Revision History	4
2 Set Up	5
2.1 General Flow of the Web App.....	5
3 Main Menu.....	6
4 Code Administration	6
4.1 Code Type Manager	7
4.1.1 Add Code	8
4.1.2 Code Type List	8
4.2 Code Type Page.....	9
4.2.1 Add Code	9
4.2.2 Delete Code Type	10
4.2.3 Save Code Type Changes.....	10
4.2.4 Code Entry (Edit and Delete)	10
4.3 Code Page.....	10
4.3.1 Edit Code.....	11
4.3.2 Delete Code	11
4.3.3 Save Code Changes	11
5 Company and Survey Administration	12
5.1 Company Manager	13
5.1.1 Add Company	13
5.1.2 Company List	14
5.2 Company Page.....	14
5.2.1 Edit Company.....	15
5.2.2 Delete Company	15
5.2.3 Add Survey	15
5.2.4 Survey List	16
5.3 Survey Page.....	16
5.3.1 Edit Survey	17
5.3.2 Delete Survey	17
5.3.3. Add Revision	17
5.3.4 Revision List	18
5.4 Revision Page.....	18
5.4.1 Edit Revision.....	19
5.4.2 Delete Revision	19
5.4.3 Add Question	19
5.4.4 Question List	20
5.4.5 Link Questions	20



5.4.6 Perform Automated Test.....	21
5.5 Question Page.....	21
5.5.1 Edit Question.....	22
5.5.2 Delete Question	22
5.5.3 Add Form Element	22
5.5.4 List of Form Elements	23
5.5.5 Add Automation Element.....	23
5.5.6 List of Automation Elements.....	24
5.5.7 Delete All Form Flows	24
5.5.8 Delete All Automation Flows.....	24
5.6 Form Element Page	25
5.6.1 Edit Form Element.....	25
5.6.2 Delete Form Element	26
5.6.3 Add Form Element Option	26
5.6.4 List Form Element Options	26
5.6.5 Add Form Element Attribute.....	27
5.6.6 List Form Element Attribute	27
5.7 Form Element Option Page	28
5.7.1 Edit Form Element Option.....	28
5.7.2 Delete Form Element Option	29
5.8 Command Element Page	29
5.8.1 Edit Command Element.....	30
5.8.2 Delete Command Element.....	30
5.8.3 Add Command Element Identifier	31
5.8.4 List Command Element Identifier	31
5.8.5 Add Command Form Element Relation	32
5.8.6 List Command Form Element Relation	32
5.9 Command Element Identifier Page.....	33
5.9.1 Edit Command Element Identifier.....	33
5.9.2 Delete Command Element Identifier	34
7 Frequently Asked Questions.....	35
7.1 I made a change, but the webpage didn't update?.....	35
7.2 Are there any credentials need to use the Admin Site	35



1 Introduction

The purpose of this document is to explain and give an in depth analysis on the structure, design and make up of Company Survey Websites. By doing so, this will give a great understanding to the complexity involved with having to re-create these surveys in the system.

1.1 Intended Audience

This document is intended for persons trying to understand the fundamental business side of how survey work for this project. The goal is for those individuals to learn about the surveys and learn what is involved to recreate the surveys in the system.

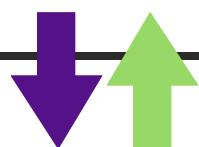
1.2 References

No references were used.

1.3 Revision History

Managing the change history of this document will occur in this table.

Name	Date	Reason For Change	Version
Andy Bottom	08/08/2013	Created the introduction and parts of the document.	0.1
Andy Bottom	08/22/2013	Finished the rest of the manual.	1.0

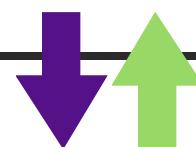
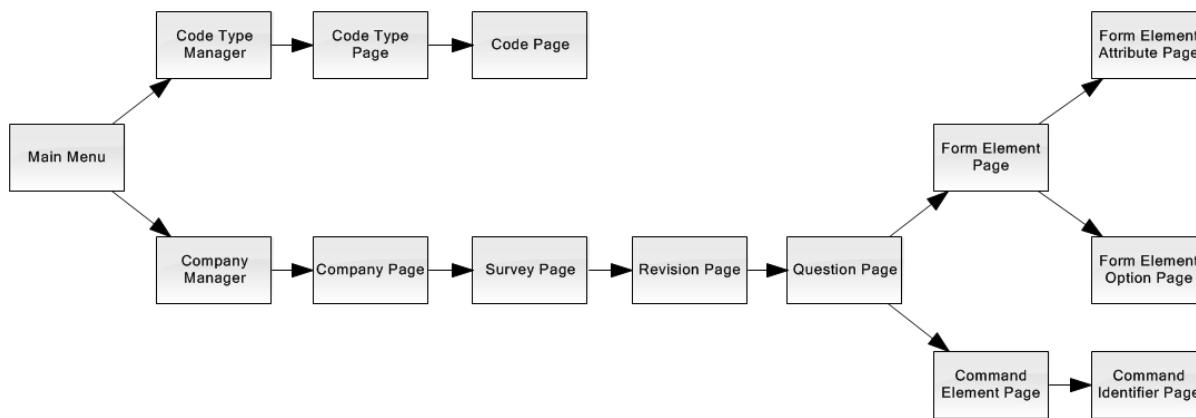


2 Set Up

The Web Admin requires a server running .Net 4.0 and up in order to run. The project can be supplied upon request.

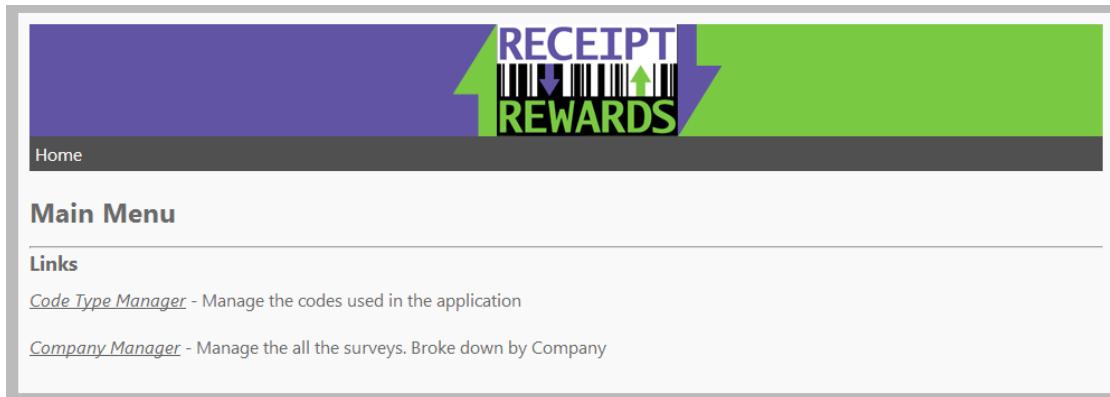
2.1 General Flow of the Web App

Below is a diagram which shows the overall flow of the Web Admin CMS Application and the interaction between pages.



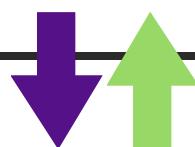
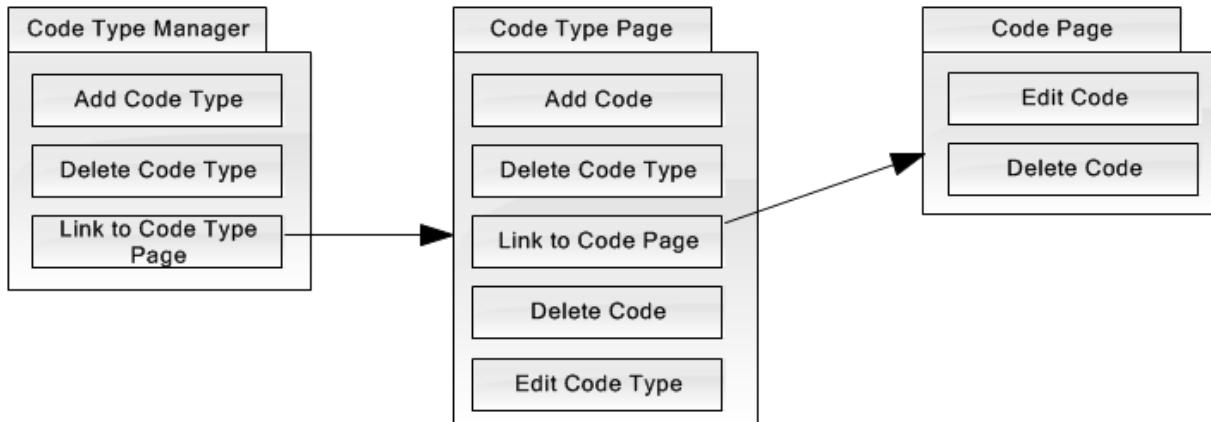
3 Main Menu

The main menu contains the links to the major sections of the application. All the administrator functionality can be linked to from this page.



4 Code Administration

The Code Administration of the Application is used to manage all the codes and code types of the system. The codes are essentially constants of the system that are used and referenced in the code.

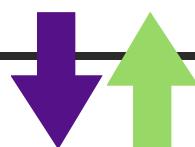


4.1 Code Type Manager

The Code Type Manager page is used to manage all the Code Types in the system.

Code Types are essentially categories for the Codes. The Code Types have a certain purpose and part of the system for specialized functionality.

Code Types will be updated infrequently and almost always be updated by a developer. The reason is because when code types need to be updated, it almost always involves is due to a new development request that needs the new Code Type. Otherwise, most code types have no need to be updated once they are made. Also, it is very important to be diligent when deleting a code or code type because there may be references associated to those.





Add Code Type

Create Code Type

Type:

Description:

[Add Code Type](#)

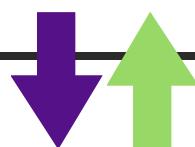
4.1.1 Add Code

- Use the Add Code Type form to add a new Code Type to the system.
- The Type field is a unique string to identify a single Code Type
- The description field is a string which types will be used in reference

4.1.2 Code Type List

On the Code Type Manager is a list of all the Code Types in the system. With this list you can perform a delete of the Code Type. Also, if you wish to edit the Code Type, click the “edit” link and that will take you to the Code Type Page.

List of Code Types			
Title	#	Description	Options
COMMAND_ELEMENT	11	The command element types	Edit Delete
FORM_ELEMENT_TY	8	form element type	Edit Delete
COMPANY_TYPE	1	What the category of the company is.	Edit Delete
REWARD_TYPE	1	The type that the reward will be.	Edit Delete
COMMAND_ELEMENT	7	The Type Code for a Command Element's Id	Edit Delete
FORM_ELEMENT_AT	2	The Type for a Form Element's Attribute	Edit Delete
FORM_ELEMENT_OP	0	The Type for a Form Element's Option Obj	Edit Delete



4.2 Code Type Page

The Code Type page is the page that allows you to manage all aspects of a code that is in the system. You can also manage Codes that are part of that Code Type.

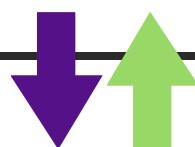
The screenshot shows the Receipt Rewards Code Type management interface. At the top, there's a purple header with the Receipt Rewards logo, which includes a barcode icon. Below the header, the URL 'Home >> TEST_CODE_TYPE' is visible. The main content area has a title 'TEST_CODE_TYPE'. On the left, there are input fields for 'Type' (set to 'TEST_CODE_TYPE') and 'Description' (containing 'This is a Test Code Type.'), both with scroll bars. To the right of these fields is a vertical menu with three options: 'Add Code', 'Delete Code Type', and 'Save All Changes'. Below this menu, there's a section titled 'List of Codes' containing a single entry: 'Code: TEST_CODE', 'Name: Test Code', 'Description: This is a test code.', and 'Options: Edit | Delete'. The entire interface is framed by a light gray border.

The screenshot shows the 'Add Code' form. It has a dark header bar with the text 'Add Code'. Below it is a 'Create Code' section with three input fields: 'Code' (empty), 'Name' (empty), and 'Description' (empty). At the bottom of the form is a single button labeled 'Add Code'.

4.2.1 Add Code

Use the Add Code form to add a new Code to the current Code Type.

- The Code field is a unique identifying name.
- The name is the human readable version of the code.
- Description is a description to describe what purpose the Code has.



4.2.2 Delete Code Type

Delete Code Type

To delete a Code Type, click the “Delete Code Type” button.

4.2.3 Save Code Type Changes

Save All Changes

To save changes to the Code Type, click the “Save All Changes” button.

4.2.4 Code Entry (Edit and Delete)

At the bottom of the page is a list of Codes. These codes are all the codes that are part of the current Code Type. There are links to delete or edit that code. The edit will link to the Code Page for that code.

Code	Name	Description	Options
TEST_CODE	Test Code	This is a test code.	Edit Delete

4.3 Code Page

The Code Page is the page to manage a Code that is in the system.

RECEIPT
REWARDS

Home >> TEST_CODE_TYPE >> TEST_CODE

TEST_CODE

Code
TEST_CODE

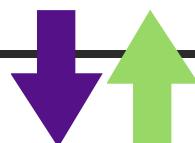
Type
Test Code

Description
This is a test code.

Code Type
TEST_CODE_TYPE

Delete Code

Save All Changes



4.3.1 Edit Code

The Edit form allows the user to update properties of the current Code.

- Code field is a string that represents the code, which is a computer readable identifier
- The name field is a human readable name for the code.
- The Description field is a string that summarizes the purpose of that code and what it represents.

Code	TEST_CODE
Type	Test Code
Description	This is a test code.
Code Type	TEST_CODE_TYPE

4.3.2 Delete Code

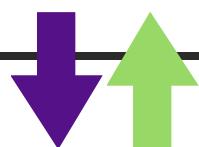
To delete the code, click the “Delete Code” button.

Delete Code

4.3.3 Save Code Changes

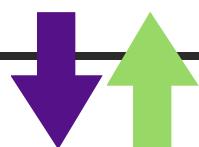
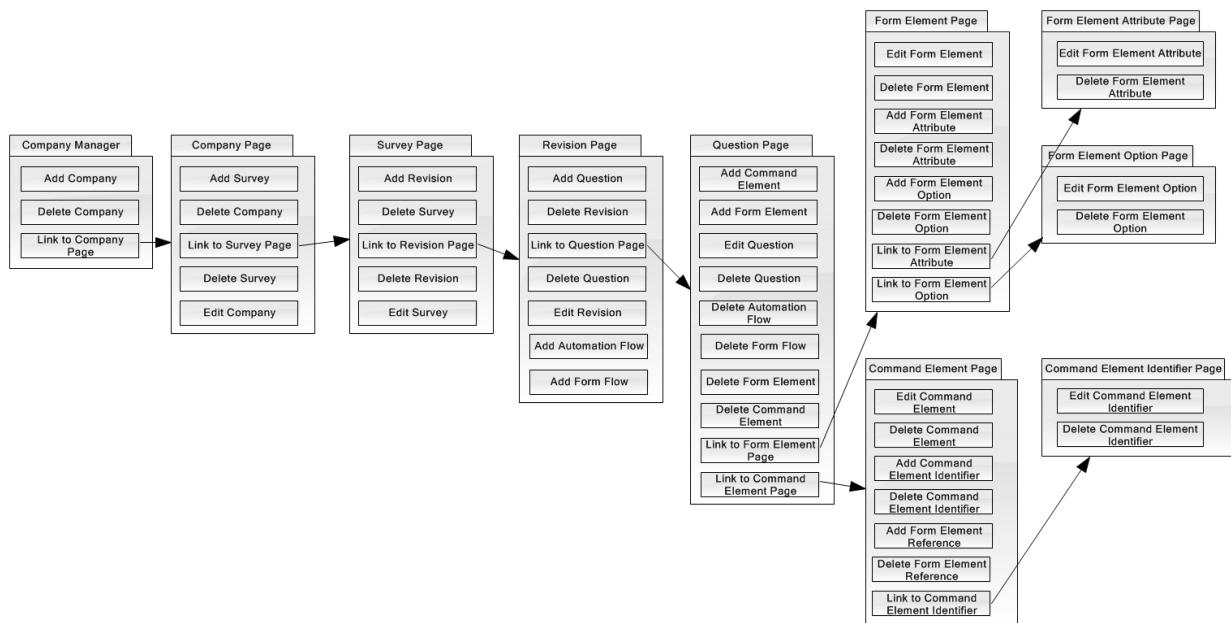
To save all the changes on the Edit Code form, then click the “Save All Changes” button

Save All Changes



5 Company and Survey Administration

The following next module of the application involves managing all the surveys and company in the CMS. Below is a diagram showing the flow of the Company and Survey Administration Module with all the actions of each page.



5.1 Company Manager

The company manager is a list that displays all the companies in the system. This is the starting point to drill down into the companies.

The screenshot shows the Receipt Rewards Company Manager page. At the top, there's a purple header bar with the Receipt Rewards logo, which includes a barcode graphic. Below the header, a dark grey navigation bar contains the word "Home". The main content area has a light grey background. On the left, under "Company Info", it says "These are all the companies in the system. Actions include Add, Delete, Edit." Below this is a table titled "Company List" showing five entries:

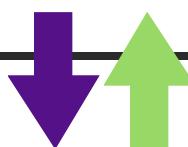
Name	#	Options
Culvers	0	Edit Delete
Taco Bell	0	Edit Delete
Dunkin Donuts	0	Edit Delete
Test Company	0	Edit Delete

On the right side, there's a sidebar with a dark grey header "Add Company" and a section titled "Create Company" containing a "Name:" input field and a "Add Company" button.

5.1.1 Add Company

To add a new company to the system use the add company form. The “Name” field requests a string which would be the name of the Company. Then click the “Add Company” button to add the company to the system.

The screenshot shows the "Add Company" form. It has a dark grey header "Add Company" and a section titled "Create Company". It contains a "Name:" input field and a "Add Company" button.



5.1.2 Company List

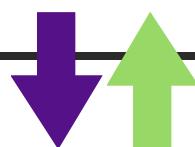
On the Company Manager Page, you will find that there is a list which shows all the companies in the system. With this list, you can delete the company by clicking the “Delete” link. To edit the company, click the “Edit” link and that will direct you to the Company Page.

Test Company 0 [Edit | Delete](#)

5.2 Company Page

The Company Page is used to manage all the properties for any given Company object that is in the system. A Company is a representation of the actual Company of which the desired survey is actually a part of.

The screenshot shows the Company Page interface. At the top, there is a header with the "RECEIPT" logo and a barcode graphic. Below the header, the page title is "Home >> Test Company". The main content area has a light gray background. On the left, there is a form with a "Name" field containing "Test Company". On the right, there are three dark gray buttons labeled "Add Survey", "Delete Company", and "Save All Changes". Below this, under the heading "List of Surveys", there is a table with one row. The table columns are "Name", "#", and "Options". The row contains "Test Survey", "0", and an "Edit | Delete" link. The entire screenshot is framed by a thick gray border.



5.2.1 Edit Company

To edit attributes about the Company, use the Edit Company Form. This form contains the Name of the Company which should be a string. Once you have entered in all the desired information, click the “Save All Changes” button to save your changes.

Save All Changes

Name
Test Company

5.2.2 Delete Company

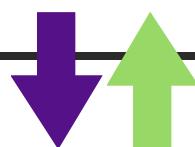
To delete the current company from the system, click the “Delete Company” button. This will delete the company and redirect you back to the Company Manager Page. Remember, this action cannot be undone, and all the information associated to company will also be removed.

Delete Company

5.2.3 Add Survey

To create a new survey for the company, use the “Add Survey” form. In this form, input a string value into the “Name” field, which would represent a meaningful name to that survey. Once the fields have been entered, click the “Add Survey” button and this will create the survey. You can now find the newly created Survey in the Survey List.

Add Survey	
Survey	
Name	<input type="text"/>
Add Survey	



5.2.4 Survey List

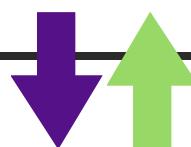
To view a list of all the Surveys that are associated to the Company, look at the Survey List. In this list you can perform the following actions to the Survey. To delete the Survey, click the “Delete” link, which will remove the Survey and all related objects of the Survey in the system. If you want to make changes to the Survey, click the “Edit” link and you will be redirected to the Survey Page.

Name	#	Options
Test Survey	0	Edit Delete

5.3 Survey Page

The Survey Page is a representation of a Survey Object in the database. The Survey is a representation of an actual Survey that will be created for users to fill out.

The screenshot shows a web page titled "RECEIPT REWARDS" with a barcode graphic. The top navigation bar includes links for "Home", "Test Company", and "Test Survey". The main content area displays the survey details: "Name: Test Survey". On the right, there are three buttons: "Add Revision", "Delete Survey", and "Save All Changes". Below this, a section titled "List of Revisions" shows one entry: "Revision Number: Test Revision", "Active: True", "Date Created: 8/12/2013", and "Options: Edit | Delete".



5.3.1 Edit Survey

To change information of the current survey, use the “Edit Survey” form. This form contains a Name field which is a string of a meaningful name for the survey. Once all the fields have been entered, click the “Save All Changes” button. The page will refresh and you can see the new changes.

The screenshot shows a user interface for editing a survey. At the top, there is a header bar with the title 'Edit Survey'. Below it is a form with a single input field labeled 'Name' containing the text 'Test Survey'. To the right of the input field is a dark grey button labeled 'Save All Changes'.

5.3.2 Delete Survey

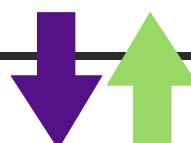
To delete the current Survey from the system, click the “Delete Survey” button. This will delete the survey and redirect you back to the parent Company Page. Remember, this action cannot be undone, and all the information associated to survey will also be removed.

The screenshot shows a user interface element consisting of two horizontal bars. The top bar is dark grey and contains the white text 'Delete Survey'. Below it is a lighter grey bar.

5.3.3. Add Revision

To create a new Revision for the Survey, use the “Add Revision” form. In this form, input a string value into the “Revision Number” field, which would represent a meaningful name for what version of the revision it is. The “Active” checkbox sets whether it will be the active Revision of the Survey. Once all the fields have been entered, click the “Add Revision” button and this will create the Revision into the system. You can now find the newly created Revision in the Revision List. Note that if the new Revision gets marked as the active one, then the other Revision of the Survey will be set to inactive.

The screenshot shows a user interface for adding a new revision. It has a header bar with the title 'Add Revision'. Below it is a form with a single input field labeled 'Revision Number' containing a placeholder 'Enter Revision Number...'. Underneath the input field is a checkbox labeled 'Active' with an unchecked state. At the bottom of the form is a dark grey button labeled 'Add Revision'.



5.3.4 Revision List

On the Survey Page, you will find that there is a list which shows all the Revisions associated to the Survey. With this list, you can delete a Revision by clicking the “Delete” link. To edit the Revision, click the “Edit” link and that will direct you to the Revision Page.

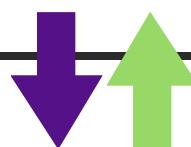
Revision Number	Active	Date Created	Options
Test Revision	True	8/12/2013	Edit Delete

5.4 Revision Page

The Revision Page is a representation of a Revision Object in the database. The Revision is a realization of an individual version of a Survey.

The screenshot shows a web-based application interface for managing a revision. At the top, there's a header with the logo 'RECEIPT REWARDS' featuring a barcode graphic. Below the header, a breadcrumb navigation shows 'Home >> Test Company >> Test Survey >> Test Revision'. The main content area has a title 'Test Company: Test Survey vTest Revision'. On the left, there are input fields for 'Name' (containing 'Test Revision') and 'Active' (with a checked checkbox). On the right, a vertical sidebar contains five buttons: 'Add Question', 'Link Questions', 'Delete Revision', 'Save All Changes', and 'Perform Automated Test'. Below these buttons, there's a large text area containing placeholder text: 'Lorem ipsum dolor sit amet, consectetur adipiscing elit. Pellentesque nisl mauris, mollis in pretium non, feugiat ut mi. Etiam facilisis suscipit turpis, vitae rhoncus elit sollicitudin at. Mauris id dui id metus egestas feugiat quis iaculis magna. Pellentesque fringilla venenatis lorem sed porta. Donec feugiat dui sed massa imperdiet ac euismod sapien sodales. Duis in congue purus. Quisque sed tincidunt arcu. Pellentesque enim urna, imperdiet sit amet pretium ut, auctor nec neque.' At the bottom, there's a section titled 'Questions' with a table listing three questions:

ID	Name	Frm Ref	Cmd Ref	Options
80	Start Question	81	81	Edit Delete
81	Question 01			Edit Delete
82	Question 02			Edit Delete
83	Question 03			Edit Delete



5.4.1 Edit Revision

To change information of the current Revision, use the “Edit Revision” form. This form contains the “Name” field which should be a string containing meaningful version name of the Revision. The “Active” field is a checkbox to determine if the current

Save All Changes

Revision is the active one for the Survey. Once all the fields have been entered, click the “Save All Changes” button. The page will refresh and you can see the new changes. Note that if you set the Revision to active, the other Revision will be set to inactive as only one Revision can be active per Survey.

Name	
Test Revision	
Active	<input checked="" type="checkbox"/>

5.4.2 Delete Revision

To delete the current Revision from the system, click the “Delete Revision” button. This will delete the revision and redirect you back to the parent Survey Page. Remember, this action cannot be undone, and all the information associated to survey will also be

Delete Revision

removed. Also note, that very rarely, revisions should never be deleted if

the Revision has been in use by users because this will delete all the references.

Instead, it is advised to simply create a new revision and set that one as active. This way, this survey will be inactive, but still in the database to preserve the data.

5.4.3 Add Question

To create a new Question for the revision, use the “Add Question” form. In this form, input a string value into the “Name” field, which would

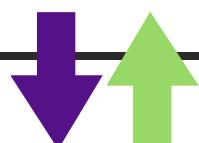
Add Question

Add Question

Name

Add Question

represent a meaningful name to the question. Once all the fields have been entered, click the “Add Question” button and this will create the Question into the system. You can now find the newly created Question in the Question List.



5.4.4 Question List

On the Revision Page, there is a list of all the Questions associated to the current Revision. The columns show properties of the Question. The ID column is the Question ID. The name column is the meaningful name of the Question. The Frm Ref is shows the ids of the next Questions in the Form flow. The Cmd Ref shows the next ids of Questions for the Automation flow. In the Options column, you can delete a Question by clicking the “Delete” link. To edit the Question, click the “Edit” link and that will direct you to the Question Page.

ID	Name	Frm Ref	Cmd Ref	Options
80	Start Question	81	81	Edit Delete

5.4.5 Link Questions

The “Link Questions” form is used to add new flows to a Question. There are two types of Flows, Form and Automation. The flow is essentially a reference for where the question could possibly go to next.

To create a new Flow for a Question, use the “Link Questions” form. Select either Form or Automation in the dropdown box for what flow the reference should be added to. The “Start Question” field requires an integer of the ID of the Question which the reference will be added to. The “End Question” field requires an integer of the ID of the Question which is the target of the reference. The “Value / Weight” field varies on the

Link Questions

Link Questions

Form Start Question

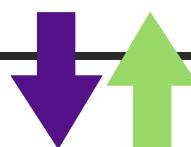
End Question

Value / Weight

Link Questions

flow. In the Form Flow, the field requires a string for the value that the flow will take if found, (the default flow is left blank.) In the Automation Flow, the field requires an integer to identify the prioritization weight of the flow. The default flow should have the largest value, which will execute last. Once all the fields have been entered, click the “Link Questions” button and this will

create the Flow reference into the system. You can now see the added flow reference under the Question in either the Frm Flow or Cmd Flow columns of the Question List.



5.4.6 Perform Automated Test

To test whether or not the revision still works, you can manually perform an automated test, but clicking the “Perform Automated test” button. If the test succeeds, then a success message will display saying the test was a success. Otherwise an error message will be returned. If an error message does occur, then a more

Perform Automated Test

thorough look is needed by the admin. Remember, a success means that the survey is up-to-date, but a failure doesn't necessarily mean that the revision is out of date, but does require a further analysis.

5.5 Question Page

The Revision Page is the management of Question object in the database. The Question is a representation of a single question or section of the satisfaction survey.

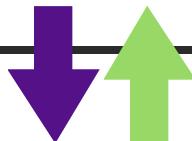
The screenshot shows a web-based application interface for managing questions. At the top, there's a purple header bar with the text "RECEIPT" and "REWARDS" in white, accompanied by a barcode icon. Below the header, a navigation breadcrumb path is visible: Home >> Test Company >> Test Survey >> Test Revision >> Question 01. The main content area has a light gray background. On the left, there's a sidebar with several dark gray buttons containing white text: "Add Automation Command Element", "Add Form Element", "Delete Question", "Delete All Form Flows", "Delete All Automation Flows", and "Save All Changes". The main content area contains two tables: "Automation" and "Form". The "Automation" table has two rows:

ID#	Type	Idntfs	Optns
132	TEXTBOX	0	Edt Dlt
133	BUTTON	0	Edt Dlt

The "Form" table has two rows:

ID#	Type	Idntfs	Optns
137	LABEL	0	Edt Dlt
138	TEXTBOX	0	Edt Dlt

Below the tables, there is a large amount of placeholder text (Lorem ipsum) and a small note about the survey's status.



5.5.1 Edit Question

To change information of the current Question, use the “Edit Question” form. This form contains the “Name” field which should be a string containing meaningful name of the Question. Once all the fields have been entered, click the “Save All Changes” button. The page will refresh and you can see the new changes.

Save All Changes

Name
Question 01

5.5.2 Delete Question

To delete the current Question from the system, click the “Delete Question” button. This will delete the question and redirect you back to the parent Revision Page. Remember, this action cannot be undone, and all the information associated to question will also be removed.

Delete Question

5.5.3 Add Form Element

To create a new Form Element for the Question, use the “Add Form Element” form. In this form, select the desired “Type” from the drop down box which would represent

Add Form Element

Add Form Element

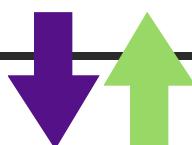
Type
Textbox

or from existing

Add Form Element

what type of form element it is, (the options are created from the Codes in the FORM_ELEMENT_TYPE code type.) Another option is to input an integer of an existing Form Element’s id. This will create a new reference to the current question, (and not create a new Form

Element.) Otherwise, once all the desired fields have been entered, click the “Add Form Element” button and this will create the Form Element into the system. You can now find the newly created or associated Form Element in the Form Element List.



5.5.4 List of Form Elements

On the Question Page, there is a list of all the Form Elements associated to the current Question and Form. The columns display several attributes. The ID column is the Form Element Id. The type column is the Type of the Form Element. In the Options column, you can delete a Command Element by clicking the “Delete” link. To edit the Form Element, click the “Edit” link and that will direct you to the Form Element Page.

ID#	Type	Idntfs	Optns
137	LABEL	0	Edt DltUpDown
138	TEXTBOX	0	Edt DltUpDown

5.5.5 Add Automation Element

To create a new Automation Element for the Question, use the “Add Command Element” form. In this form, select the desired “Type” from the drop down box which would represent what type of Command Element it is, (the options are created from the Codes in the COMMAND_ELEMENT_TYPE code type.) Another option is to input an integer of an existing Command Element’s id. This will create a new reference to the current question, (and not create a new Command Element.) Otherwise, once all the desired fields have been entered, click the “Add Command Element” button and this will create the Command Element into the system. You can now find the newly created or associated Command Element in the Command Element List.

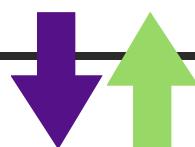
Add Automation Command Element

Add Command Element

Type
Textbox

Test Data
or from existing

Add Command Element



5.5.6 List of Automation Elements

On the Question Page, there is a list of all the Command Elements associated to the current Question and Automation. The columns display several attributes. The ID column is the Command Element Id. The type column is the type of the Command Element.

ID#	Type	Idntfs	Optns
132	TEXTBOX	0	Edt Dlt
133	BUTTON	0	Edt Dlt

In the Options column, you can delete a Command Element by clicking the “Delete” link. To edit the Command Element, click the “Edit” link and that will direct you to the Command Element Page.

5.5.7 Delete All Form Flows

Delete All Form Flows

To delete the all Form Flows related to the current Question, click the “Delete All Form Flows” button. This will delete the associated Form Flows. Remember, this action cannot be undone, and all the references that the Form Flows created will be removed.

Delete All Automation Flows

To delete the all Automation Flows related to the current Question, click the “Delete All Automation Flows” button. This will delete the associated Automation Flows. Remember, this action cannot be undone, and all the references that the Automation Flows created will be removed.



5.6 Form Element Page

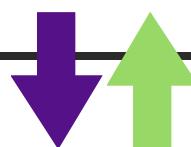
The Form Element Page is the management of Form Element object in the database. The Form Element is a representation of a single element (textbox, checkbox, etc...) in the satisfaction survey.

The screenshot shows a web page titled "RECEIPT REWARDS" with a purple and green header. The main content area shows a breadcrumb navigation: Home >> Test Company >> Test Survey >> Test Revision >> Question 01. Below this, the page title is "Test Company: Test Survey". A dropdown menu labeled "Type" is set to "Textbox". To the right is a vertical sidebar with buttons for "Add Attribute", "Add Option", "Delete Form Element", and "Save All Changes". The "List of Attributes" section has columns for Type, Value, and Options. The "List of Options" section shows one entry with ID 408, Text "Test", Value "Test", and Options "Edit | Delete".

5.6.1 Edit Form Element

To change information of the current Form Element, use the “Edit Form Element” form.

This screenshot shows the "Edit Form Element" form. It features a dropdown menu labeled "Type" with "Textbox" selected. Below it is a large text input field containing the word "Test". To the right is a large button labeled "Save All Changes".



5.6.2 Delete Form Element

To delete the current Form Element from the system, click the “Delete Form Element” button. This will delete the form element and redirect you back to the parent Question Page. Remember, this action cannot be undone, and all the information associated to form element will also be removed.

Delete Form Element

5.6.3 Add Form Element Option

To create a new Form Element Option for the Form Element, use the “Add Option” form. In this form, input a string value into the “Display Text” field, which would represent what the text of the option will be displayed to the user. The “Value” field is a string of the hidden value of the field that will be placed in the Submission Answer when the user submits the Form.

Add Option

Form Element Option

Display Text

Value

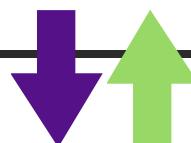
Add Option

Once all the fields have been entered, click the “Add Option” button and this will create the Form Element Option into the system. You can now find the newly created Form Element Option in the Form Element Option List.

5.6.4 List Form Element Options

On the Form Element Page, there is a list of all the Form Element Options associated to the current Form Element. The columns display several attributes. The ID column is the Form Element Option Id. The text column is the Display Text of the Form Element Option. The value column shows the hidden value of the Form Element Option. In the Options column, you can delete a Form Element Option by clicking the “Delete” link. To edit the Form Element Option, click the “Edit” link and that will direct you to the Form Element Option Page.

ID	Text	Value	Options
408	Test	Test	Edit Delete



5.6.5 Add Form Element Attribute

To create a new Form Element Option for the Form Element, use the “Add Option” form. In this form, select an option in the “Attribute Type” field which indicated which attribute property will be added to the Form Element.. The “Value” field is a string of what the value for the new attribute should be. Once all the fields have been entered, click the “Add Attribute” button and this will create the Form Element Attribute into the system. You can now find the newly created Form Element Attribute in the Form Element Attribute List.

Add Attribute

Form Element Attribute

Attribute Type
the width

Value

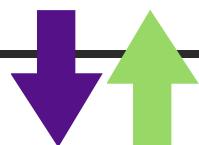
Add Attribute

5.6.6 List Form Element Attribute

On the Form Element Page, there is a list of all the Form Element Attributes associated to the current Form Element. The columns display several attributes. The ID column is the Form Element Attribute Id. The type column is the Form Element Attribute’s type.

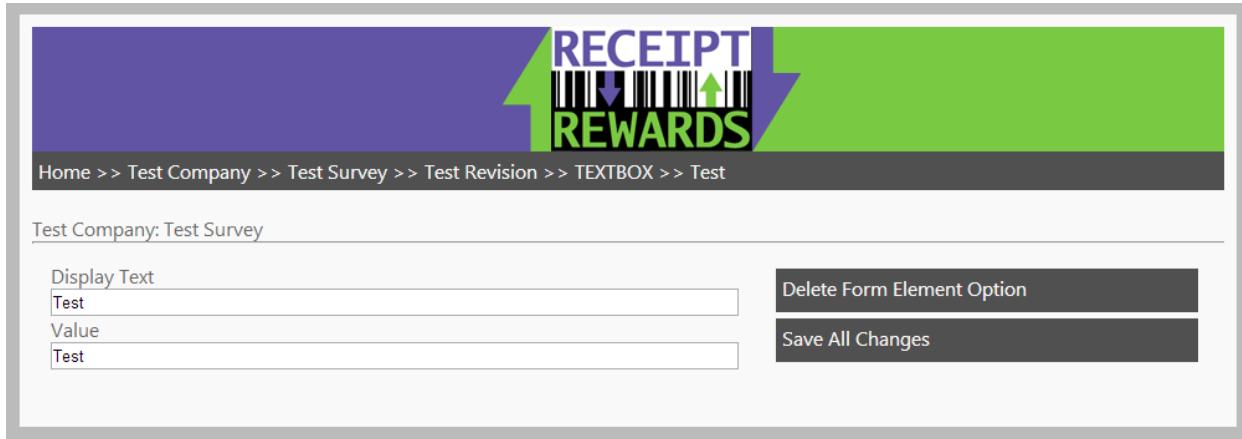
Type	Value	Options
------	-------	---------

The value column is the value of the Form Element Attribute. In the Options column, you can delete a Form Element Attribute by clicking the “Delete” link. To edit the Form Element Attribute, click the “Edit” link and that will direct you to the Form Element Attribute Page.



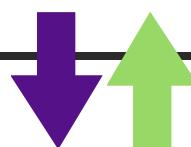
5.7 Form Element Option Page

The Form Element Option Page is the management of a single Form Element Option object in the database. The Form Element Option is a representation of a single option of a dropdown box, radio button, or checkbox in the satisfaction survey.



5.7.1 Edit Form Element Option

To change information of the current Form Element Option, use the “Edit Form Element Option” form. This form contains the “Display Text” field which should be a string of what the text should display as. The “Value” should be a string for what the hidden value will be stored. Once all the fields have been entered, click the “Save All Changes” button. The page will refresh and you can see the new changes.

A screenshot of a form titled "Edit Form Element Option". It has two input fields: "Display Text" (containing "Test") and "Value" (containing "Test"). To the right of the form is a large dark grey button labeled "Save All Changes".

5.7.2 Delete Form Element Option

To delete the current Form Element Option from the system, click the “Delete Form Element Option” button. This will delete the form element and redirect you back to the parent Form Element Page. Remember, this action cannot be undone, and all the information associated to form element option will also be removed.

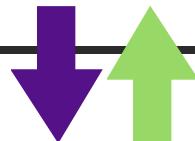
Delete Form Element Option

5.8 Command Element Page

The Command Element Page is the management of a single Command Element object in the database. The Command Element is a representation of the single action involved with filling out the satisfaction survey.

Type	Value	Options
XPATH	/html/body/div[@id='Test']	Edit Delete

Relation ID	Form Element ID	Options
72	138	Delete



5.8.1 Edit Command Element

To change information of the current Command Element, use the “Edit Command Element” form. This form contains the “Type” field which is a dropdown box. Choose the option for what type the Form Element should be displayed as. The “Test Data” should be a string of what the Web Driver will use for the test automation.

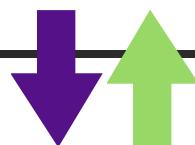
Save All Changes

Once all the fields have been entered, click the “Save All Changes” button. The page will refresh and you can see the new changes.

The form has a "Type" dropdown set to "Textbox" and a "Test Data" input field containing "Test".

5.8.2 Delete Command Element

To delete the current Command Element from the system, click the “Delete Command Element” button. This will delete the command element and redirect you back to the parent Question Page. Remember, this action cannot be undone, and all the information associated to command element will also be removed.



5.8.3 Add Command Element Identifier

To create a new Command Element Identifier for the Command Element, use the “Add Identification” form. In this form, select an option in the “Identification Type” field which indicates what the Command Element will be identified by the Web Driver. The “Value” field is a string of what the value for the new identifier is and will be looked for by the Web Driver. Once all the fields have been entered, click the “Add Identification” button and this will create the Command Element Identifier into the system. You can now find the newly created Command Element Identifier in the Command Element Identifier List.

Add Command Identification

Command Element Identifier

Identification Type

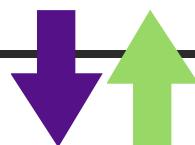
Value

5.8.4 List Command Element Identifier

On the Command Element Page, there is a list of all the Command Element Identifier associated to the current Command Element. The columns display several attributes.

Type	Value	Options
XPATH	/html/body/div[@id='Test']	Edit Delete

The ID column is the Command Element Identifier Id. The type column is the Command Element Identifier’s type. The value column is the value of the Command Element Identifier. In the options column, you can delete a Command Element Identifier by clicking the “Delete” link. To edit the Command Element Identifier, click the “Edit” link and that will direct you to the Command Element Identifier Page.



5.8.5 Add Command Form Element Relation

To create a new Form Element Relation for the Command Element, use the “Add Form Element Relation” form. In this form, you may select an option in the “Identification

Add Form Element Relation

Form Element Relation

Identification Type

137 ▾

or manually add id:

Add Relation

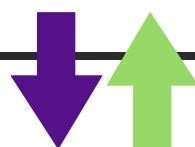
Type” field which contains the Form Element Ids of the current Question. Otherwise, you may input an integer into the “Manually Add ID” field of another element that isn’t in the current form. Once one of the options has been entered, click the “Add Relation” button and this will

create the Form Element Relation to the Command Element into the system. You can now find the newly created Form Element Relation in the Form Element Relation List.

5.8.6 List Command Form Element Relation

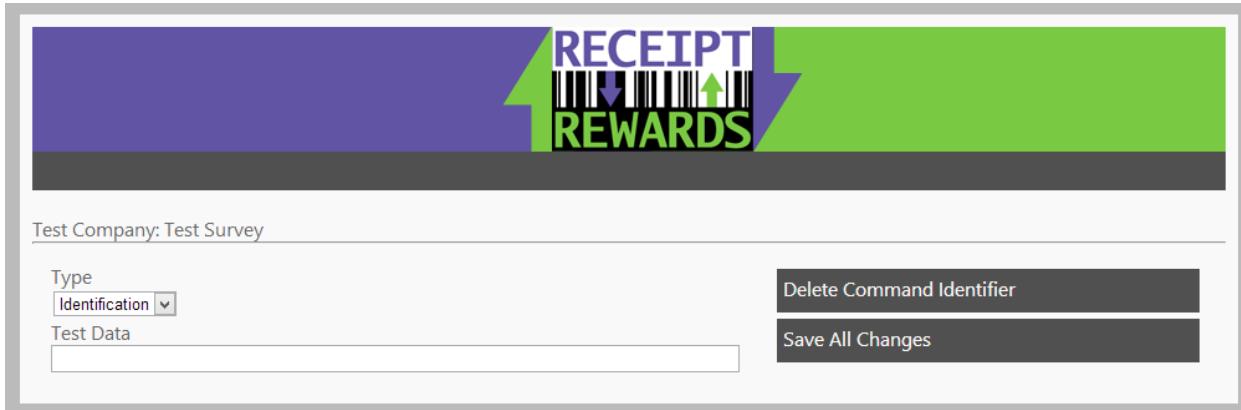
On the Command Element Page, there is a list of all the Form Element Relations associated to the current Command Element. The columns display several attributes. The ID column is the Form Element Relation Id. The form element id column is the Form Element Id that is being referenced. In the options column, you can delete a Form Element Relation by clicking the “Delete” link.

Relation ID	Form Element ID	Options
72	138	Delete



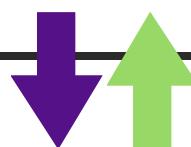
5.9 Command Element Identifier Page

The Command Element Identifier Page is the management of a single Command Element Identifier object in the database. The Command Element Identifier is a representation of the ID, Class, or Name of an HTML Element in the satisfaction survey.



5.9.1 Edit Command Element Identifier

To change information of the current Command Element Identifier, use the “Edit Command Element Identifier” form. This form contains the “Type” field which is a dropdown box. Choose the option represents how the Web Driver will identify the Web Element. The “Test Data” field is a string for what the Web Driver will use during the Automation Test. Once all the fields have been entered, click the “Save All Changes” button. The page will refresh and you can see the new changes.

A screenshot of a form titled "Edit Command Element Identifier". It includes a "Type" dropdown menu set to "Identification", a "Test Data" input field, and a "Save All Changes" button.

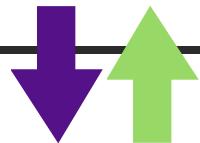
5.9.2 Delete Command Element Identifier

To delete the current Command Element Identifier from the system, click the “Delete Command Element Identifier” button.

This will delete the command element

Delete Command Identifier

identifier and redirect you back to the parent Command Element Page. Remember, this action cannot be undone, and all the information associated to command element identifier will also be removed.



7 Frequently Asked Questions

7.1 I made a change, but the webpage didn't update?

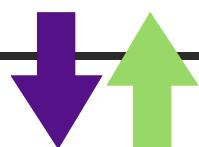
Sometimes after performing an insert, update or delete, the screen doesn't refresh the content and just displays what was cached.

If this happens, simply perform click on the breadcrumb navigation of the current page, and it will perform a hard refresh of the content. Your changes should now be displayed correctly.

Home >> TEST_CODE_TYPE

7.2 Are there any credentials need to use the Admin Site

Currently there are no credentials because the website is only hosted on local computers of the Administrators computers. However, if in the future the administrator site is put up on a public server, then yes, credentials will need to be added for security of the data.





PHONE APP USER MANUAL

8/23/2013

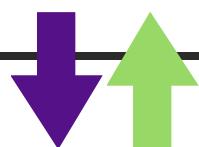
Graduate Capstone

RECEIPT

REWARDS

Table of Contents

Table of Contents	2
1 Introduction	3
1.1 Intended Audience	3
1.2 References	3
1.3 Revision History	3
2 Set Up	4
3 Navigation Actions	4
4 Booting the App	4
5 Splash Screen.....	5
6 Main Panorama.....	5
6.1 Quick Start	6
6.2 Navigation	6
7 Browsing Companies	7
8 Company Panorama.....	8
8.1 Company Information.....	8
8.2 Survey Information	8
9 Survey.....	9
9.1 Structure.....	9
9.2 Form Elements	9
9.2.1 Labels	9
9.2.2 Textbox and Textareas.....	10
9.2.3 Radio Button Group.....	11
9.2.4 Slider Bar	11
9.2.5 Dropdown Box	12
9.3 Submit Survey	13
10 Frequently Asked Questions	14
10.1 How to Find a Company	14



1 Introduction

The purpose of this document is to give a tutorial and walkthrough on how to navigate and use the Receipt Reward Application.

1.1 Intended Audience

This document is intended for any persons who wish to utilize the Phone Application and gain a better understanding of how the screens work. This document is written for people with any degree of familiarity with the application.

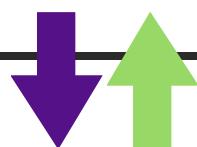
1.2 References

No references were used.

1.3 Revision History

Managing the change history of this document will occur in this table.

Name	Date	Reason For Change	Version
Andy Bottom	08/13/2013	Created the introduction and parts of the document.	0.1
Andy Bottom	08/20/2013	Finished complete manual	1.0



2 Set Up

The following assumption is made that the application is already installed on the desired device. If this is not the case, please contact Administrators of Receipt Rewards and they can direct you on where to find and install the application on your device.

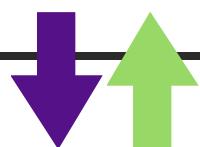
3 Navigation Actions

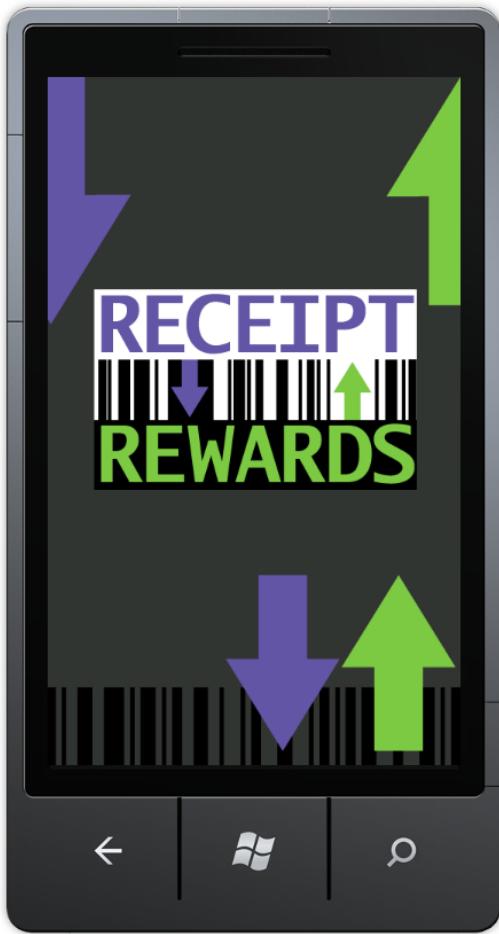
Prior to navigating the application, there are several controls and actions that are important to go over.

- Click: This action of clicking on a link is used to navigate from one page to another.
- Swipe: On panorama view, pages are located next to one another. To navigate to a new page in this type of view, simply swipe your finger and drag in the direction you want. This will slide the panorama over and display the page.
- Back Button: The physical back button on the phone is used to navigate backwards to previous pages.

4 Booting the App

The app will be found on the under the list of apps. If the app is pinned to your start menu, you may find it there as well. To start the app, simply click on the Receipt Reward Application. This starts the boot process and the loading screen should follow immediately after clicking the application.





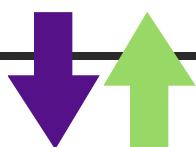
5 Splash Screen

As mentioned in the previous section, after selecting the Receipt Reward application, the application will start up. During the booting process, the splash screen should become visible. This screen has no functionality or interactive aspects to it. Its only function is to provide the logo of the application. This screen allows for the user to see that the process of the app opening is currently active and that the application will be loaded and ready for use momentarily.

This screen loads very fast and the application should be ready within 1 to 5 seconds. If the splash screen is still being shown and the time elapse exceeds 15 seconds, this may indicate another problem. The best solution for this occurrence is to restart the phone.

6 Main Panorama

The first view that will be displayed to the user is the main panorama view. Since this is a panorama view, it contains multiple pages next to each other. By doing this, the most important functionality can be easily accessible and used.

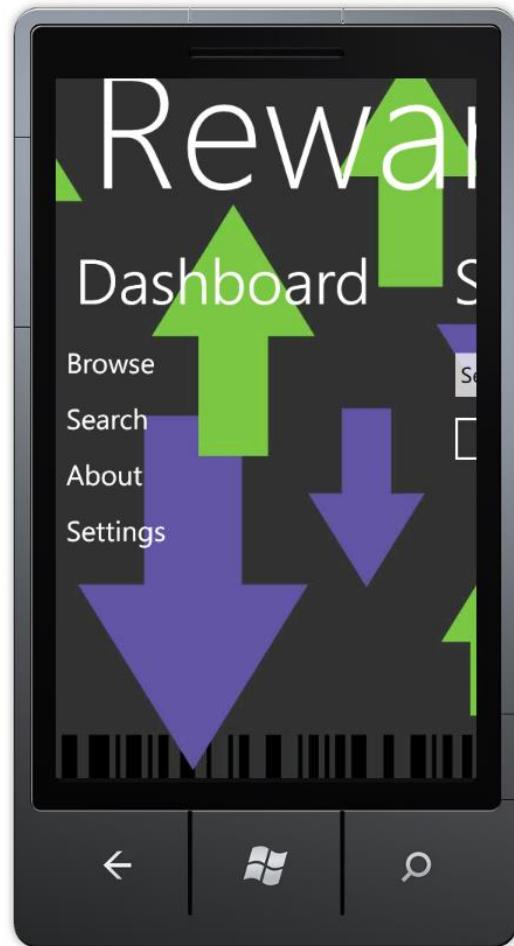




6.1 Quick Start

The quick start menu brings the functionality that you want directly to you in the fastest way possible.

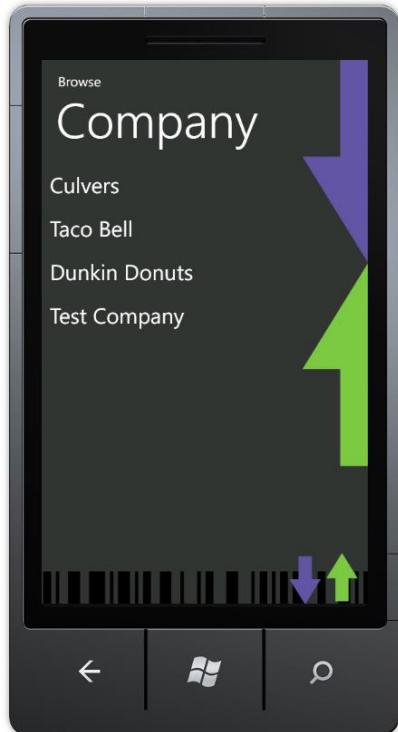
To start a survey, simply type in the company name you would like to fill out a survey for. The results will then display. Next, simply click the company link and you will be sent to the survey page.



6.2 Navigation

The navigation menu has a list of links to the main functionality of the application.

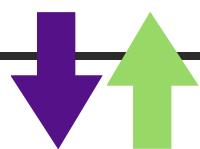




7 Browsing Companies

The ability to find and learn about new companies is very important aspect to the user. To achieve this, use the Brose Company function. This page is simple a complete list of all the companies that are in the system.

Each company in the list has been verified to offer the satisfaction surveys and rewards. To learn more about the company, simple click on the company name, and that will bring you to the Company Information View.



8 Company Panorama

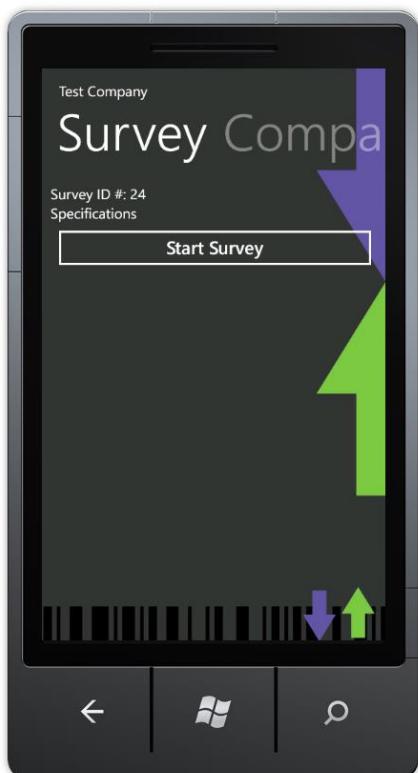
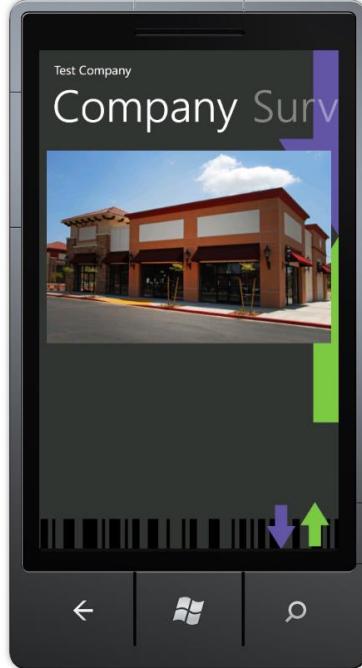
The Company Panorama view contains the information about a company and its corresponding survey.

8.1 Company Information

The company information page contains details about company. The page contains an elegant picture of the company as well as helpful fact about the company for those who may not be familiar.

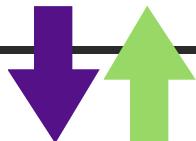
Some of the categories and facts that you will learn include:

- Business Type
- Locations
- Website
- Founding Year



8.2 Survey Information

The survey information page contains details about the survey. Such information includes: Redeemable Code or Coupon, the estimated time the survey will take, etc... If you want to take the survey, then all you have to do is click the "Start Survey" button, and you will be directed to the actual survey.



9 Survey

9.1 Structure

The survey structure is very self-intuitive. When you start the survey, you will be immediately given the first question. None of the, are you ready to take the survey nonsense. We get straight to the point to get you done faster and get the reward in your hand sooner

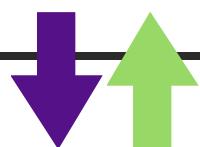
In addition, we don't use any backwards navigation through the survey. The back button on the phone will take you back to the company page. But be careful, as we currently do not save your answers for you. So, please use with caution.

9.2 Form Elements

In order to make the process of filling out the application, we've focused on using the most intuitive form elements. By doing this, we make it as easy as possible for input.

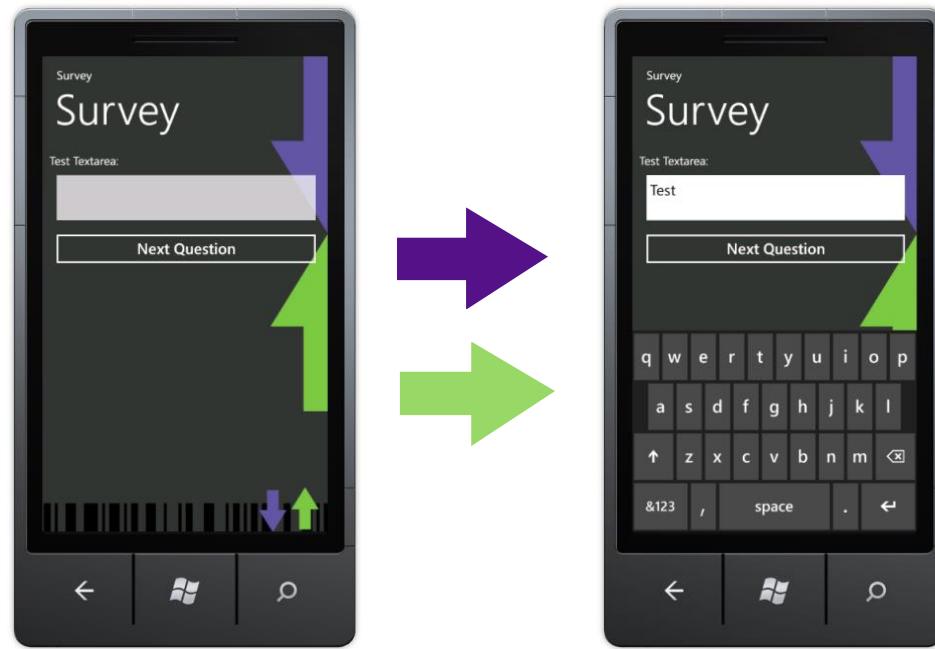
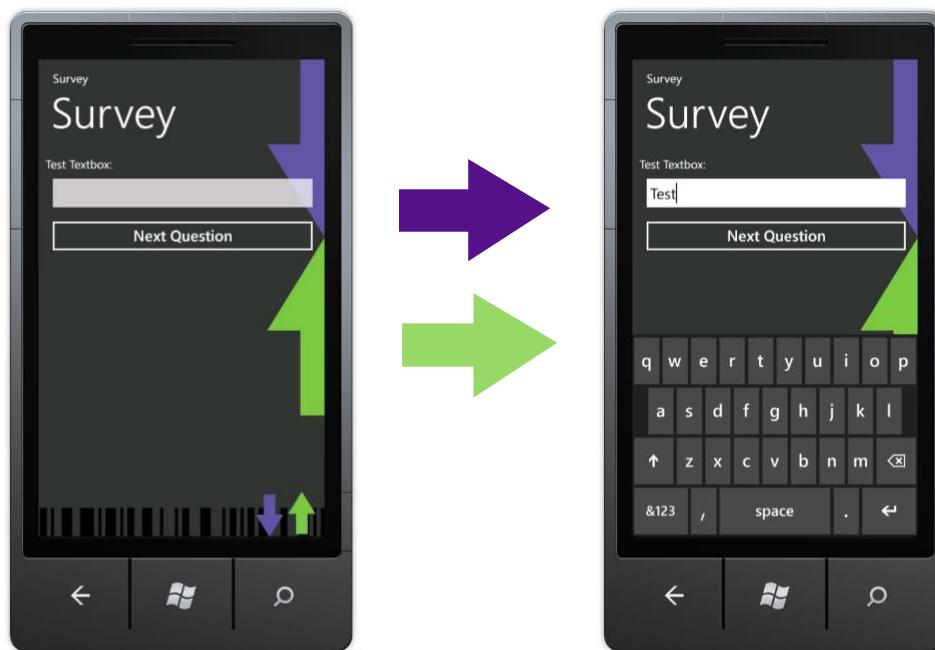
9.2.1 Labels

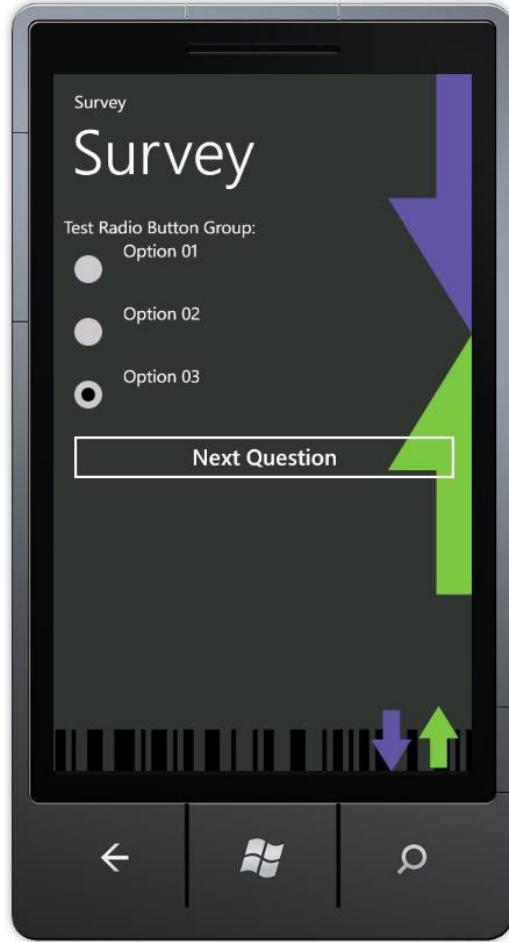
Labels are a useful element that is used to display the question to you.



9.2.2 Textbox and Textareas

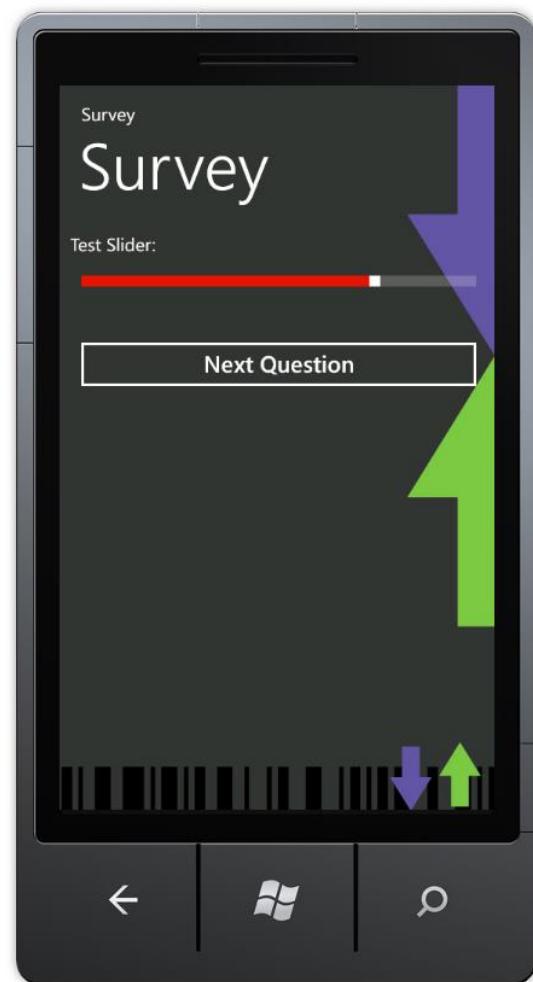
The Textbox and Textarea controls are a great way to get input from you. Typically, these controls are used on questions that are intended to be open ended or fact of some kind. You can see below examples of both input elements and how they display on the field.





9.2.3 Radio Button Group

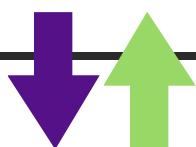
Radio Button Groups are the rounded buttons that indicate that only one value of the following can be chosen. Radio buttons are often used with questions which have predefined answers that are acceptable.



9.2.4 Slider Bar

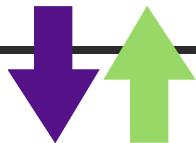
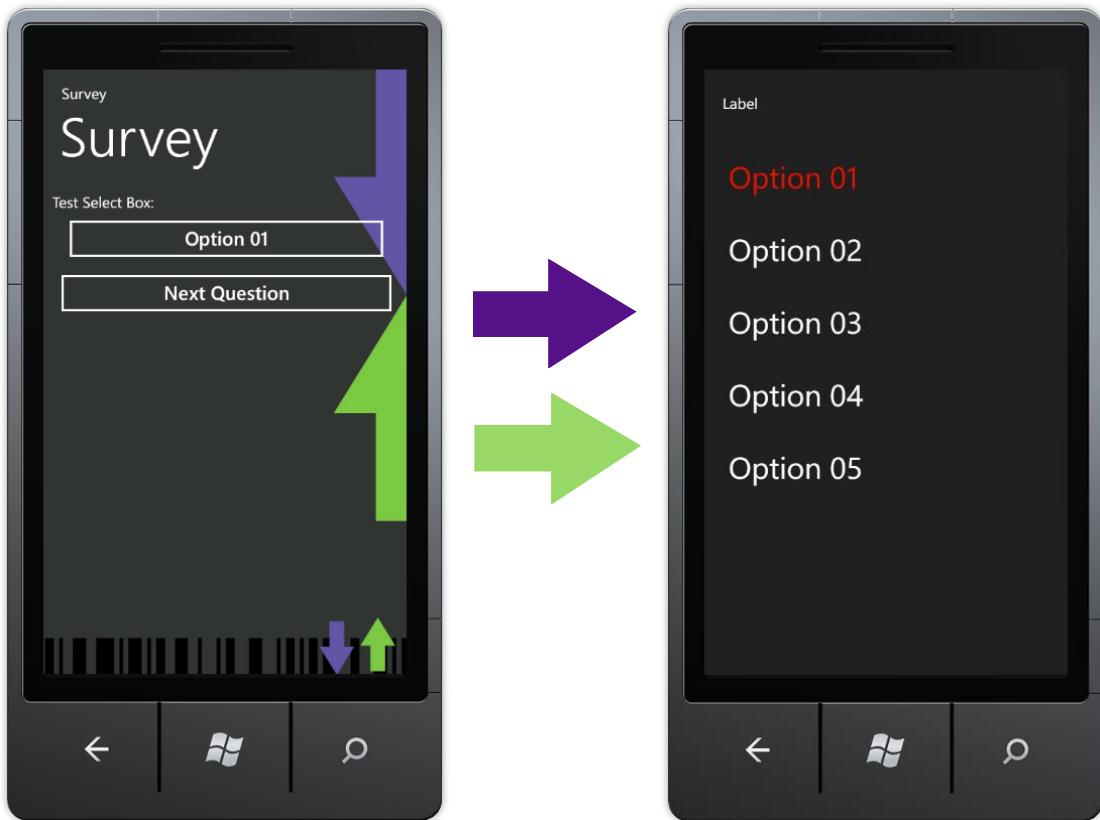
The sliding bar is a horizontal bar which users can click and slide to change the value.

Typically, sliding bars will be used on questions involving a rating scale. In direction, the left side of the bar is negative and the right side of the bar is positive. Neutral is in the middle.



9.2.5 Dropdown Box

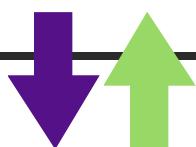
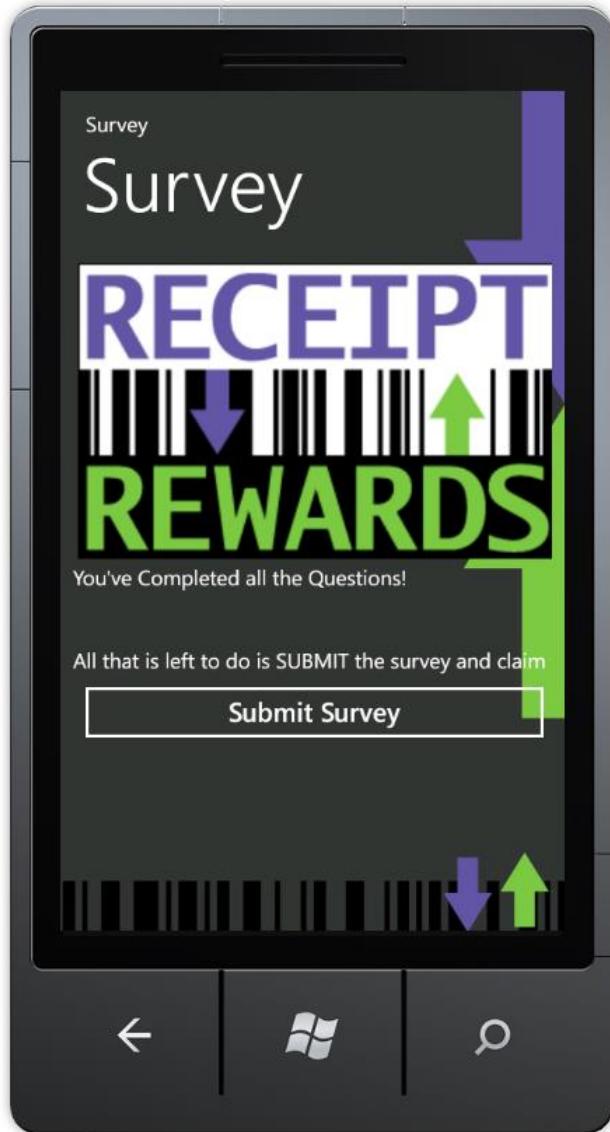
The dropdown list is an alternative way to answer from a list of options. To choose an option, simply click the drop down box. The dropdown will expand to a full screen list of all the options. To selection an option, simply click the desired option and that value will be displayed in the dropdown box.



9.3 Submit Survey

When you have reached the end of the survey, you will reach the “Submit Survey” page. This page informs you that you have successfully finished the survey and are ready to submit your answers. To submit, click the “Submit Survey” button.

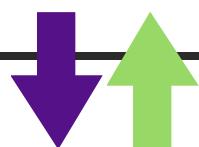
Upon submitting that application will run the results and return back your reward information. It may take a few moments to retrieve your reward.

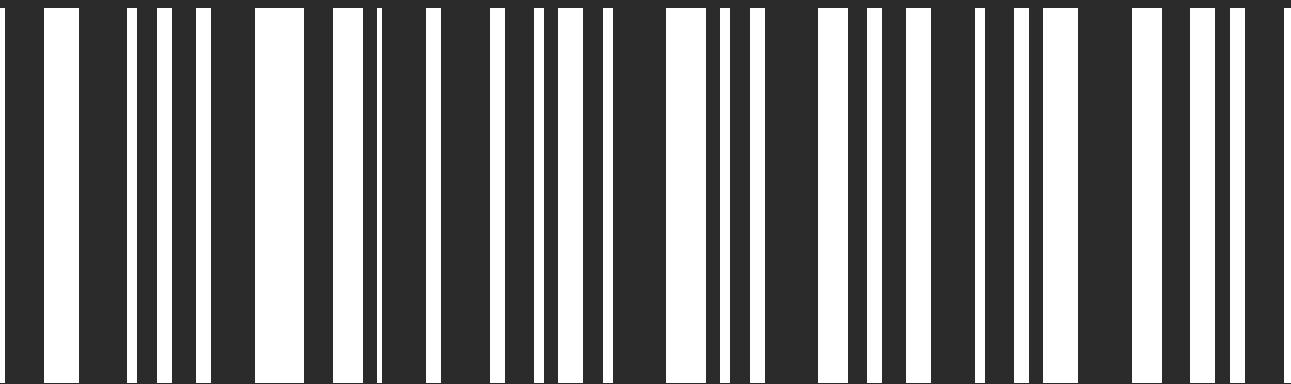


10 Frequently Asked Questions

10.1 How to Find a Company

You can discover companies in a variety of ways. If you want to find a specific company, use the search and search by the company name. If you want to discover other companies, use the browse feature. If the company that you are looking for has a survey, but is not in the system, the contact Receipt Rewards, and we will do our best to add that company to our system.





ACKNOWLEDGEMENT AND LICENSE DOCUMENT

8/23/2013

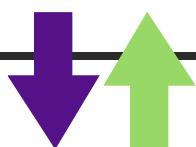
Graduate Capstone

RECEIPT

REWARDS

Table of Contents

1 Introduction.....	3
1.1 Intended Audience.....	3
1.2 References.....	3
1.3 Revision History	3
2 General Acknowledgements	4
3 Third Party Acknowledgements.....	5
3.1 Java Proxy Pattern - Caching proxy Project.....	6
3.1.1 Credit	6
3.1.2 Services.....	6
3.2 BTI360	6
3.2.1 Credit	6
3.2.2 Services.....	6
3.3 MVVM Cross	7
3.3.1 Credit	7
3.3.2 Services.....	7
3.4 MVVM Cross	8
3.4.1 Credit	8
3.4.2 License Info	8
3.5 JSON.net	10
3.5.1 Credit	10
3.5.2 License	10
3.6 jQuery	11
3.6.1 Credit	11
3.6.2 License	11
3.7 Selenium.....	12
3.7.1 Credit	12
3.7.2 License	12
3.8 Xamarin	13
3.8.1 Credit	13
3.8.2 License	13
3.9 Tek271 Reverse Proxy Server	14
3.9.1 Credit	14
3.9.2 License Info	14
3.10 Android OS	15
3.10.1 Credit.....	15
3.10.2 License Info	15
3.11 Drawer Sample.....	21
3.11.1 Credit.....	21
3.11.2 License Info	21



1 Introduction

This document is to provide acknowledgement of individuals who had a part in the project. Both to those who gave input or simply supported me through the entire process. Also, mentions to third-party software that was used or inspired any part of the project.

1.1 Intended Audience

This document is intended for anyone, specifically persons seeking lists of the credits of the project.

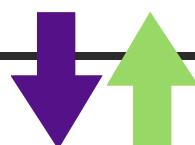
1.2 References

- http://webhelp.esri.com/arcims/9.3/java/arcgis93_acknowledgements.pdf
- <http://ves.vitalimages.com/vital/help/en/pdf/AcknowledgementsCopyrights.pdf>
- <http://www.intermedia.net/legal/Intermedia-SecuriSync-Open-Source-Notice-File.pdf>

1.3 Revision History

Managing the change history of this document will occur in this table.

Name	Date	Reason For Change	Version
Andy Bottom	05/28/2013	Started the format of the document and couple credits	0.1
Andy Bottom	08/22/2013	Finished the acknowledgement document	1.0



2 General Acknowledgements

My Capstone project has been a large effort and would not have been possible without the support and guidance of many people.

I would like to express my deepest gratitude to my advisor, Dr. Chenglie Hu for his excellent guidance throughout my college career and put me on a path of success to not only complete my Bachelor's Degree, but to have the opportunity to enroll in Graduate School and earn my Master's Degree.

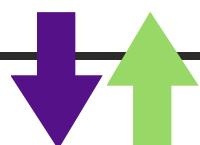
In addition to Dr. Hu, I would also like to thank the rest of the Computer Science faculty, especially Dr. Gerald Isaacs, Michael Konemann, and Crispian Sievenpiper. Through their instructions, I have not only improved and learned new skills, but have gained a profound appreciation for the field of computer science.

I would like to say a special thanks all my closest and dearest friends, (you know who you are.) Whether it is brainstorming and bouncing ideas around, listening to my ramblings, or simply enjoying our company. I am indeed blest to have such true friends.

I would like to thank my dearest Sonya. You have truly been a blessing to me. With your love and care, I have grown and became a better person thanks to you. I've enjoyed our past years, going through undergraduate and graduate school together and I look forward to all the years to come.

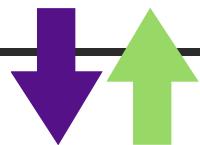
Thanks to my two Grandmas, Mathilda Bottom and Kathleen Zblewski. You have given me much support throughout my life and am truly thankful to have you in my life.

Lastly, to my family: my parents Ernie Bottom and Julie Bottom, and my sister Abbie Bottom. Through my years, you have instilled morals in me, and to always do what is right. I've also learned to do my best and I am truly blest to have such a loving and caring family. Thank you.



3 Third Party Acknowledgements

In the following section, information about code or tutorials that were used or inspired code in my project are acknowledged. In addition, and open source libraries, code, or platforms that were also utilized or were a dependency of the system are also acknowledged. This is to give full credit to those companies and individuals for all their work that provide great assistance.



3.1 Java Proxy Pattern – Caching proxy Project

I used this post and code as a starting point to implement the caching ability into the reverse proxy server.

3.1.1 Credit

Name	Nico Giangregorio
E-mail	nicogorio@gmail.com
Website	http://nicogiangregorio.blogspot.com/

3.1.2 Services

- Blog
- Tutorial
- Code Snippets

3.2 BTI360

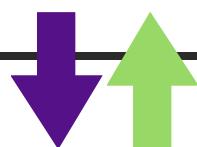
I had a lot of trouble setting up RESTful Services using Netbeans. Their screencast and project code had a major part in me finally getting my RESTful Web Services up and running.

3.2.1 Credit

Company Name	BTI360
E-mail	solutions@bti360.com
Website	http://www.bti360.com/

3.2.2 Services

- Tutorials
- Screencasts
- Code Snippets



3.3 MVVM Cross

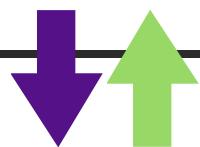
When I was looking into how to develop an application for the phone, I had to look for a lot of helps and examples. Stuart's very active support in the Xamarin community was an immensely helpful resource. He was also the individual to mention the idea to use Model-View-ViewModel as a design pattern. In addition to providing a large amount of tutorials and screencasts, I picked up on a lot and allowed me to begin development even faster. Altogether, Stuart was a huge help to me during the development of the phone application.

3.3.1 Credit

Name	Stuart Lodge
Twitter	@slodge
Website	http://slodge.blogspot.com/

3.3.2 Services

- Tutorials
- Screencasts
- Code Snippets
- Open Source Code
- Forums Help



3.4 MVVM Cross

The MVVM Cross is an open source library that I utilized to obtain functionality that allowed me to have the PCL send and receive data from the Restful Services.

3.4.1 Credit

Name	Stuart Lodge
Github Website	https://github.com/slodge/MvvmCross

3.4.2 License Info

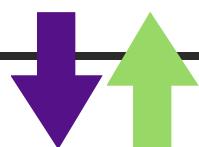
Excerpt from Readme File

MvvmCross v3

This project provides a cross-platform mvvm mobile development framework built on top of:

- Silverlight for WP7, WP8
- Mono for Android (or Xamarin.Android)
- MonoTouch for iOS (or Xamarin.iOS)
- the WinRT XAML framework for Windows 8 Store apps.
- WPF
- Mono for Mac (or Xamarin.Mac)

This project makes extensive use of Portable Class Libraries to provide maintainable cross platform C# native applications.



Microsoft Public License (MS-PL)

This license governs use of the accompanying software. If you use the software, you accept this license. If you do not accept the license, do not use the software.

1. Definitions

The terms "reproduce," "reproduction," "derivative works," and "distribution" have the same meaning here as under U.S. copyright law.

A "contribution" is the original software, or any additions or changes to the software.

A "contributor" is any person that distributes its contribution under this license.

"Licensed patents" are a contributor's patent claims that read directly on its contribution.

2. Grant of Rights

(A) Copyright Grant- Subject to the terms of this license, including the license conditions and limitations in section 3, each contributor grants you a non-exclusive, worldwide, royalty-free copyright license to reproduce its contribution, prepare derivative works of its contribution, and distribute its contribution or any derivative works that you create.

(B) Patent Grant- Subject to the terms of this license, including the license conditions and limitations in section 3, each contributor grants you a non-exclusive, worldwide, royalty-free license under its licensed patents to make, have made, use, sell, offer for sale, import, and/or otherwise dispose of its contribution in the software or derivative works of the contribution in the software.

3. Conditions and Limitations

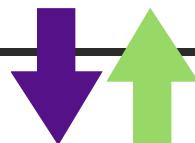
(A) No Trademark License- This license does not grant you rights to use any contributors' name, logo, or trademarks.

(B) If you bring a patent claim against any contributor over patents that you claim are infringed by the software, your patent license from such contributor to the software ends automatically.

(C) If you distribute any portion of the software, you must retain all copyright, patent, trademark, and attribution notices that are present in the software.

(D) If you distribute any portion of the software in source code form, you may do so only under this license by including a complete copy of this license with your distribution. If you distribute any portion of the software in compiled or object code form, you may only do so under a license that complies with this license.

(E) The software is licensed "as-is." You bear the risk of using it. The contributors give no express warranties, guarantees or conditions. You may have additional consumer rights under your local laws which this license cannot change. To the extent permitted under your local laws, the contributors exclude the implied warranties of merchantability, fitness for a particular purpose and non-infringement.



3.5 JSON.net

The library JSON.net was utilized with the PCL to more easily convert objects to and from JSON format so that it could then be used for the web services.

3.5.1 Credit

Website	http://json.codeplex.com/
---------	---

3.5.2 License

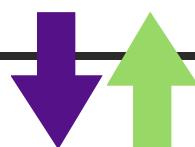
The MIT License (MIT)

Copyright (c) 2007 James Newton-King

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.



3.6 jQuery

jQuery is a JavaScript library that I utilized to improve the user experience of the Web Admin CMS. Specifically the collapsing dropdown menus are implemented via jQuery.

3.6.1 Credit

Company	The jQuery Foundation
Website	https://jquery.org/

3.6.2 License

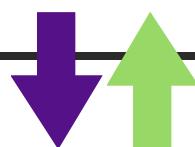
The MIT License (MIT)

Copyright 2013 jQuery Foundation and other contributors
<http://jquery.com/>

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.



3.7 Selenium

Selenium is the library used to perform the Web Driver Automated process.

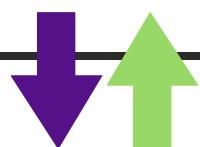
3.7.1 Credit

Company	Selenium Developers Group
Website	http://www.seleniumhq.org/

3.7.2 License

Excerpt from Readme File

All Selenium projects are licensed under the Apache 2.0 License.



3.8 Xamarin

Xamarin is the framework used to develop a cross platform application for Android and Windows Mobile Phones.

3.8.1 Credit

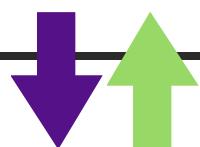
Company	Xamarin Inc.
Website	http://xamarin.com/

3.8.2 License

Excerpt from Readme File

Commercial Mono Licensing

Mono is an open source project. The Mono virtual machine is licensed under the [LGPLv2 license](#). This license places a number of restrictions on the use of Mono.



3.9 Tek271 Reverse Proxy Server

This project was used as a major starting point to create the reverse proxy server. Most of the original code is still implemented, just implemented a better caching ability for the pages.

3.9.1 Credit

Name	Abdul Habra
E-mail	ahabra@yahoo.com
Website	http://www.tek271.com/

3.9.2 License Info

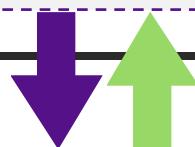
GNU Lesser General Public License (LGPL)

Excerpt from Readme File

This program uses GNU LGPL.
Copyright © Abdul Habra 2011
ahabra@yahoo.com

Excerpt from Code Snippets

```
/*
 * This file is part of Tek271 Reverse Proxy Server.
 *
 * Tek271 Reverse Proxy Server is free software: you can redistribute it and/or
 * modify
 *   * it under the terms of the GNU Lesser General Public License as published by
 *   * the Free Software Foundation, either version 3 of the License, or
 *   * (at your option) any later version.
 *
 * Tek271 Reverse Proxy Server is distributed in the hope that it will be useful,
 * but WITHOUT ANY WARRANTY; without even the implied warranty of
 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
 * GNU Lesser General Public License for more details.
 *
 * You should have received a copy of the GNU Lesser General Public License
 * along with Tek271 Reverse Proxy Server. If not, see http://www.gnu.org/licenses/
 */
```



3.10 Android OS

The Android Operating System is an open source project. I wrote code that was developed for the android os.

3.10.1 Credit

Company Name	Google
Website	http://www.android.com/

3.10.2 License Info

Copyright (c) 2005-2008, The Android Open Source Project

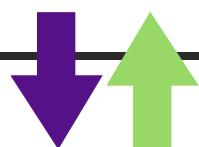
Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License.

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.

See the License for the specific language governing permissions and limitations under the License.

Apache License
Version 2.0, January 2004
<http://www.apache.org/licenses/>

(continued)



TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

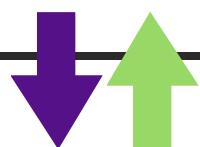
"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

(continued)



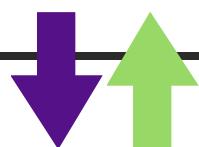
"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

(continued)

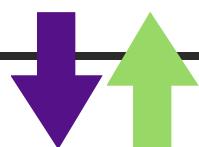


3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

- (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
- (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
- (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(continued)



(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

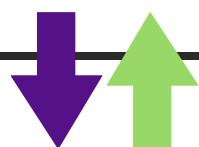
You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions.

Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

(continued)

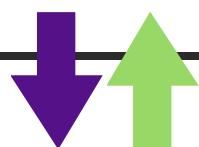


7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS



3.11 Drawer Sample

The drawer sample was used as a reference point that I used to build off of to implement a sliding drawer in the Android Application. Because the drawer was not native in the versions of android that I was using, this open source drawer allowed me to have the key element and still target older versions of android os.

3.11.1 Credit

Name	Tomasz Cielecki
E-mail	tomasz@ostebaronen.dk
Website	http://ostebaronen.dk/
GitHub	https://github.com/Cheesebaron

3.11.2 License Info

DrawerSample

A couple of samples showing the Drawer Navigation pattern and the SlidingPane Layout recently added in the Support packages.

The [Drawer Navigation sample](<http://developer.android.com/training/implementing-navigation/nav-drawer.html>) is ported from the original documentation.

The SlidingPane Layout sample is ported from [this sample repository](<https://github.com/mastro/android-support-library-archive/blob/master/samples/Support4Demos/src/com/example/android/support/v4/widget/SlidingPaneLayoutActivity.java>).

License

Both samples are licensed under the Apache 2.0 license.

