

**CZ2006**

**Software Engineering**

**Final Report**

**twoCube™**

|  |  |
| --- | --- |
| **Name:** | **Matric No.:** |
| **Chen Yuhui, Tiffany** | **U1123037H** |
| **Cheok Jia De** | **U1121561B** |
| **June Quak Ren Feng** | **U1121476E** |
| **Khok Hong Jing** | **U1122104L** |
| **Lim Guan** | **U1122655H** |
| **Peh Weileng** | **U1123164B** |
| **Sri Hartati** | **U1123011H** |
| **Wesley Djingga** | **U1121491E** |
| **Xu Ai** | **U1120855F** |

**Date of Submission: 5 November 2012**

**Contents Page**

1. Project Team Information 4

2. Refined Software Requirement Specification 4

2.1. Product Description 4

2.1.1. Product Vision 4

2.1.2. Business Requirements 5

2.1.3. Stakeholders and Users 5

2.1.4. Project Scope 5

2.1.5. Assumptions 6

2.1.6. Constraints 6

2.2. Functional Requirements 6

2.3. Data Requirements 8

2.4. Non-functional requirements 11

2.4.1. Compatibility 11

2.4.2. User interface 11

2.4.3. Security 11

2.4.4. Performance 11

2.4.5. Backup and Recovery 12

2.4.6. Reliability 12

2.4.7. System Maintenance 12

2.5. Interface Requirements 12

2.5.1. User Interfaces 12

2.5.2. Hardware interfaces 13

2.5.3. Software interfaces 13

2.6. Use Case Models 14

2.6.1. Use Case Diagram 14

2.6.2. Use Case Descriptions 15

2.7. Glossary 31

2.8. References 31

2.9. Revision History 32

3. Use Case/ Activity Diagrams 33

4. Analytical Model – Class Diagrams 34

5. Design Model – Sequence Diagrams 35

5.1. Account Management 35

5.1.1. Create member account 35

5.1.2. Login 35

5.1.3. Update member account 36

5.1.4. Update password 36

5.2. Survey Management 37

5.2.1. Create Survey 37

5.2.2. Create Survey Questions 37

5.2.3. Create Survey Options 38

5.2.4. View Survey 38

5.3. Survey Report Management 39

5.3.1. Generate Survey Report 39

5.3.2. View Survey Report 40

5.3.3. Export Survey Report 40

5.4. Respondents Actions 41

5.4.1. Submit Survey 41

5.5. Ground Data Logging 41

5.5.1. Data Logging 41

6. Testing 42

6.1. Unit Testing 42

6.2. Functional Testing 42

7. Discussion 63

8. Deriving analytical design and models 63

9. Going about performing UI design 63

10. Difficulties encountered and solutions applied 64

11. Work breakdown structure 64

1. Project Team Information

|  |  |  |  |
| --- | --- | --- | --- |
| **Team Name:** | twoCube™ | | |
| **Project Title:** | NTUSurvey | | |
| **Team Website URL:** | [http://twocube1.elasticbeanstalk.com](http://twocube1.elasticbeanstalk.com/) | | |
| **Project Description:** | [http://twocube1.elasticbeanstalk.com](http://twocube1.elasticbeanstalk.com/) | | |
|  | | | |
|  | **Name** | **Email Address** | **Contact No.** |
| **Team Leader** | Cheok Jia De | [jdcheok1@e.ntu.edu.sg](mailto:jdcheok1@e.ntu.edu.sg) | 92390354 |
| **Member 1** | Wesley Djingga | [wdjingga1@e.ntu.edu.sg](mailto:wdjingga1@e.ntu.edu.sg) | 92234537 |
| **Member 2** | June Quak Ren Feng | [jquak1@e.ntu.edu.sg](mailto:jquak1@e.ntu.edu.sg) | 97368902 |
| **Member 3** | Peh Weileng | [wlpeh001@e.ntu.edu.sg](mailto:wlpeh001@e.ntu.edu.sg) | 94593932 |
| **Member 4** | Sri Hartati | [hsri001@e.ntu.edu.sg](mailto:hsri001@e.ntu.edu.sg) | 81127957 |
| **Member 5** | Lim Guan | [c110050@e.ntu.edu.sg](mailto:c110050@e.ntu.edu.sg) | 92230282 |
| **Member 6** | Xu Ai | [xuai0001@e.ntu.edu.sg](mailto:xuai0001@e.ntu.edu.sg) | 93362832 |
| **Member 7** | Chen Yuhui, Tiffany | [chen0791@e.ntu.edu.sg](mailto:chen0791@e.ntu.edu.sg) | 92266801 |
| **Member 8** | Khok Hong Jing | [hjkhok1@e.ntu.edu.sg](mailto:hjkhok1@e.ntu.edu.sg) | 96964903 |

1. Refined Software Requirement Specification
   1. Product Description
      1. Product Vision

twoCube**™** offers a one-stop solution to all survey needs. Getting feedback and responses from the people are the vital activities every organization will conduct. twoCube**™** survey allows users to create surveys, distribute them and view statistics report on the responses.

twoCube**™** is a new player in the survey industry. It strives to assist individuals and organizations by creating a platform to gather and analyze data. It also strives to achieve to make its web survey as simple and convenient for all the users.

Not only measuring customer satisfaction, a survey aids organizations in learning customers’ preferences and uncover new ideas. In addition, individuals are also able to determine a business idea’s feasibility through surveys. Online surveys offer the advantage of convenience and time saving as well. Over the years, online survey websites has come to play in the process of business improvement.

twoCube**™ offers** 3 simple steps:

* Creating a survey
* Collecting data
* Generating reports
  + 1. Business Requirements

The first version of the survey system must be available within three months of the development project launch.

The survey system aims to reduce the manpower in distributing and collecting survey responses thus saving time and efforts.

* + 1. Stakeholders and Users

The targeted users are users (members) and survey respondents.

Users are those who designs and creates the survey whom twoCube**™** named them as members as they have to be registered with twoCube**™** in order to enjoy the features provided.

Survey respondents are people who respond to the survey by answering the survey questions.

Survey respondents can answer the survey questions using personal computers, laptops, and smart phones with web browsers that can easily be accessible at anytime, anywhere.

As for the members, they are able to do the following features with similar equipment as the respondents. They are able to create member account to register with twoCube™, login to the website, manage members account, create surveys, generate survey report, and exporting them to a separate file.

* + 1. Project Scope

The scope is to develop a survey system which provide a platform for users to publish survey question online and have respondents to respond to the question.

The users are able to design survey and customize each question and publish the survey by distributing the link for respondent’s responses. At any point in time, users can view the survey’s responses and also generate reports.

By launching the system, it increases the efficiency of survey data collection and reduces manual workload which can be very tedious.

Smartphone users can also download our android app, <twocube mobile> to access all of the features available on our site as well.

* + 1. Assumptions
* Users must be able to have web browser software such as Chorme, FireFox in order to access twoCube website.
* Users are required to have basic IT knowledge so that they are able to navigate the website.
* Once the survey questions are published, users are not allowed to edit, add or delete questions.
  + 1. Constraints
* System is unable to save the survey questions created automatically if the user have leave the page in the half way of creating,when the user came back to the system, he has to restart the procedures.
  1. Functional Requirements

**Overview**

**For members:**

* Create Member account
* Login
* Update Member account
* View Member account
* Update password
* Create survey
* Create survey questions
* Edit, delete and reorder survey questions
* Create survey options
* View survey statistics
* Generate survey report
* View survey report
* Export survey report
* Data logging

**For respondents:**

* Submit survey
* View survey

**Create Member account**

* Allow users to input their details
* System must save these details into database

**Login**

* Allow users to input username and password or alternatively use Facebook account to login
* For system login, username and password will be compared with the username and password in database. System save username session
* For Facebook login, system will compare Facebook ID with the current Facebook ID exist in the database. If exist, it will load all account data in the database. If it is not, system will get several data to be input into the database such as Facebook ID, first name, last name and email

**Update Member account**

* System display user’s information
* Users can change other details
* System will save changes by updating the database

**View Member account**

* Allow users to view the details of their account
* System displays details of users account like username, email, etc.

**Update password**

* Allow users to change the password of their account.
* System will save the changes by updating the database

**Create survey**

* Allow users to input the title of each question
* Allow users to select survey questions type, options, question’s status (whether is compulsory question)
* Allow users to publish the survey questions
* Allow users to edit, delete and reorder survey questions
* System will save new survey into the database

**Create survey questions**

* Allow users to create different types of surveys questions
* The questions can be multiple choice, text, scale and etc.
* Allows users to use images as question options
* Allow users to make the question a compulsory question

**Create survey options**

* Allow users to create survey options based on the questions
* Allow users to choose the range of scale for scale questions
* Allow users to set minimum and maximum value for slider questions
* Allow users to create multiple options for multiple choice questions

**View survey**

* Allow users to view the created survey(s)

**Generate survey report**

* Allow users to collect survey respondent data
* System will generate survey report

**View survey report**

* Allow users to view survey report which consist of statistics of question options and how long the users take to answer a question.
* System will display generated report

**Export survey report**

* System will export selected survey’s responses in Spreadsheet format

**Data logging**

* System will record respondent’s IP address to ensure one person only does the survey once.
* System will also record the time taken to answer each question.

**Submit survey**

* Respondent will answer each survey question and submit it.
  1. Data Requirements

|  |  |  |  |
| --- | --- | --- | --- |
| **Member** | | | |
| **Field Name** | **Data Type** | **Description** | **Possible Values** |
| Id | int4(10) | Primary Key |  |
| memberFirstName | varchar(255) | Member’s first name |  |
| memberLastName | varchar(255) | Member’s last name |  |
| userName | varchar(255) | Member’s login name |  |
| memberPassword | varchar(255) | Member’s login password | Min: 6 chars |
| memberAge | Int4(2) | Member’s age |  |
| dateOfBirthday | timestamp | Member’s birth date |  |
| memberLocation | varchar(255) | Member’s location |  |
| memberEmail | varchar(255 | Member’s email |  |
| memberFBID | Varchar(255) | Member’s Facebook ID |  |
| memberQuestion | varchar(255 | Secret question for forgetting Member ID or password |  |
| memberAnswer | varchar(255 | Secret answer for forgetting Member ID or password |  |

|  |  |  |  |
| --- | --- | --- | --- |
| **Survey** | | | |
| **Field Name** | **Data Type** | **Description** | **Possible Values** |
| Id | int4(10) | Primary Key |  |
| surveyTitle | varchar(255) | Title of the survey |  |
| surveyDescription | varchar(255) | Description of survey |  |
| surveyStatus | bool | Is survey open or closed | True : open  False : closed |
| surveyCreated | timestamp | Date the survey was created |  |
| surveyStartDate | timestamp | Date the survey opens |  |
| surveyEndDate | timestamp | Date the survey ends |  |
| Member\_Id | int4(10) | Foreign Key |  |

|  |  |  |  |
| --- | --- | --- | --- |
| **SurveyQuestion** | | | |
| **Field Name** | **Data Type** | **Description** | **Possible Values** |
| Id | int4(10) | Primary Key |  |
| surveyQuestionType | int4(10)) | Type of survey question | 0 : Radio button  1 : Checkbox  2 : Scale Slider  3 : Numerical Input  4 : Date Input  5 : Scale Radio Button  6 : Text  7 : Textarea  8 : Signature  9 : Image  10 : Dropdown List |
| surveyQuestionIsCompulsory | bool | Is answering the question compulsory or not? | True : Question compulsory  False : Not compulsory |
| surveyQuestionTitle | varchar(255) | Title of survey question |  |
| survey\_Id | int4(10)) | Foreign Key |  |

|  |  |  |  |
| --- | --- | --- | --- |
| **SurveyQuestionOption** | | | |
| **Field Name** | **Data Type** | **Description** | **Possible Values** |
| Id | int4(10) | Primary Key |  |
| surveyQuestionOptionType | int4(10)) | Type of survey question option. Follow the question type or a text input | 0 : According to Qn type  1 : Text input |
| surveyQuestionOptionTitle | varchar(255) | Title of option | True : Question compulsory  False : Not compulsory |
| surveyQuestionOptionTitleType | int4(10)) | Text, Link or Img | 0 : Text  1 : Link  2 : Image |
| surveyQuestionOptionRange | int4(10)) | Range from 0 to this number | Integer larger than 0 |
| surveyQuestionOptionMinText | varchar(255) | If survey question is scale type, the text at the min |  |
| surveyQuestionOptionMaxText | varchar(255) | If survey question is scale type, the text at the max |  |
| surveyQuestion\_Id | int4(10)) | Foreign Key |  |

|  |  |  |  |
| --- | --- | --- | --- |
| **SurveyQuestionResponse** | | | |
| **Field Name** | **Data Type** | **Description** | **Possible Values** |
| Id | int4(10) | Primary Key |  |
| responseIsAnwered | bool |  | True : Answered  False : Unanswered |
| responseType | int4(10) | Check boxes, radio buttons, sliders will be integer type. Date, and text will be String. Question options that allow text response will be both. In this case Integer will be the option they chose, String will be the answer. | 0 : Integer  1 : String  2 : Both |
| responseIntegerValue | int4(10) | Value if response is an integer type |  |
| responseStringValue | varchar(255) | Value if response is a String type |  |
| respondent\_Id | int4(10 | Foreign Key |  |
| surveyQuestion\_Id | int4(10) | Foreign Key |  |

* 1. Non-functional requirements
     1. Compatibility

The website must be compatible with all HTML5 compliant browsers.

The mobile application must be compatible with all android 2.1+ devices.

* + 1. User interface

The user interface must be as familiar as possible to users who have used other web applications and Windows desktop applications. E.g., we will follow the UI guidelines for naming menus, buttons, and dialog boxes whenever possible.

* + 1. Security

Access shall be controlled with usernames and passwords

Access to the database will be done via web services.

* + 1. Performance

The system must be up and running with a 99% uptime.

It must support at least 100 users replying to surveys concurrently without any noticeable lag.

* + 1. Backup and Recovery

There must be a backup server and database to prevent service interruption or loss of data when the main server and database are down.

Downtime must not last more than 30sec when switching from main server to the backup server in case of a breakdown.

* + 1. Reliability

The whole survey system must achieve a 99% success rate. I.e. downtime should not be more than 1% of its total operating time.

System review shall take place monthly. Any lack in performance or reliability shall be addressed and improved on after each review.

* + 1. System Maintenance

Maintenance of the system shall be conducted weekly. Maintenance shall be conducted during off-peak hours e.g. between 12am - 6am.

* 1. Interface Requirements
     1. User Interfaces

User interfaces describes how users interact and engage with a specific page on a website. The goal of user interface is to make user’s interaction as simple and interactive as possible.

Overview of the screen must be clear. For instance, break the screen up into sections such that at the top section it will display the logo and navigation links, on the middle of the page it will be the contents of what the users are expected to see and lastly at the bottom of the page it will be footer.

User must also be able to navigate from one page to another page without any problem or broken links. Also, the links should be obvious to where the users would be directed to (it should not direct user to login page when user clicks on registration page). Error message should also be displayed whenever there is an error faced by the users and provide solutions on how the users can solve the problem or error.

Requirement of user interface is that users need familiar with the handling basic UI components such as radio buttons, dropdown list, and checkboxes.

It provides a user-friendly graphic interface for any users who come to twoCubeTM website, performing tasks such as:

* Registering for an account

Registration should be able to create new users in the database successfully with user name, password and email address

* Login

This should require a username and password to successfully login to the website

* Create survey

This should allow the users to create survey without any problem

* Viewing survey results

This should enables user to generate analytic reports on the survey created

In addition, the overview of the website must be consistent in all aspects. For example, the logo should be at the top left hand corners on all pages. The colour and size of the font must also be consistent such that it does not confuse and frustrate the users (i.e **bold** only when the content is important) to reduce ambiguity.

* + 1. Hardware interfaces

Hardware devices such as PCs, notebooks, tablets, mobile phone devices are supported by the website.

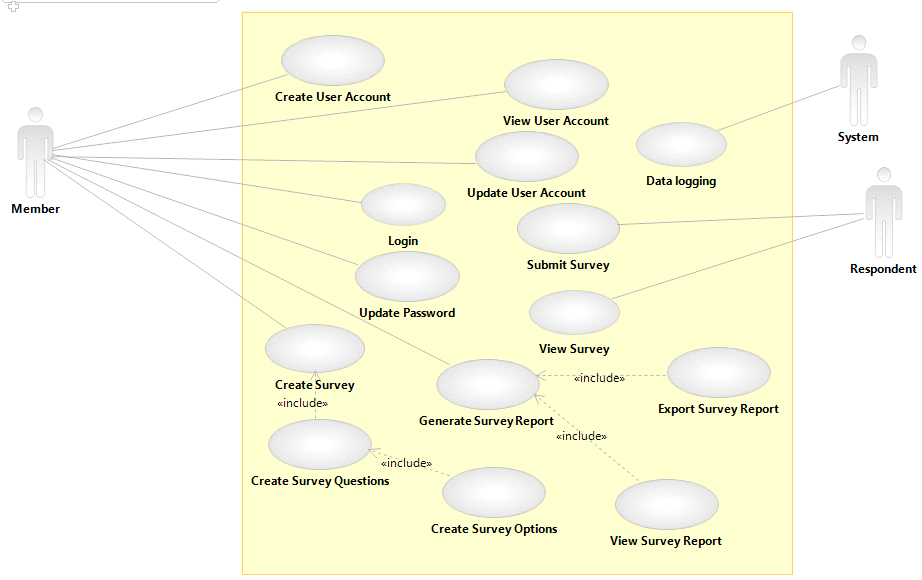
It also requires Android 2.2 above for the mobile application.

* + 1. Software interfaces

twoCubeTM website enables users to interact with the server and get access to the Internet.

Some of the software requirements would be:

* Html5 browser
* Datebase: postgresql
* Application: c# webservices
* Web Server: IIS7 running on a scalable cloud
* Flash
  1. Use Case Models
     1. Use Case Diagram



* + 1. Use Case Descriptions
* Use Case List

|  |  |
| --- | --- |
| ***Primary Actor*** | ***Use Cases*** |
| Member | Create Member Account |
| Member | Login |
| Member | View Member Account |
| Member | Update Member Account |
| Member | Update Password |
| Member | Create Survey |
| Member | Create Survey Questions |
| Member | Create Survey Options |
| Respondent | View Survey |
| Member | Generate Survey Report |
| Member | View Survey Report |
| Member | Export Survey Report |
| Respondent | Submit Survey |
| System | Data logging |

<use case diagrams>

* 1. Glossary
* **Actor** – a type of agent that interacts with a product
* **Assumption** – a state of affairs taken for granted in product development
* **Business Requirement** – a statement of a client or a development organization goal that a product must meet
* **Class** – an abstraction of a set of objects with common operations and attributes
* **Client** – someone outside a development organization with an interest in a product, such as customer or a user
* **Constraint** – any factor that limits developers and particularly one that limits design solutions; in UML, a property that a model element must satisfy or a relationship between model elements that must be maintained
* **Customer** - someone who pays for a product
* **Database** – a computerized collection of records
* **Data requirement** – a statement that certain data must be input to, output form, or stored in a product
* **Functional- requirement** – a statement of how a software product must map program inputs to program outputs; also called a behavioral requirement
* **High Fidelity (abbr. Hi-Fi)** – a term use to describe a high quality of an entity
* **ID** – identification number which uniquely identify different products.
* **Implementation** – the creation of executable artifacts for delivery to customers in a software product; the relation between an abstract
* **Interface** – a boundary across which two entities communicate.
* **Low Fidelity (abbr. Lo-Fi)** – a term use to describe a lower quality of an entity
* **Microsoft.NET Framework** – is a software framework developed by Microsoft that runs primarily on Microsoft Windows
* **NHibernate** – is an object-relational mapping solution for the Microsoft .NET platform: it provides a framework for mapping an object-oriented domain model to a traditional relational database
* **Non-functional requirement** – a statement that a software product must have certain properties; also called a non-behavioral requirement
* **Object-oriented programming (abbr. OOP)** – is a programming paradigm using "objects" – usually instances of a class – consisting of data fields and methods together with their interactions – to design applications and computer programs.
* **Object-relational mapping (abbr. ORM)** – is a programming technique for converting data between incompatible type systems in object-oriented programming languages
* **Product** – Refers to the twoCube system.
* **Product vision** – a general description of a product’s purpose and form
* **Product scope** – the work to be done in a project
* **Respondents** – refers to the people who do the survey
* **Responses** – refers to the answers the respondents enter when doing the survey.
* **RESTful** – derived from REpresentational State Transfer (REST) which is a style of software architecture for distributed systems such as the World Wide Web; conforming to the REST constraints is referred to as RESTful.
* **Service Oriented Architecture** – a set of principles and methodologies for designing and developing software in the form of interoperable services
* **Software requirement** – a statement that a software product must have certain feature, function, capability, or property
* **Software requirements specification (abbr. SRS)** – a complete description and documentation of the twoCube system. It includes a description of all the interactions the users will have with the software.
* **Structured Query Language (abbr. SQL) -** special-purpose programming language designed for managing data
* **Survey** – a sampling, or partial collection, of facts, figures, or opinions taken and used to approximate or indicate what a complete collection and analysis might reveal
* **Stakeholder** – anyone affected by a product or involved in or influencing its development
* **System –** Refers to the twoCube system
* **Use case model** – a use case diagram together with use case descriptions for each use case in the diagram
* **Use case diagram** – a UML notation representing a prodcut’s use case and actors involved in each use case
* **User Interface (abbr. UI)** – refers to the visual component of the system by which the user and a computer system interact

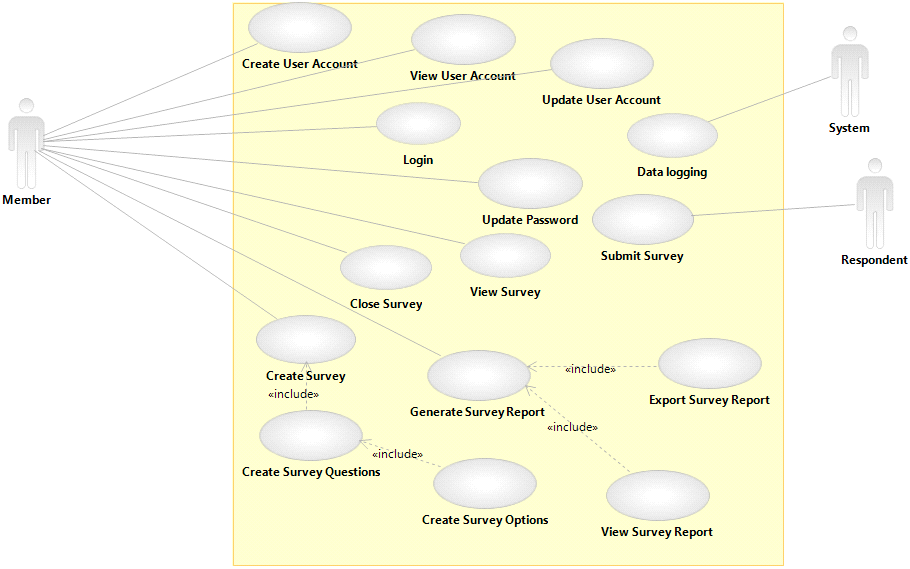
* 1. References

|  |  |  |  |
| --- | --- | --- | --- |
| **Document No.** | **Document Title** | **Date** | **Author** |
| 1 | CZ2002 Object-Oriented Designing and Programming | 4th Sept 2012 | Tan Kheng Leong |
| 2 | Sample of Software Requirements Specifications | 5th August 2010 | Ravi Bandakkanavar |
| 3 | Time Monitoring Tool Software Requirements Specifications | 1st Jan 2001 | Martin Robillard |
| 4 | Software Requirements Specifications Elevator System Controller | - | Alex Kalaidjian |
| 5 | Post Grass Software Requirements Specifications | 15th March 2002 | Prof. Antoniol Giulio |
| 6 | Software Requirements Specifications for Online Shopping System | 2nd April 2012 | Naresh Prajapati |

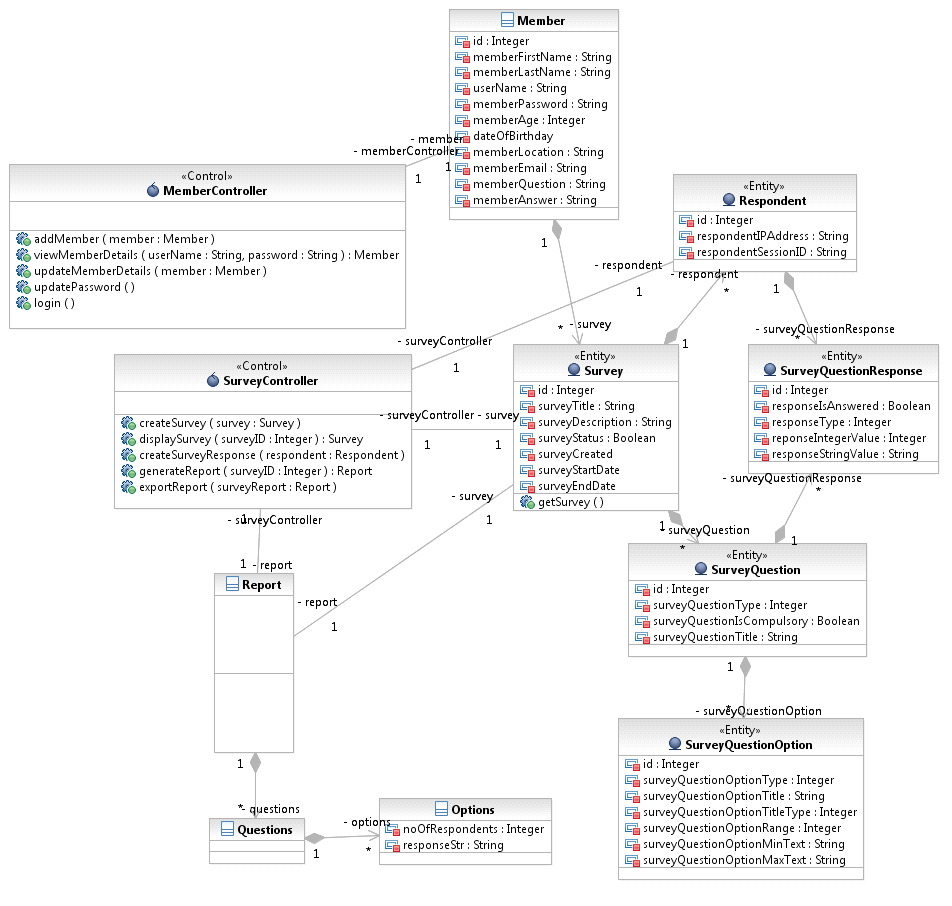
* 1. Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Version** | **Date** | **Name** | **Description** |
| Ver. 1.1 | 31/08/2012 | SRS Version 1.1. | Section 1: Project Team Information added  Section 2.1: Refined Software Requirements added  Section 2.2: Functional Requirements added  Section 2.4: Non-Functional Requirements added  Section 10.1: WBS added |
| Ver. 1.2 | 07/09/2012 | SRS Version 1.2. | Section 2.6: Use Case Model added  Section 3: Use Case/Activity Diagrams added |
| Ver. 1.3 | 21/09/2012 | SRS Version 1.3. | Section 2.1: Refined Software Requirements edited  Section 4: Analytical Model – Class Diagrams added  Section 5: Design Model – Sequence Diagrams added |
| Ver. 1.3 | 12/10/2012 | SRS Version 1.3. | Section 2.6: Use Case Model added  Section 3: Use Case/Activity Diagrams added  Section 4: Analytical Model – Class Diagrams edited  Section 5: Design Model – Sequence Diagrams edited |
| Ver. 1.4 | 03/11/2012 | SRS Version 1.4. | Section 6: Testing added |
|  |  |  |  |

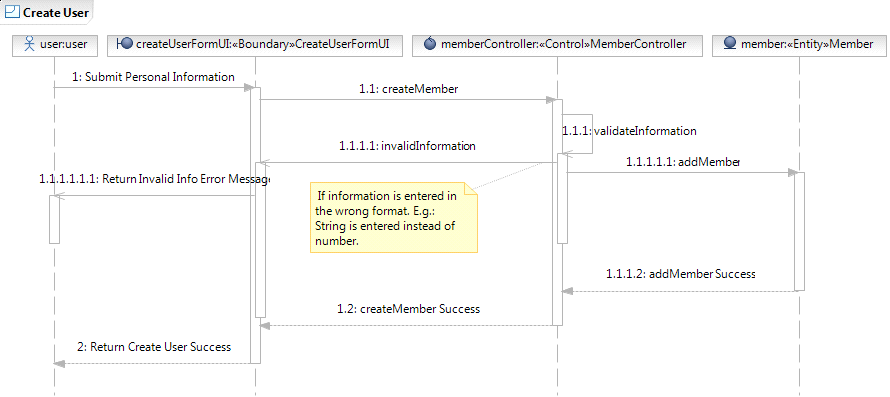
1. Use Case/ Activity Diagrams



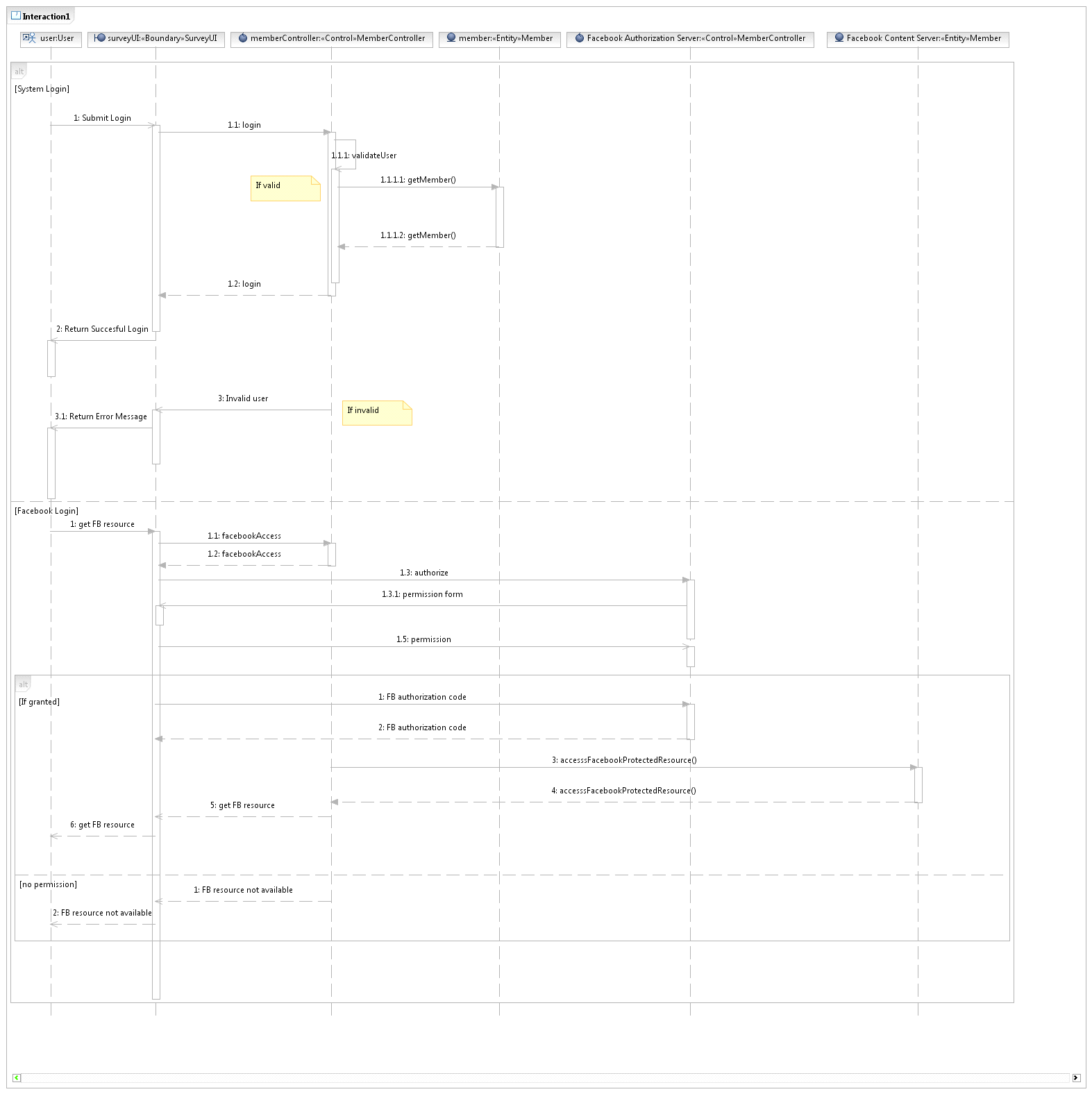
1. Analytical Model – Class Diagrams



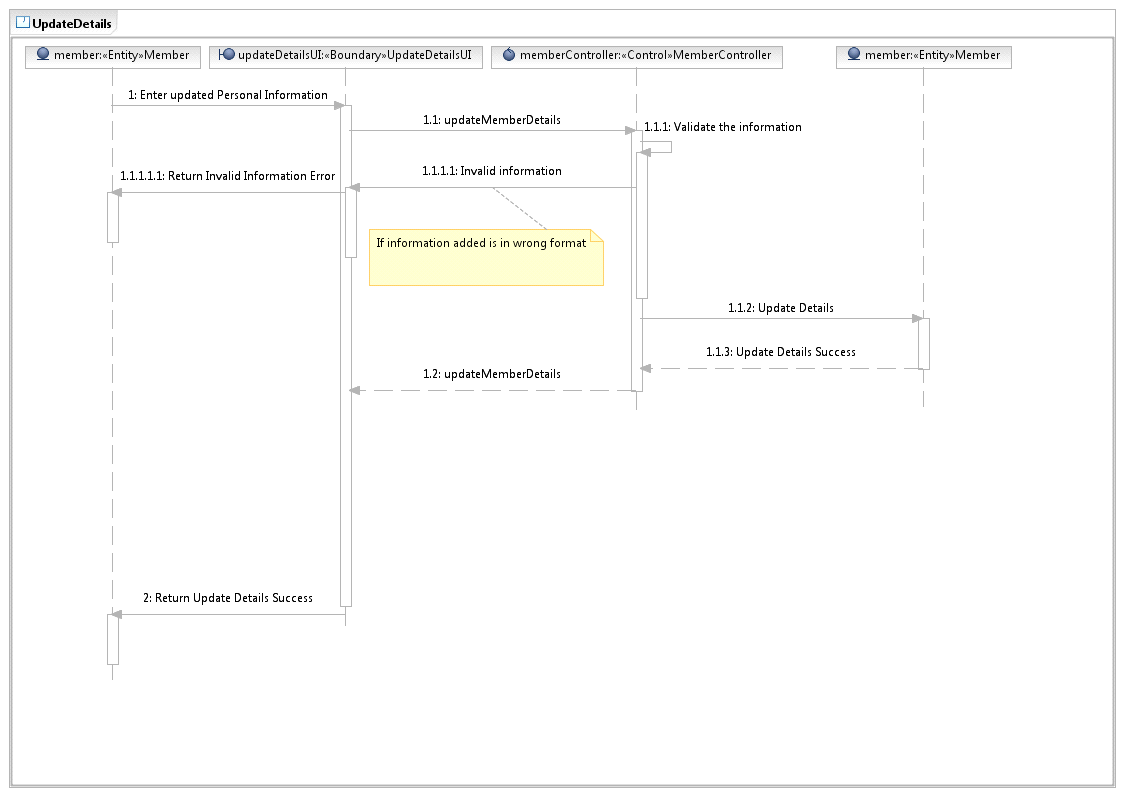
1. Design Model – Sequence Diagrams
   1. Account Management
      1. Create member account



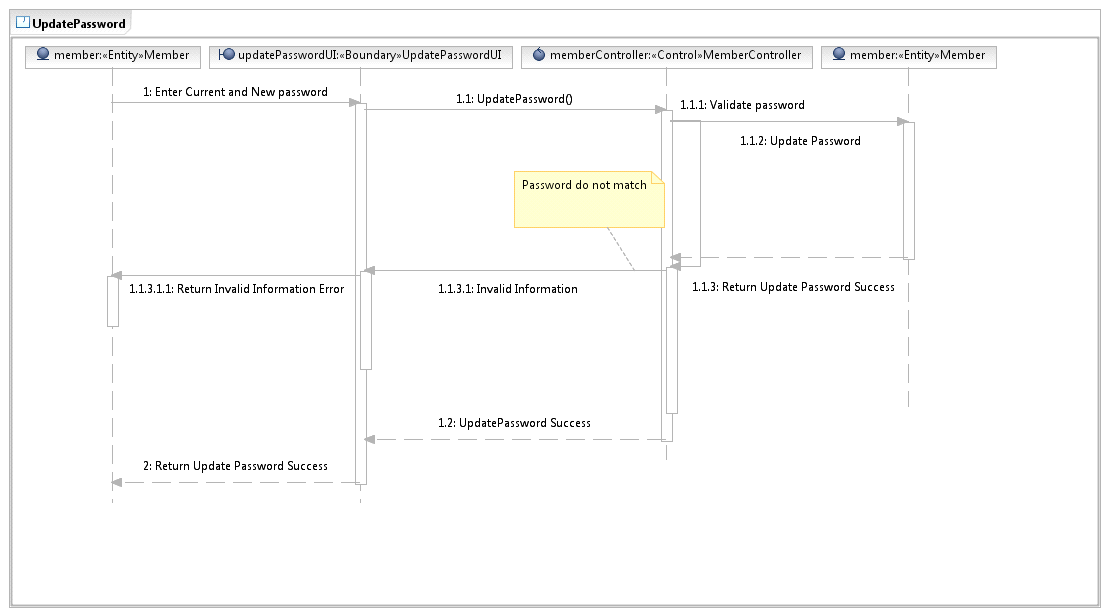
* + 1. Login



* + 1. Update member account



* + 1. Update password



* 1. Survey Management
     1. Create Survey



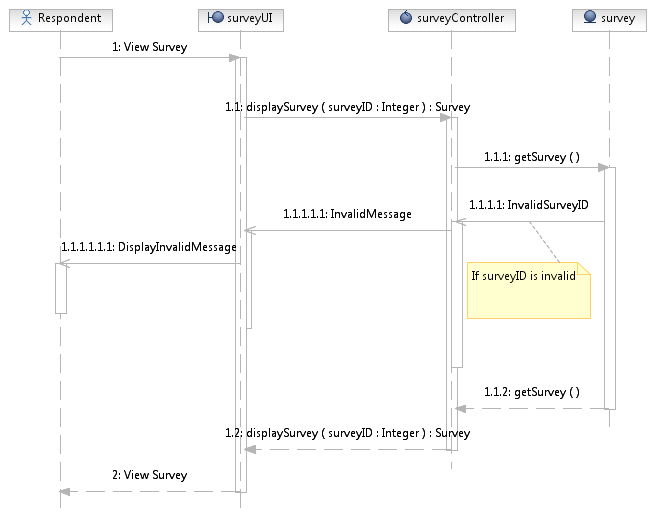
* + 1. Create Survey Questions



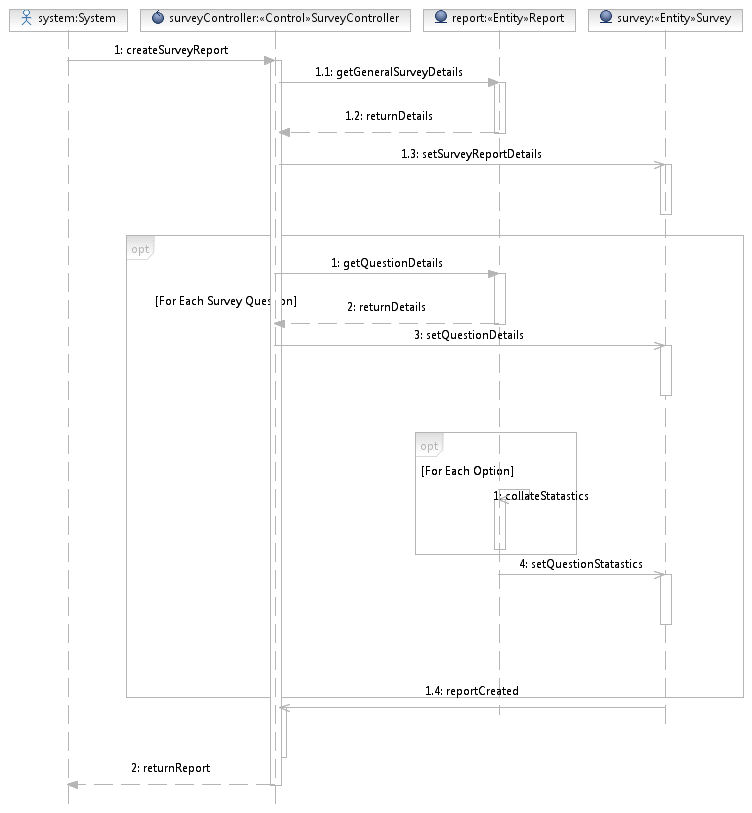
* + 1. Create Survey Options



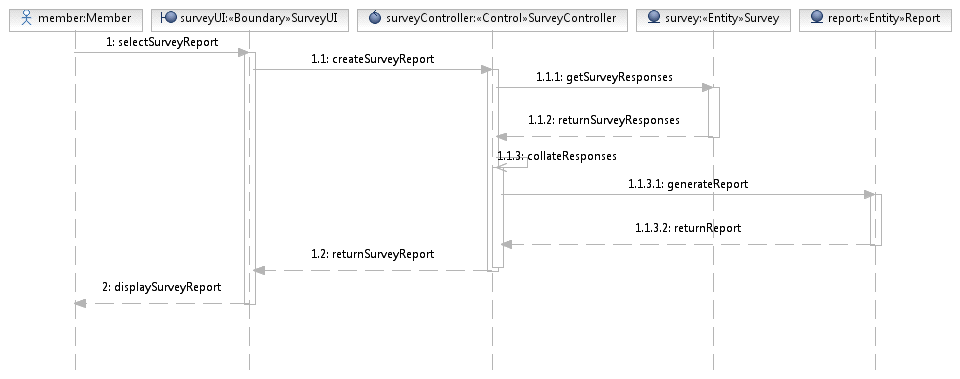
* + 1. View Survey



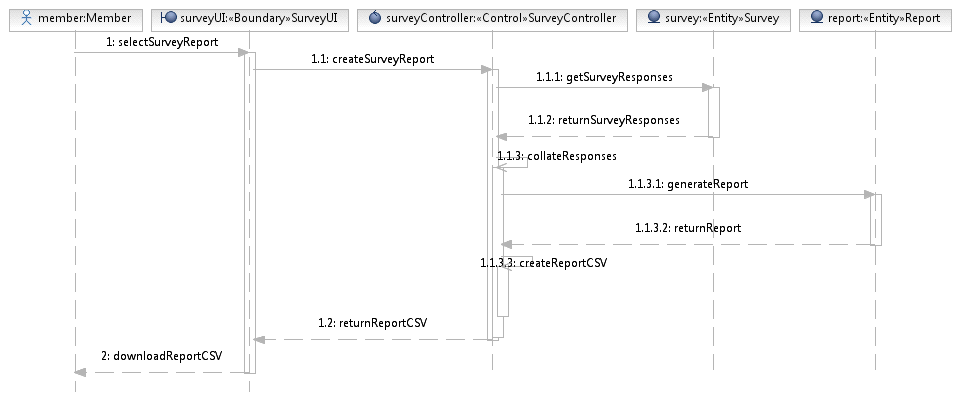
* 1. Survey Report Management
     1. Generate Survey Report



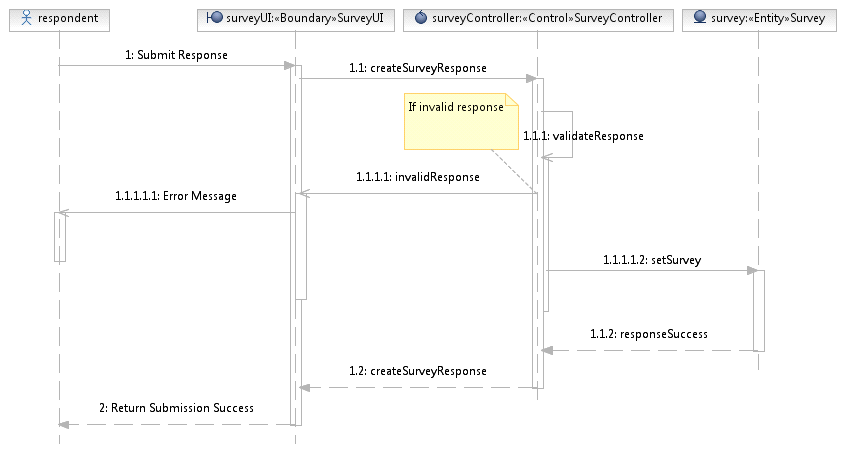
* + 1. View Survey Report



* + 1. Export Survey Report



* 1. Respondents Actions
     1. Submit Survey



* 1. Ground Data Logging
     1. Data Logging

1. Testing
   1. Unit Testing
   2. Functional Testing

**createSurvey.html**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Functional Requirements | Execution Details | System response | Remarks |
| Survey Form | User must be able to enter title and description for the survey form | Create survey form by clicking on “Form Properties” tab and edit in the text box for title and description | System instantly displays the changes while user updates the textbox. | Success |
|  | User must be able to add survey questions. | Create survey questions by choosing the question from “Add a Field” tab | System instantly displays question types and details correspondingly to the question added on the right side of the panel | Success |
|  | User must be able to reorder survey questions. | Reorder survey questions by clicking on the question and drag to desired positon. | System instantly displays the new order of survey questions. | Success |
|  | User must be able to edit survey questions. | Edit survey infomation by clicking on the corresponding field. | System instantly displays the changes while user updates the field. | Success |
|  | User must be able to delete survey questions. | Delete survey questions by clicking on the delete icon positioned at the bottom right hand corner of the survey question. | System instantly removes the survey question from the current form. | Success |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Single Line Text | User must be able to update the description of text type question | Add text type question by clicking on the survey question and go to “Field Properties”.  Default vaule is “text”.  For instance, user changes the question title to “What is your opinion?” | System instantly displays the changes while user updates the field. | Success |
| Number | User must be able to update the description of number type question | Update field label by clicking on the survey question to go to “Field Properties”  Default value is “Number”.  For instance, user changes the word to “Please enter your number:” | System instantly displays the changes while user updates the field to “Please enter your number”. | Success |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Paragraph Text | User must be able to update the description of paragraph text type of question | Update text type question by clicking on the survey question and to editing in the “Field Properties”  Default value is “Paragraph"  For instance, user changes the question title to “From the investigation, what would you like to provide to us? | System instantly displays the changes while user updates the field. | Success |
| Checkbox | User must be able to update the description of checkbox type question and options | Update field label by clicking on the survey question to go to “Field Properties”  Default value is “Checkboxes”.  For instance, user changes the word to “What days of the week are you available? (Check all that apply)” and options to “Monday”, “Tuesday”, “Wednesday”, “Thursday”, “Friday” and “All of the above” | System instantly displays the changes while user updates the field to “What days of the week are you available? (check all that apply)” | Success |
| Multiple Choice | User must be able to update the descriptions and options of multiple choice type of question | Update description and option by clicking on the survey question to go to “Field Properties”.  Defualt vaule is "Multiple Choice”.  For instance, user changes the desciption to “What is your favourite colour?”  Option can be edited also such as “Red”, “Black”, “White”. | System instantly displays the changes while user updates the field. | Success |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Drop Down | User must be able to update the descriptions and options of dropdown question and options. | Update field label by clicking on the survey question to go to “Field Properties”  Default value is “Drop down”.  For instance, user changes the word to “What is your favourite colour?” and options to “Red”, “Blue”, “Green”, “Yellow‘ and “None of the above” | System instantly displays the changes while user updates the field to “What is your favourite colour?” | Success |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Name | User must be able to update the format of name | Update name by clicking on the survey question to go to “Field Properties”  Default value is “Name”.  For instance, user can choose the format of the name want to display | System instantly displays the changes while user updates the field.  Eg: Simple Name  Eg: Extended Name | Success |
| Date | User must be able to update the descriptions and format of date question. | Update field label by clicking on the survey question to go to “Field Properties”  Default value is “Date”.  For instance, user changes the word to “Please enter your birthday:” | System instantly displays the changes while user updates the field to “Please enter your birthday:” | Success |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | User must be able to update the descriptions and format of date question. | Update field label by clicking on the survey question to go to “Field Properties”  Default format is “MM/DD/YYYY”.  For instance, user changes the format to “DD/MM/YYYY” | System instantly displays the changes while user updates the format to “DD/MM/YYYY” | Success |
| Phone | User must be able to update the descriptions and format of phone question. | Update field label by clicking on the survey question to go to “Field Properties”  Default value is “Phone”.  For instance, user changes the word to “Please enter your phone number:” | System instantly displays the changes while user updates the field to “Please enter your phone number:” | Success |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | User must be able to update the descriptions and format of phone question. | Update field label by clicking on the survey question to go to “Field Properties”  Default value is “(###)-###-####”.  For instance, user changes the format to “International” | System instantly displays the changes while user updates the field to “International” format. | Success |
| Website | User must be able to update the descriptions of website question. | Update field label by clicking on the survey question to go to “Field Properties”  Default value is “Web Site”.  For instance, user changes the word to “Please your company website:” | System instantly displays the changes while user updates the field to “Please your company website:” | Success |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Price | User must be able to update the format of currency type question | Update field label by clicking on the survey question to go to “Field Properties”  Default value is “$Dollar”  For instance, user changes the currency format to pound. | System instantly displays the changes while user updates the field.  Eg: $ Dollar  Eg: $ Pound | Success |
| Email | User must be able to update the descriptions of email question. | Update field label by clicking on the survey question to go to “Field Properties”  Default value is “Email”.  For instance, user changes the word to “Please enter your e-mail address:” | System instantly displays the changes while user updates the field to “Please enter your e-mail address:” | Success |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| File Uploaded | User must be able to update the descriptions of file upload question. | Update field label by clicking on the survey question to go to “Field Properties”  Default value is “Upload a File”.  For instance, user changes the word to “Upload file here:” | System instantly displays the changes while user updates the field to “Upload file here:” | Success |
| Slider | User must be able to update the descriptions of slider question and options | Update field label by clicking on the survey question to go to “Field Properties”.  Default value is “Slider”.  For instance, user changes the word to “How will you rate this website? (1 to 10, 10 is the most)” | System instantly displays the changes while user updates the field to “How will you rate this website? (1 to 10, 10 is the most)” | Success |
|  | User must be able to update the options of the slider question | Update field label by clicking on the survey question to go to “Field Properties”  Default value of slider range is “10”.  Default value of minimum text is “1.”  Default value of maximum text is “10.” | System instantly displays the changes while user updates the minimum text vaule to “2” and maximum text vaule to “8”. | Success |
| Scaler | User must be able to update the descriptions of scalar question and options. | Update field label by clicking on the survey question to go to “Field Properties”  Default value is “Scaler”.  For instance, user changes the word to “Please rate your attractiveness on a scale of 1 to 5.” | System instantly displays the changes while user updates the field to “Please rate your attractiveness on a scale of 1 to 5.” | Success |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | User must be able to update the descriptions of scalar question and options. | Update field label by clicking on the survey question to go to “Field Properties”  Default value of scale amount is “5”.  Default value of worse text is “Worst.”  Default value of best text is “Best.”  For instance, user changes the scale amount to “10”, worst text to “Ugly” and best text to “Pretty” | System instantly displays the changes while user updates the scale amount to “10”, worst text to “Ugly” and best text to “Pretty”. | Success |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Satisfactory | User must be able to update the descriptions of satisfactory type of question and its options. | Update field label by clicking on the survey question to go to “Field Properties”  Default value is “Satisfactory”.  For instance, user changes the word to “How do you rate our service provide to you” | System instantly displays the changes while user updates the field | Success |
|  | User must be able to update the options of the question for text input | Update field label by clicking on the survey question to go to “Field Properties”  Default value of options are “Very unsatisfactory, Unsatisfactory, Neutral, Satisfactory, Very Satisfactory”  For instance, user changes the options to “Very poor, Poor, Neutral, Good, Excellent” | System instantly displays the changes while user updates the field | Success |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | User must be able to update the options of the question for input image url | Update field label by clicking on the survey question to go to “Field Properties”  Default value of options are empty, user need to input the url link of the image in order to output the image  For instance, user input the image link into the field | System instantly displays the image while user updates the field | Success |
| Save Form | User must be able to save form after create the survey questions. | Save survey questions by clicking on the “Save Form” tab | System will save the survey questions into the database and publish the survey | Success |

viewsurveylist.html

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Functional Requirements | Execution Details | System response | Remarks |
| View Created Survey | User must be able to see a list of surveys created in the account | Click on “View Created Survey” link | System should display a list of survey created by the user. Created survey should have a link to view survey report and a download link as well as a survey link to go to the survey | Success |
|  | User must be able to see “No Surveys have been created” message if there is no existing surveys | Click on “View Created Survey” link | System should display a message as shown below: | Success |
| View Survey Report | User must be able to view graphs and statistics of the survey responses | Click on “View Survey Report” link | System should display the statistic results on the survey responses in graphs and charts.   * Pie Chart for radio button and scale radio questions  * Bar Graph for checkbox questions  * List for date, numerical inputs, text and textarea inputs questions | Success |
|  | User must be able to view the respondent data logging (time taken for the respondents to complete the survey) and the number of respondents who responded to the survey | Click on “View Survey Report” link | System should display the average time that respondents taken to complete the survey and the number of respondents who took and responded to the survey | Success |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Download CSV Report - Choose to save | User can download a version of a Comma-Seperated-Values (CSV) Report | User clicks on “Download Report” link | System will prompt the user to choose where they want to save the CSV report. | Success |
| Download CSV Report - Save the Report | After the user chooses where to save the CSV report, the user can then click the save button to store the CSV file | User clicks on the “Save” button | System will save the CSV Report in the desired place. | Success |

register.html

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Functional Requirements | Execution Details | System response | Remarks |
| Register | User must be able to register an account. | User keys in the valid response for every fields.  User clicks on “Register” link. | System will prompt user to enter user’s personal and account information.  Fields are all correctly entered. Registration done! | Success |
|  | User will be able to see “Enter valid email address” if email address format is incorrect. | User keys in an invalid email address.  User clicks on “Register” link. | A pop-out will appear to prompt the user that he/she has entered an invalid email. | Success |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | User will be able to see “Enter valid username” if username format is incorrect. | User keys in an invalid username.  User clicks on “Register” link. | A pop-out will appear to prompt the user that he/she has entered an invalid username. | Success |
|  | User will be able to see “Enter another username” if username is already taken. | User keys in a duplicated username.  User clicks on “Register” link. | A pop-out will appear to prompt the user that he/she has entered a username that has already been taken by another user. | Success |
|  | User will be able to see “Enter valid password” if password is too short. | User keys in an invalid password.  User clicks on “Register” link. | A pop-out will appear to prompt the user that he/she has entered an invalid password. | Success |
|  | User will be able to see “Password don’t match” if password and confirm password field does not match. | User keys in different password on two different fields. | A pop-out will appear to prompt the user that he/she has entered an unmatched password. | Success |

EditUserDetails.htm

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Functional Requirements | Execution Details | System response | Remarks |
| Edit Details | User must have an account and is able to update the account | User edit the fields and he/she has to re-enter password in order to update the details. | System will prompt that details are updated if all the fields are entered correctly. | Success |
|  | User tries to update with incorrect password. | User key in an invalid password when “update” button is clicked. | System will prompt the user that he/she has entered an incorrect password. | Success |
|  | User tries to update the account details with blank field. | User leave 1 or more fields empty when “update” button is clicked. | System will prompt the user that he/she has to fill in all the fields before proceeding. | Success |
|  | Email address format is not entered correctly. | User key in an invalid email address format when “update” button is clicked. | System will prompt the user that he/she has entered an invalid email. | Success |
|  | Email address entered is already in use. | User enters an email address that has already been used. | System will prompt the user that the email address has already been used. | Success |

UpdatePassword.htm

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Functional Requirements | Execution Details | System response | Remarks |
| Update password | User must have an account and is able to update the account password | User key in the old password, new password and confirm the password before proceeding to update. | System will prompt that password are updated successfully if all the fields are entered correctly. | Success |
|  | User tries to update with incorrect password. | User key in an invalid password that does not match with the username when “update” button is clicked. | System will prompt the user that he/she has entered an incorrect password. | Success |
|  | User tries to update with mismatch password. | User key in mismatch password for the new and confirm password field when “update” button is clicked. | System will prompt the user that the passwords entered are mismatch. | Success |
|  | User tries to update with password with less than 6 characters long. | User key in password with less than 6 characters when “update” button is clicked. | System will prompt the user that the password should be at least 6 characters long. | Success |

Login.htm

|  |  |  |  |
| --- | --- | --- | --- |
| **Functional Requirement** | **Execution Details** | **System Response** | **Remarks** |
| Non-registered username during system login | Fill in the username with the non-registered username. Fill in the password field with the correct password. Click on submit. | Not successful login and system will send alert saying username or password not found. | Success |
| Wrong password during system login | Fill in the username with the registered username. Fill in the password field with the wrong password. Click on submit. | Not successful login and system will send alert saying username or password not found. | Successfully implement the function |
| System login function | Fill in the username with the registered username. Fill in the password field with the correct password. Click on submit. | Registered user is logged in. User is redirect to registered user page. | Successfully implement the function |
| System facebook login function | Click on the "Login with Facebook" button. Log in using the registered Facebook account. | User is logged in to the page.  The facebook button will change to welcoming message and then user will be redirected to the home page of registered user. | Successfully implement the function |
| Non-registered facebook user login function | Click on the "Login with Facebook" button. Log in using the non-registered Facebook account. | Facebook user is registered. User data is captured and saved in the database. Then it will continue with the same response as the registered Facebook user login function. | Successfully implement the function |

1. Discussion

Due to our inexperience in Microsoft CLR languages, we initially struggled to setup the development environment.

A major and early problem was the setting up of the persistence layer. Due to the requirement for a quick and hassle free environment where we could just focus on business logic, we decided to use a ORM instead of handling SQL queries.

Therefore we chose nHibernate. However, there were few examples on how to get nHibernate to work with postgresql. Ultimately though, we managed to setup nHibernate and so made all persistence calls/interactions much easier.

1. Deriving analytical design and models

We derived our analytical designs and models from the functional requirements as listed in section 2.2 and also the use cases from section 2.6 and 3. By looking at the functional requirements, we are able to determine what at the things we want our program to be able to do, and thus, develop use cases. The use cases tell us what the users can do to perform the various functionalities as depicted in section 2.2.

For example, one of the functionalities of the program is to enable users to register an account with our system, hence, becoming a member. Thus, this feature is added into the use case diagram as depicted in section 2.6.1. Also, the corresponding use case description is added in section 2.6.2. Finally, from there, we determine that we require a member class store all the details a user has keyed in. Hence, a member class is added to the analytical design.

1. Going about performing UI design

Our UI design has gone through many stages of changes to become the final product that it is now. Firstly, we look at commercial websites and learn from those websites on how we can design our UI. We also looked for plug-ins we can put into our system to make user interaction with our system much easier and more dynamic. In the initial stages of designing the UI, we come up with a lo-fi prototype which is hand-drawn on a hard copy to visualize how we want our system to look like. Subsequently, we come up with a hi-fi prototype which has a closer resemblance to our final UI design. Finally, we added in other features to beautify our website, (eg. a slideshow of the features of our website) as finish touches to our UI design.

1. Difficulties encountered and solutions applied

We have faced technical limitations as Amazon places a high cost on computational cycles consumed and database transactions. Thus, we have moved most of application logic to client.

Another issue is the standard method of serving webpages by server generating HTML pages and sending them to the client not flexible enough to allow a mobile application. Thus, we used a Stateless RESTful Service Oriented Architecture

Another difficulties is that most of us are new to Javascript and C#. However we went on research from the web and with the help of programmer in our team, we managed to overcome this problem and has completed this project successfully.

1. Work breakdown structure