Analysis and Design of a Survey Management System

Tamara Barker
11861636

INFO 620-09F-OL
Information Systems Analysis and Design

Term Project
Project Category: Analysis & Design
# Table of Contents

**INTRODUCTION** .......................................................................................................................... 1  
**SYSTEM ANALYSIS** .................................................................................................................... 2  
  - Problem Statement ...................................................................................................................... 2  
  - Requirements ............................................................................................................................... 2  
    - Functional Requirements ......................................................................................................... 2  
    - Data Requirements .................................................................................................................. 3  
    - Business Rules and Logic ....................................................................................................... 3  
  - Use Case Model .......................................................................................................................... 4  
    - Actors and their Goals ............................................................................................................. 4  
    - Use Case Diagram .................................................................................................................. 4  
    - Use Case Description - UC: Take Survey ............................................................................... 5  
    - High-Level Use Case Descriptions ....................................................................................... 7  
    - Activity Diagram - Use Case: Take Survey ......................................................................... 10  
    - Use Case Model Discussion ................................................................................................ 11  
  - Class Model ............................................................................................................................... 11  
    - Analysis Procedure .............................................................................................................. 11  
    - Analysis Class Diagram ...................................................................................................... 12  
    - Selected Class Definitions .................................................................................................. 12  
    - Selected Association Definitions ......................................................................................... 13  
    - Class Model Discussion ........................................................................................................ 13  
**SYSTEM DESIGN** ....................................................................................................................... 15  
  - System Sequence Diagram – UC: Take Survey .................................................................... 15  
  - Expanding Sequence Diagram – UC: Take Survey ............................................................... 16  
  - State Diagram – Attribute: stu_resp_flag() ......................................................................... 16  
  - Design Class Diagram ............................................................................................................ 17  
  - Validation and Discussion ....................................................................................................... 17  
**EVALUATION OF ANALYSIS AND DESIGN** ........................................................................... 20  
**SUMMARY AND LESSONS LEARNED** .................................................................................... 20  
**REFERENCES** ............................................................................................................................ 21  
**APPENDIX A - USE CASE DIAGRAM** ................................................................................... 23  
**APPENDIX B – UNSOLVED PROBLEMS** .................................................................................. 24
INTRODUCTION

The purpose of this project is to put into practice the concepts and diagrams learned during this course. The concepts and diagrams involved are that of object-oriented analysis and design and the use of the Unified Modeling Language.

Due to the difficulties in working on a class project with students on a three-hour time zone difference and the fact that I work full time so my schedule is not terribly flexible, I chose to do my project as an individual, rather than team, project. This presented its own challenges, in that I did not have other students with whom to share the work load and I also did not have teammates with whom to brainstorm ideas regarding the analysis and design of the system.

The system I chose to analyze and design is a college Survey Management System. The system records graduating student responses to survey questions presented to them near the end of their educational career.

I began this project by focusing on System Design, starting with documenting the problem statement and system requirements. From this information I developed the Use Case Model, including a use case diagram and activity diagram. This was a relatively easy process to begin with. However, both diagrams were modified later in the project as I delved deeper into the actual design of the system and the use case diagram was also modified after receiving feedback from the instructor regarding the initial design. After the Use Case Model, I developed the Class Model, which included the analysis class diagram. The original design of this diagram remained fairly consistent throughout project development, with some minor modification after instructor feedback was received.

System Design was the more complex part of the project. The sequence diagrams were developed through several iterations. It was through the development of the sequence diagrams that I recognized the modifications that I needed to make to the Use Case Model. The design class diagram presented the greatest challenge to me, with the difficulty lying in identification and placement of class operations.

Although I did not have time to perform the Physical Design and Applications Design work, I am confident that the project I have presented in the following pages has been developed to the best of my ability in the time allotted and reflects the concepts taught during this course. I am also confident that I will be able to apply these concepts and methodologies to future business projects with guidance from my more experienced colleagues.
SYSTEM ANALYSIS

Problem Statement

The College of Information Science and Technology wants to create a web-based survey system (SMS) for graduating students. The overall goals of the system are to collect graduating students’ responses to survey questions and produce various survey charts and reports.

It is important for any organization to learn what it is doing well and where improvements are needed. Students often feel their opinion is unheard. A graduating-student survey management system will provide the College with a method to acquire important feedback from students so they may continue those efforts that work well and improve those areas that do not work well.

The SMS will include only questions about student life at the College, courses, professors and student services. Responses will be allowed on a scale of one to five, as well as provide for final written comments. The SMS will be targeted only to students graduating from the College of Information Science and Technology.

The SMS will not target non-graduating students, students of other colleges, faculty or staff. The SMS will not include questions about non-IST courses, professors or student services.

Requirements

Functional Requirements

The system will require a login with either a valid student school ID and system-generated password that will be sent to graduating students via e-mail or a valid administrator login ID.

In addition, the SMS must perform the following functions:

1. Manage administrator IDs
2. Manage respondent data
   a. Import list of graduating student data
   b. Generate passwords for imported student IDs
   c. E-mail generated passwords
   d. Remind students of the due date
3. Develop questions
4. Create survey from question pool
5. Present survey to respondents
6. Collect responses
7. Generate reports and charts
   a. For each category of concern
   b. For specific time frame (term, year, overall)
   c. Response statistics (number surveys sent vs. number surveys taken)
8. Back up data
**Data Requirements**

Respondents: ID, name, e-mail address, password, graduation term, major, response flag, invitation sent flag and date, reminder sent flag and date, last question answered, date survey completed

Employees: ID, name, password

Surveys: ID, term, creation date, open date, close date, reminder date, intro text, closing text

Questions: Category, text, response type

Responses: Response value or text, depending on type of question

Reports: Creation date, ID of employee who created report, last run date, ID of employee who last ran report

**Business Rules and Logic**

1. A new survey shall be generated each quarter. Questions are not expected to deviate often unless new programs or services need to be added.
2. Reports may be generated at any time.
3. Survey invitations will be sent 30 days before end of term, then weekly as new graduating students are uploaded to system.
4. A reminder message will be sent to non-responders 21 days after term end.

**Non-Functional Requirements**

1. Usability
   a. SMS will have online help documentation for administrators and survey takers.
   b. SMS shall require a response to each question, including comments.
2. Reliability
   a. SMS shall note respondent’s last valid response and return respondent to next question if respondent logs in again after exiting prematurely.
   b. SMS will be backed up nightly.
   c. System backups will be kept for 90 days.
3. Performance
   a. SMS will be available to survey takers any time except during scheduled backups. Backups will be scheduled in accordance with other IST recovery operations.
4. Supportability
   a. SMS shall allow for future modifications for use by other colleges and to allow new categories and reports.
5. Implementation
   a. SMS shall utilize English-language surveys.
   b. SMS shall be a web-based application developed in ASP.net.

6. Legal
   a. SMS shall be licensed for use only by Drexel University.

**Use Case Model**

*Actors and their Goals*

1. Respondent:
   a. Log into system
   b. Answer survey questions

2. Staff:
   a. Log into system
   b. Manage respondent data
   c. Manage survey questions
   d. Manage surveys
   e. Manage help documentation
   f. Send e-mail notices
   g. Generate reports

3. Administrator:
   a. Same goals as Staff, plus
   b. Manage system access
   c. Back up system data

**Use Case Diagram**

See Appendix A for full-page version of Use Case Diagram.
**Use Case Description - UC: Take Survey**

<table>
<thead>
<tr>
<th><strong>USE CASE #</strong></th>
<th>2</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>USE CASE Name</strong></td>
<td>Take Survey</td>
</tr>
<tr>
<td><strong>ACTOR</strong></td>
<td>Respondent</td>
</tr>
<tr>
<td><strong>Goal</strong></td>
<td>To fully respond to all questions of an open survey.</td>
</tr>
<tr>
<td><strong>Overview and Scope</strong></td>
<td>The system displays introductory text, then presents each question to the respondent and awaits an answer before presenting the next question. As the system receives each answer, it stores the answer and current respondent/survey status in the system database. After the final question has been answered, the system presents closing text and the respondent exits the survey.</td>
</tr>
<tr>
<td><strong>Level</strong></td>
<td>Primary</td>
</tr>
</tbody>
</table>
| **Preconditions** | 1. Respondent has online access.  
2. Respondent received a survey invitation containing password.  
3. Respondent has logged into the system.  
4. Survey has not yet expired. |
| **Post-Conditions** | 1. Each response was recorded in system database.  
2. Student survey status was recorded in system database.  
3. Closing text was presented to the respondent after last question was answered. |
| **Trigger** | A Respondent logs into system in response to a survey invitation. |
| **Included Use Cases** | None |
| **Extending Use Cases** | None |

**MAIN SUCCESSFUL SCENARIO**

<table>
<thead>
<tr>
<th><strong>Actor Action</strong></th>
<th><strong>System Action</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Respondent clicks on &lt;start survey&gt; button</td>
<td>2. System checks respondent’s last question number</td>
</tr>
<tr>
<td>3. If last question number is populated, system displays next survey question. Else system displays first survey question.</td>
<td></td>
</tr>
<tr>
<td>4. Respondent answers question and clicks on &lt;Submit&gt; button</td>
<td>5. System verifies that a response was entered</td>
</tr>
<tr>
<td>6. System records response in database</td>
<td></td>
</tr>
<tr>
<td>7. System records survey status in database</td>
<td></td>
</tr>
</tbody>
</table>
8. Additional survey questions?  
Yes: System displays next question. Repeat steps 4 through 7 until final question is answered.  
No: System displays closing text

<table>
<thead>
<tr>
<th>Step</th>
<th>Branching Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>*</td>
<td>Survey may be terminated prematurely at any time by respondent or due to system/network error. Upon restarting survey, step 2 ensures that system does not present respondent with previously answered questions.</td>
</tr>
</tbody>
</table>

**OTHER SUCCESSFUL SCENARIOS**

9. Respondent leaves website and/or closes browser

10. No action is required by system

**UNSUCCESSFUL SCENARIOS**

<table>
<thead>
<tr>
<th>Conditions</th>
<th>Actions</th>
</tr>
</thead>
<tbody>
<tr>
<td>*</td>
<td>Survey may be terminated prematurely at any time by respondent or due to system/network error. No further action will be taken by system.</td>
</tr>
<tr>
<td>4a. Respondent clicked on <code>&lt;submit&gt;</code> button without entering a response to question</td>
<td>Re-display question with message to enter response before clicking <code>&lt;submit&gt;</code>.</td>
</tr>
</tbody>
</table>

**Priority in scheduling**  
First

**Frequency**  
This use case may occur an unlimited number of times each day. May occur multiple times per respondent.

**Other non-functional requirements**  
None

**Superordinates**  
None

**Developer**  
Tamara Barker

**Creation date and last modified date**  
11-29-09 Created  
12-03-09 Modified

**Other Comments**  
None
**High-Level Use Case Descriptions**

<table>
<thead>
<tr>
<th>Use Case Name</th>
<th>Actor(s)</th>
<th>Purpose</th>
<th>Overview</th>
</tr>
</thead>
<tbody>
<tr>
<td>Log In</td>
<td>Respondent, Staff, Administrator</td>
<td>To log in to survey management system</td>
<td>Shows the process of logging into the system.</td>
</tr>
<tr>
<td>Take Survey</td>
<td>Respondent</td>
<td>To take an open survey</td>
<td>Shows the process of a respondent taking a survey.</td>
</tr>
<tr>
<td>Manage Respondent Data</td>
<td>Staff, Administrator</td>
<td>To manage respondent data</td>
<td>Shows the process of managing respondent data.</td>
</tr>
<tr>
<td>Import Student Data</td>
<td>Staff, Administrator</td>
<td>To import student data into system</td>
<td>Extends UC: Manage Respondent Data. Shows the process of importing graduating student data into the system.</td>
</tr>
<tr>
<td>Edit Student Data</td>
<td>Staff, Administrator</td>
<td>To modify or delete student data into system</td>
<td>Extends UC: Manage Respondent Data. Shows the process of modifying or deleting student data already in the system.</td>
</tr>
<tr>
<td>Generate Passwords</td>
<td>Staff, Administrator</td>
<td>To generate student passwords for survey response</td>
<td>Extends UC: Manage Respondent Data. Shows the process of generating passwords for graduating students whose information has been imported into the system.</td>
</tr>
<tr>
<td>Manage Help Documentation</td>
<td>Staff, Administrator</td>
<td>To edit online help documentation</td>
<td>Shows the process of editing online help documentation for all system users.</td>
</tr>
<tr>
<td>Manage Survey Questions</td>
<td>Staff, Administrator</td>
<td>To manage survey questions</td>
<td>Shows the process of creating, editing and deleting survey questions.</td>
</tr>
<tr>
<td>Use Case Name</td>
<td>Description</td>
<td></td>
<td></td>
</tr>
<tr>
<td>--------------------</td>
<td>-----------------------------------------------------------------------------</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Create Question</td>
<td>To add a new question to the question database. Extends UC: Manage Survey Questions. Shows the process of adding a new question to the database for inclusion in a survey.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Modify Question</td>
<td>To modify existing questions in the question database. Extends UC: Manage Survey Questions. Shows the process of editing or deleting existing questions in the question database.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Manage Surveys</td>
<td>To manage surveys. Shows the process of creating, modifying and closing surveys.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Create Survey</td>
<td>To create a new survey. Extends UC: Manage Surveys. Shows the process of creating a new survey.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Copy Survey</td>
<td>To copy an existing survey as a template for a new survey. Extends UC: Create Survey. Shows the process of creating a new survey by copying an existing survey’s question and text attributes.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Modify Survey</td>
<td>To modify an existing survey. Extends UC: Manage Surveys. Shows the process of editing or deleting an existing survey.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Send E-Mail</td>
<td>To send e-mail notifications regarding surveys. Shows the process of sending an e-mailing.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Send Survey Invitation</td>
<td>To send survey invitations to graduating students. Extends UC: Send E-Mail. Shows the process of e-mailing a survey invitation to graduating students.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Use Case Name</td>
<td>Description</td>
<td></td>
<td></td>
</tr>
<tr>
<td>---------------------------</td>
<td>-----------------------------------------------------------------------------</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
| **Send Survey Reminder**  | Actor(s): Staff, Administrator  
                                  Purpose: To send survey reminder to non-responding invitees  
                                  Overview: Extends UC: Send E-Mail. Shows the process of e-mailing a survey reminder to non-respondents. |
| **Generate Report**        | Actor(s): Staff, Administrator  
                                  Purpose: To generate a report of survey data  
                                  Overview: Shows the process of generating reports of survey data. |
| **Create Report**         | Actor(s): Staff, Administrator  
                                  Purpose: To create a new report  
                                  Overview: Extends UC: Generate Report. Shows the process of creating a new report. |
| **Modify Report**         | Actor(s): Staff, Administrator  
                                  Purpose: To create a new report  
                                  Overview: Extends UC: Generate Report. Shows the process of modifying or deleting an existing report. |
| **Manage System Access**  | Actor(s): Administrator  
                                  Purpose: To manage staff access to system  
                                  Overview: Shows the process of managing staff user id’s, passwords, and system security. |
| **Back Up Data**          | Actor(s): Administrator  
                                  Purpose: To back up system data  
                                  Overview: Shows the process of backing up system data. |
Activity Diagram - Use Case: Take Survey

Main Success Scenario

<table>
<thead>
<tr>
<th>Respondent</th>
<th>System</th>
</tr>
</thead>
<tbody>
<tr>
<td>Start Survey</td>
<td>Record Survey Status</td>
</tr>
<tr>
<td></td>
<td>Determine Starting Question</td>
</tr>
<tr>
<td></td>
<td>Present Question</td>
</tr>
<tr>
<td>Respond to Question</td>
<td>Validate Response</td>
</tr>
<tr>
<td></td>
<td>Record Response</td>
</tr>
<tr>
<td></td>
<td>Record Question Number</td>
</tr>
<tr>
<td>Additional Questions?</td>
<td>Record Survey Status</td>
</tr>
<tr>
<td>No</td>
<td>Record Completion Date</td>
</tr>
<tr>
<td></td>
<td>Present Final Text</td>
</tr>
<tr>
<td>Exit Survey</td>
<td></td>
</tr>
</tbody>
</table>

Yes
Use Case Model Discussion

Modeling decisions include:

1. Use Case Diagram
   a. Administrator can perform all functions of Staff

2. UC: Take Survey Activity Diagram
   a. Loop assumes that questions are presented to respondents one at a time

Alternative solutions considered include:

1. UC: Take Survey Activity Diagram
   a. Considered presenting all questions at one time. Experiences with surveys presented in this manner has shown that some respondents tend to glance ahead and not focus on the immediate question, so felt it best to loop through the questions.

Class Model

Analysis Procedure

<table>
<thead>
<tr>
<th>Nouns</th>
<th>Class Elimination Rules</th>
<th>Class Categories</th>
<th>Class</th>
</tr>
</thead>
<tbody>
<tr>
<td>College</td>
<td>CER2 - Irrelevant</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Survey</td>
<td>No</td>
<td>CC7 – Concept w/Properties</td>
<td>X</td>
</tr>
<tr>
<td>Student</td>
<td>No</td>
<td>CC1 – People (Rename to Respondent)</td>
<td>X</td>
</tr>
<tr>
<td>Response</td>
<td>No</td>
<td>CC7 – Concept w/Properties</td>
<td>X</td>
</tr>
<tr>
<td>Question</td>
<td>No</td>
<td>CC7 – Concept w/Properties</td>
<td>X</td>
</tr>
<tr>
<td>Report</td>
<td>No</td>
<td>CC7 - Concepts w/Properties</td>
<td>X</td>
</tr>
<tr>
<td>Organization</td>
<td>CER2 - Irrelevant</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Feedback</td>
<td>CER7 – Attribute</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Course</td>
<td>CER8 - Values</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Professor</td>
<td>CER8 - Values</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Student Services</td>
<td>CER8 - Values</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Comments</td>
<td>CER7 – Attribute</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Login</td>
<td>CER4 – Operation</td>
<td></td>
<td></td>
</tr>
<tr>
<td>ID</td>
<td>CER7 – Attribute</td>
<td></td>
<td></td>
</tr>
<tr>
<td>E-Mail</td>
<td>No</td>
<td>Template</td>
<td>X</td>
</tr>
<tr>
<td>Administrator</td>
<td>No</td>
<td>CC1 – People (Rename to Employee)</td>
<td>X</td>
</tr>
<tr>
<td>Due Date</td>
<td>CER7 – Attribute</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Security</td>
<td>No</td>
<td>CC7 – Concept w/Properties</td>
<td>X</td>
</tr>
<tr>
<td>Response_Type</td>
<td>No</td>
<td>CC7 – Concept w/Properties</td>
<td>X</td>
</tr>
</tbody>
</table>
Analysis and Design of a Survey Management System
Tamara Barker (11861636)

Analysis Class Diagram

Selected Class Definitions

Response_Type is a discovered class. This is a reference class which contains acceptable types of responses to survey questions. Initially, the acceptable types of responses are value and text.
Selected Association Definitions

1. Survey – Respondent Association
   a. Surveys can be created before respondents are identified.
   b. Respondents can be imported before their survey has been created.
   c. Respondents may respond to multiple surveys, if previously responded then changed graduation date resulting in new survey invitation.

2. Survey – Question Association
   a. Surveys can be created without adding questions. Questions may be added any time before survey is opened for response.
   b. Questions can be created but not yet assigned to any surveys.
   c. Questions can be reused by many surveys.

3. Employee – Report Association
   a. Want to know who created report and when, and last date it was run and by whom, so system administrator can purge old unused reports after checking with report creator.

Class Model Discussion

Modeling decisions include:

1. Employee Subclass Attributes:
   a. Sys_Admin and Staff are not different other than their security levels, which are noted through sec_lev in Class: Security. A Sys_Admin may perform same functions of Staff. Therefore, no additional attributes in subclasses.

2. Respondent Attributes
   a. Each respondent can have a different invitation date, since respondents can be added as they are identified as candidates for graduation even after survey has opened for response. Therefore, respondent, rather than survey, has invitation date attribute.
   b. Each respondent for a given survey is likely to have the same reminder sent date, but we want to know for each respondent when their survey was actually sent versus when it was supposed to be sent. Therefore, respondent has reminder sent date attribute.
   c. Attribute stu_resp_flag has possible value of “p” which indicates respondent has partially completed the survey.

3. Survey Attributes
   a. Survey reminder date is the date when the survey reminder should be sent by staff, not the date it was actually sent. Date it was actually sent is stored in Respondent (see 5b).

4. Email_Template Attributes
   a. Flags and dates for specific instances of e-mail notifications are stored in Respondent class, as individual instances of e-mail notifications are not stored.
Alternative solutions considered include:

The Security class was the only class for which I considered an alternative. My concern for this class was in how to store the various screens, data, and reporting permissions. The alternative solution was to include a Permission_Type class, which would allow for different permissions read or write mode. I decided that the nature of this system did not require such security and that all permissions can be represented by a single attribute.

Scenarios used during modeling:

1. A survey is created from an existing survey, finalized with questions and other required information, and an invitation is sent to graduating students.
2. A respondent starts the survey but must exit before completing it. The respondent logs in later to complete the survey.
3. A reminder is sent to respondents who have not completed the survey, including those who started but did not finish.
4. A staff person creates a report, runs it, and saves it to her hard drive.
SYSTEM DESIGN

System Sequence Diagram – UC: Take Survey
Main Success Scenario

Service: Take Survey

Start: R1. startSurvey()

Display question

Respondent

Determine first question to display

[for each unanswered question]

Display question

Survey Management System

Record question number

R2. submitResponse()

Validate response

Record response

Record survey status

Record survey status

Display closing text

Record completion date
Expanding Sequence Diagram – UC: Take Survey
Main Success Scenario

State Diagram – Attribute: stu_resp_flag()
Sequence Diagrams: The sequence diagrams were developed using the ten-step heuristics methodology as detailed below. I found this methodology extremely helpful in determining the objects to be included in the expanding SQD, which was certainly one of the more difficult diagrams I developed for this project.
SQD Ten-Step Heuristic for UC: Take Survey

1. Actor
   Event
   Respondent

2. Primary Boundary Object
   Survey Window

3. Use-Case Controller
   Survey Handler

4. Secondary Control Object
   None

5. Secondary Boundary Object
   None

6. Participating Domain Classes
   Respondent, Survey, Question, Response_Type, Response

7. Block Labels
   Survey Window, Survey Handler, Respondent, Survey, Question, Response_Type, Response

8. Major Operations:
   8.1. Instance Creation
   Survey Handler, Survey, Question
   8.2. Association
   Connect Respondent to Survey
   Connect Survey to Question
   Connect Question to Response_Type
   8.3. Attribute Forming
     8.3.1. Calculation
     None
     8.3.2. Change States
     recordResponse()
     updateStudentSurveyStatus()
   8.3.3. Display/Reporting Reqs
     getLastQuestionNumber()
     getQuestion()
     displayQuestion()
     getResponseType()
     displayClosingText()
   8.3.4. Interface with External Objects or Systems
     None

9. Rearrange the Sequence
   Not necessary

10. Name Each Message
    To be done during Design stage

Use Case Verbs

Display Question, Verify Response, Record Response, Record Status, Display Closing Text
State Diagram: Due to the relative simplicity of the Survey Management System, I was unable to identify a class or attribute that would normally qualify as complex enough to require modeling. However, I wished to develop a state diagram for practice, so I chose to model the activity of the Respondent class attribute named stu_resp_flag().

From the time a respondent is added to a survey, to the time the respondent completes the survey, the stu_resp_flag() goes through three state changes. When the respondent is added to a survey the flag is set to “N”, indicating that the respondent has not yet responded to the survey. Upon starting the survey, the flag is set to “P”, indicating that the respondent has started but not completed the survey. Finally, after the respondent answers the last question the flag is set to “Y”, indicating that the respondent has completed the survey.

While this is a simple diagram, it helped me to see that I had not included the initial state change event in both my system sequence diagram and activity diagram. Initially I had actually assumed it would be set in other use cases, but I realized through the simple process of thinking through the design of the state diagram that it actually needed to occur during the “Take Survey” use case. Thus, I feel this was a useful exercise as it helped refine previously developed diagrams.

Design Class Diagram: The design class diagram was the most challenging of all the diagrams that I completed for this system. I feel that it should not be as challenging as it was, since it was building upon the previously developed analysis class diagram, but perhaps my challenge was due to my limited experience with object-oriented design and development. The most difficult part was identification of operations. I spent a great amount of time determining placement of operations into their appropriate classes. The SQD for UC: Take Survey helped with the operations involved in that use case. I’m confident that if I had had time to develop SQDs for the other use cases I would have found it easier to identify operations and their appropriate placement.
EVALUATION OF ANALYSIS AND DESIGN

Performing analysis and design of a Survey Management System was a good project for an individual project. Neither the scale nor scope of the domain was so large as to be overwhelming for just one individual, although completing the project as an individual rather than as part of a team does have its challenges in terms of not being able to share the workload or having other people with whom to brainstorm ideas. Given the opportunity to repeat this project, I would welcome the chance to work with a team as it would provide me with other points of view that may differ and perhaps improve the project design. Given more time to complete the project, I would definitely develop the optional diagrams that I did not manage to develop, as I found that with each new diagram I was able to validate my previous designs or improve them based upon the additional knowledge of the system I gained through each diagram.

I have been familiar with the benefits of diagrams in understanding work flows and a variety of projects, from non-technical to programming projects. This is the first time that I have worked with so many diagrams for one project and I understand the benefits of each diagram both for ensuring mutual understanding of the system between the designers and the customers and for helping the designers work through the more complex elements of the system. The UML itself is especially useful as it incorporates the diagrams with use case driven analysis. This enables the developers to work from the high-level needs of the users down to the details of the system by focusing on the most important needs and developing the various components of the system through iterations. The modeling tool I used for this project was Visio. I used Visio rather than Rational Rose due to my previous experiences with Visio. Visio is easy to learn and easy to use in general. I did not have time to learn Rational Rose, but I suspect that it is a far more effective tool for UML modeling since it was designed for that purpose. Nevertheless, I would have no qualms about using Visio on another project if Rational Rose was not available.

SUMMARY AND LESSONS LEARNED

Overall, I am pleased with the work that I have done on this project. I spent a great deal of time working on the project and while I do not doubt that there are areas that may use some improvement, I feel that my analysis and diagrams represent the best effort I was able to put forth. The process of developing a realistic system reinforced the concepts and diagrams I was learning and enabled me to appreciate their value in the analysis and design process.

One experience during the development of this project particularly stands out for me: Upon writing my detailed use case descriptions, I questioned the validity of the extending use case named “Restart Survey” that I had originally included in the use case diagram. However, I felt I needed more information before I would be comfortable deleting the use case. Upon developing the expanding sequence diagram for use case “Take Survey”, I confirmed my belief that the extending use case was unnecessary, as it actually only involved one step which I was easily able to incorporate into the primary use case “Take Survey”. This experience, and similar experiences that followed with each new diagram, emphasized the importance of iterative analysis and design work during system development and reinforced how each new diagram enhances and refines the understanding of the overall system.
REFERENCES

http://www.agilemodeling.com/artifacts/sequenceDiagram.htm

APPENDICES
APPENDIX A - USE CASE DIAGRAM

Full-page Use Case Diagram included here as it does not fit well in body of the report.
APPENDIX B – UNSOLVED PROBLEMS

As mentioned previously in this report, concerns remain regarding the identification and placement of the class operations in the design class diagram. This is the part of the project with which I struggled the most and I am not sure that I correctly identified all required operations or their class placement.