Mobile Student Lookup

Project Plan

Brandon Knight, Mark Vitale, Chris Gropp, Ann Say

Table of Contents

1 Process Description and Rationale

For the Mobile Schedule Lookup System, an incremental development cycle has been chosen. Each subsystem is simple enough to have its requirements captured in a short period of time and small enough to require little more than a single iteration to complete. Additionally, the system's simplicity lends itself to very well-defined requirements. The subsystems themselves are incredibly distinct – for the most part, they independent of each other, and can be developed as such.

There are 4 distinct phases in this development cycle: Inception, Elaboration, Construction, and Evaluation. Inception involves gathering requirements from the customer and assessing risks on each component that is developed. Elaboration is the design and analysis of a given component. Construction is the coding and testing, creating the system's software deliverables and integrating them into existing components. Evaluation is getting and responding to customer and user feedback.

During the Inception phase the team will meet with the customer to obtain requirements. Requirements in hand, they will draft a rough sketch of the subsystems involved with the overall system, and then plan and analyze each subsystem for risks that may be involved with its development.

The Elaboration phase contains a more in-depth architecture and design for each component. This will fully consider risks and requirements gathered in the Inception phase, and should create a robust plan for the development of each subsystem.

In Construction, the actual systems will take form. Members of the team will construct individual subsystems, test them, integrate them into the system as a whole, and then test the overall system with the component integrated in.

Finally, during Evaluation the team will initially show a working system to the client, then user test it to ensure a quality end product. All feedback will be considered and needed changes will be integrated into the final system.

2 Project Schedule

The work breakdown tasks were derived from the various components of the system. Because most are independent, they can be worked on in any order. The sole exception to this is the overlay display method, which depends upon the schedule display method for its look and feel and overall development.

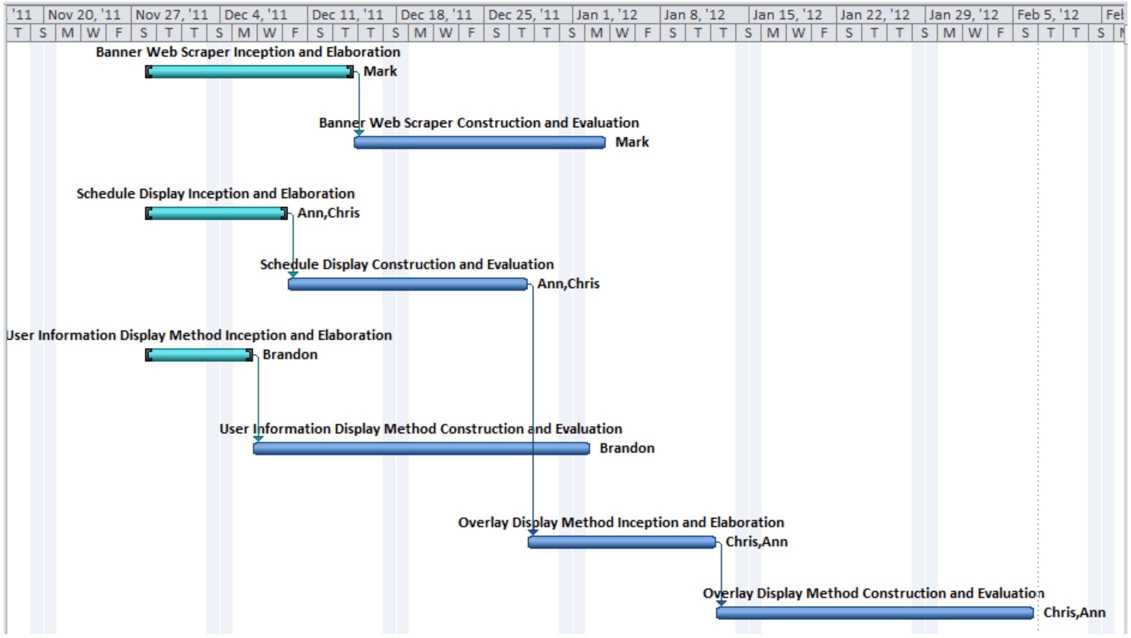
Each subsystem must have its inception and elaboration phase completed before it can be constructed and evaluated. The inception and elaboration phases have been combined, as the inception phase leads very nicely into elaboration. The construction and evaluation phases have been combined for similar reasons. Each subsystem is simple enough that it should take no more than one or two resources to complete.

2.1 Project Gantt Chart

Below is a timeline of the project that was created in Microsoft Project, and based off of the Work Breakdown Structure and critical path analysis.

Because there are essentially three paths in the project’s chart, risks and their management will have a minimal impact on the project as a whole outside of the critical path. Should any come to fruition, there is a large amount of flexibility in scheduling to allow for extra time and effort to be spent on a component.

The critical path itself has a large amount of flexibility in its start date. Because it must be finished before February 23rd, 2012, the critical path also has a fair amount of flexibility in its completion. Early completion is preferable, but should risks become reality, there is a fair amount of flex time to allow for compensation in the critical path.



Component	Early Start	Late Start	Slack
Banner Web Scraper I. & E.	11/28/2011	1/2/2012	35 days
Banner Web Scraper C. & E.	12/14/2011	1/16/2012	35 days
Schedule Display Method I. & E.	11/28/2011	11/28/2011	0 days
Schedule Display Method C. & E.	12/9/2011	12/9/2011	0 days
Overlay Display Method I. & E.	12/28/2011	12/28/2011	0 days
Overlay Display Method C. & E.	1/12/2012	1/12/2012	0 days
User Information Display Method I. & E.	11/28/2012	1/3/2012	36 days
User Information Display Method C. & E.	12/6/2011	1/11/2012	36 days

3 Risk Management

Risk will be managed throughout every step of the process. The most risky subsystems in the software will be identified in the following section, along with potential ways to mitigate these risks as well as manage them in the case that they do crop up.

3.1 Risk Categorization

3.1.1 Description of Risk Categories

Impact Values	Description
4	Catastrophic – If this risk were to occur, the project will either take major losses in terms of time, functionality, or cost or will halt development on the project altogether.
3	Critical – If this risk were to occur, the project may suffer losses of time, functionality, or cost, but to a lesser extreme than a catastrophic risk.
2	Marginal – If this risk were to occur, the project may suffer slight setback on the development timeline.
1	Negligible – If this risk were to occur, the project may suffer minimal loss of time (at most one week).

3.2 Risk Table

The table below categorizes the top ten risks associated with the Mobile Student Lookup Application according to the Risk Value of each. This is calculated from the Impact value it might have on the project times its Probability of Occurrence.

Risks	Probability of Occurrence	Impact	Risk Value
Poor team chemistry (RS1)	60%	3	1.8
Difficulty with Kerberos authentication (RS2)	35%	4	1.4
Issues parsing the raw Banner Web data (RS3)	45%	3	1.35
Problems acclimating to iOS development standards (HIG, etc.) (RS4)	50%	2	1.0
Learning curve of Cocoa Touch framework	40%	2	0.8
Learning curve of Objective-C	80%	1	0.8
Difficulty simulating actual usage environment	25%	2	0.5
Adapting to the new changes in iOS5	20%	2	0.4
Issues acquiring quality hardware	10%	4	0.4
Problems acclimating to the Mac OS X environment	25%	1	0.25

Table 1- Risk Analysis Table

3.3 Risk Mitigation, Monitoring, and Management Plan (RMMM)

Below are the detailed Risk Information Sheets for each of the Risks with associated Risk values greater than or equal to 1.0. Each risk with a value less than 1.0 would cost more to mitigate than the impact that the risk would have on the project. Therefore, only risks with high values were considered for a risk sheet.

3.3.1 Risk Information Sheet for RS1

Risk ID: RS1	**Date: 10/31/11**	**Prob: 60%**	**Impact: Critical**	**Risk Value: 1.8**

Description:

Poor team chemistry can lead to adverse effects on the project. This can include major setbacks in terms of the development timeline, application functionality, etc., as well as possibly ostracize or alienate certain members.

Refinement/context

Subcondition 1: One or more team members have different views on developing certain modules of the project.
Subcondition 2: One or more team members come from different backgrounds and this affects the ways in which they work.

Mitigation/Monitoring

1. Give anonymous team surveys once each week and have a 3rd party review them.
2. Hold weekly team meetings with this said 3rd party to discuss any major issues.
3. Use these meetings to resolve issues immediately as they appear.

Management/Contingency Plan/Trigger

Have team members interact outside of project context with planned group activities.

3.3.2 Risk Information Sheet for RS2

Risk ID: RS2	**Date: 10/31/11**	**Prob: 35%**	**Impact: Catastrophic**	**Risk Value: 1.4**

Description:

For this application to function correctly, the code must be able to properly interact with the existing databases of student data. In this specific case, the application needs to authenticate via Kerberos. If this cannot be done, data will not be able to be scraped, and the rest of the application will be rendered useless.

Refinement/context

Subcondition 1: Rose-Hulman keeps student data, including name, major, campus box, etc. in a database.
Subcondition 2: Rose-Hulman keeps student schedule data for each quarter in a database.
Subcondition 3: Access to these databases through the systems already provided requires authentication via Kerberos.

Mitigation/Monitoring

1. Create a test suite for testing the accessing of these databases.
2. Run this suite each time any major change to the code that deals with Kerberos occurs.
3. Keep in contact with IAIT of Rose-Hulman to ensure that any changes that they make to the authentication system do not adversely affect the application.

Management/Contingency Plan/Trigger

Leverage information from other open-source iOS projects that work with Kerberos authentication.

3.3.3 Risk Information Sheet for RS3

Risk ID: RS3	**Date: 10/31/11**	**Prob: 45%**	**Impact: Critical**	**Risk Value: 1.35**

Description:

The data provided by Banner Web must be parsed to provide pertinent information so the application can function correctly. Currently the raw data provided by Banner Web is not provided in a simple, clean manner, so parsing all of it to work well for the application could prove to be difficult.

Refinement/context

Subcondition 1: Rose-Hulman keeps student data, including name, major, campus box, etc. in a database.
Subcondition 2: Rose-Hulman keeps student schedule data for each quarter in a database.
Subcondition 3: Access to this data is provided via various systems such as Banner Web.
Subcondition 4: For this application, data will be scrapped directly from the pages provided from a Banner Web query.

Mitigation/Monitoring

1. Create a test suite for testing the parsing of test data.
2. Run this suite each time any major change to the code that deals with scraping data occurs.
3. Keep in contact with IAIT of Rose-Hulman to ensure that any changes that they make to the Banner Web system do not adversely affect the application.

Management/Contingency Plan/Trigger

If traditional abstract parsing fails, create a different regex for each type of page available on Banner Web.

3.3.4 Risk Information Sheet for RS4

Risk ID: RS4	**Date: 10/31/11**	**Prob: 50%**	**Impact: Marginal**	**Risk Value: 1.0**

Description:

The iOS platform is much different than a pen-based (most likely Windows-based) tablet. There are a host of design, memory, and philosophical issues that need to be considered when developing on iOS in order to make it a successful application

Refinement/context

Subcondition 1: Apple Human Interface Guidelines not understood by engineers.
Subcondition 2: Pen-less interaction not fully understood by engineers.
Subcondition 3: Code may not readily support multi-touch or finger-based input.

Mitigation/Monitoring

1. Constantly evaluate the current application against the HIG
2. Make engineers use an iPad in their daily workflow to become familiar with the platform
3. Use system on physical device (as opposed to the simulator) as often as possible when testing to ensure a finger is not too big to use all of the controls

Management/Contingency Plan/Trigger

Hire an outside consultant to evaluate the current coding based on the standards set by Apple and adjust it accordingly.

4 Tracking and Control Mechanisms

4.1 Quality Assurance

A suite of unit tests will be developed for each major module of the system. With every iteration of the project, these will be re-run and expanded as necessary to ensure that the system as a whole functions throughout the integration process. A usability test will be conducted upon the various user interfaces of the system and this data will be used to improve the overall quality of the user systems.

4.2 Change Management

Github will be used for the version control during the coding process. Github is a website for software development projects and has a built in version control system to manage code. Github will track bugs marked by users and also features requested by others. The version control system will handle conflicts if and when they arise.

When a user requests updates/features, their requests will be marked as a bug at a certain location in the code. The developers and clients would then decide whether to fulfill or deny the request.

4.3 Project Resources

4.3.1 People

Four project developers will be working on the Mobile Student Lookup application, each a Software Engineering and/or Computer Science major.

4.3.2 Hardware Requirements

The project's main hardware resources will be the customers' iOS devices and the hardware backbone of Rose-Hulman's Banner Web system. The Banner Web system is funded and maintained by Rose-Hulman, and is outside the control of X Incorporated.

So far as customer hardware goes, they will need an iOS device that supports iOS 5.

Development hardware will be supplied by the client.

4.3.3 Minimal Software Requirements

The project will require Apple's iOS development suite, xCode. It will also require the device that runs the Mobile Student Lookup system to have iOS 5.

4.3.3.1 Development

- xCode v4
- iOS 5 (free)

Who Done It: Team Member Names	Section/Part Completed	Task/Comments	# of hours effort
Mark Vitale	All	Wrote Sections	2.5 hours
Brandon Knight	All	Compiled into Project Plan	1 hour