Team Falcon Squad
Project 4 Report
Tour Buddy

Team Role	**Name**
Project Manager	Wan Zulsarhan Wan Shaari
Systems Engineer	Alberto Gomez-Estrada
Architect	Tyler Cobb
Testor/Integrator	DJ Todd

Problem Statement

When prospective students are on a Campus Tour they are typically accompanied by a tour guide who has a lot of knowledge about the campus. After this tour is over prospective students usually wander the campus by themselves. Sometimes these students may feel lost because they are not familiar with campus and all the buildings around. Also, there is so much information about different buildings that prospective students may not learn while wandering around on their own. To solve this problem, a mobile application with the purpose of enhancing the experience of prospective students touring Iowa Slate was developed. This project consists of an Android application that displayed a map as well as information about nearby buildings. This also includes pictures of nearby buildings. When users enter a building, the application display the information about the current building they are standing in. Inside this application, users are also able to save notes about buildings. These notes allow users to record information about certain buildings that they want to remember. Aside from the user end application, there is also information that is stored on a website about the number of buildings that have been visited. This information could be used by qualified University officials to review which buildings on campus have the most activity.

In this project expansion we plan to focus on polishing up the app and ready it for a possible market release following the end of this 4 week sprint. The original project create functional and well developed application. To accomplish this we will need to do a general overhaul and redesign of the front facing GUI. As well as put the app through rigorous testing on both the development side as well as with a possible user group to see how the app handles with people trying to learn more about ISU.

Feasibility

Every member of our team has experience working in Android Development which allows us to not waste any time learning the software. Along with previous experience, current mobile development tools allow for an efficient and quick implementation of the app. Using Android Studio as the IDE and the Google Maps API, we can develop and ship an application with almost full functionality. The main challenge lies in implementing the interface between the server and the application, but this can be managed with tools, such as MySQL Workbench.

Risk

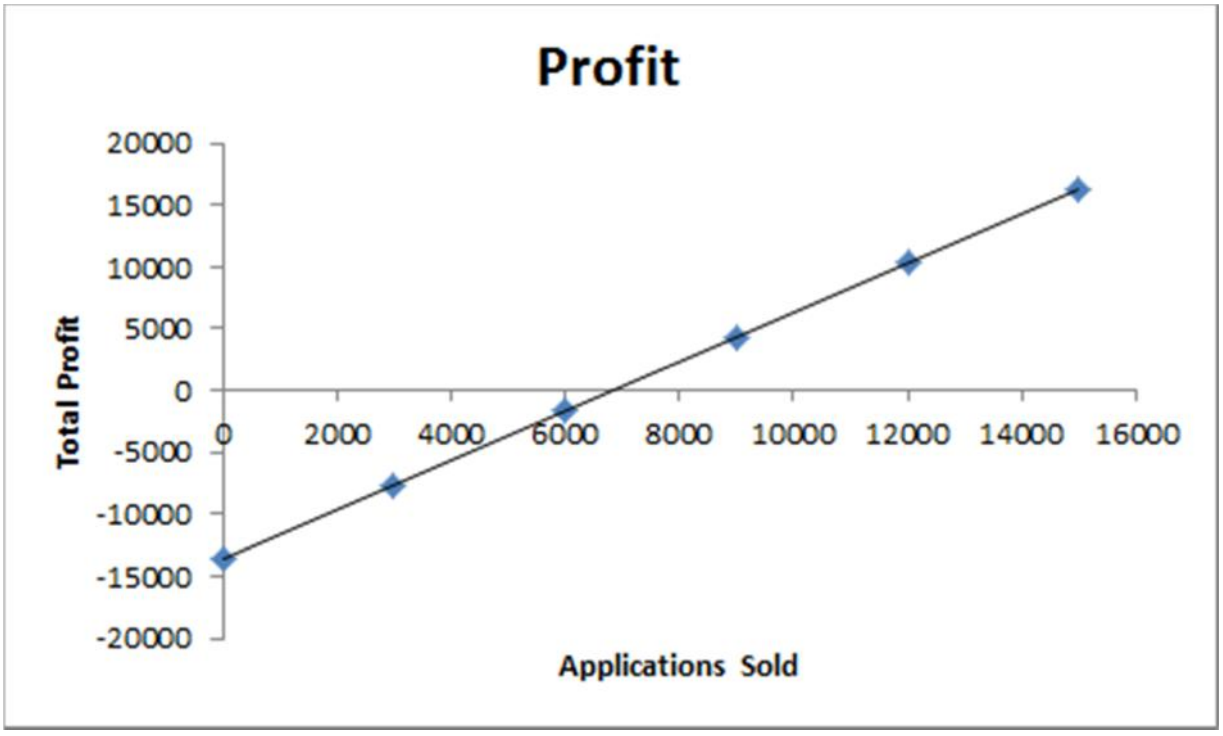
Risk	Probability	Criticality	Risk Factor	Mitigation
Server goes down	.10	50	5.0	Store the data locally and keep retrying server connection. Hold data until server connection is reached.
Cannot connect to internet	.20	20	4.0	If internet connection cannot be established then the app cannot be started because user will not be able to log in. Continue to try and log in until internet connection is established.
Competition with similar applications	.05	30	1.5	Investigate other apps to see how to make ours stand out. App is currently specific to Iowa State and competition has not been found… yet.
TourNote contains profanity	.20	5	1.0	Implement note checking to ensure that a note cannot be submitted until it has been checked for profanity.

GQM

Goal	Question	Measure
Learn project code base first week.	How long will it take team members to review code?	How many work days are spent learning code before all members understand the code base.
Complete UI redesign by end of Spring 2015	How long will the UI take to produce?	Weeks to produce UI, as well as weeks to approve UI adjustments via case studies.
Add 5 new buildings complete with photos and facts.	How many photos will be needed? Can they be found on the internet or do they need to be taken by the designers?	How many buildings are completed by end of current sprint.
Create a more engaging tour experience at Iowa State University.	When first coming to visit Iowa State do visitors feel that the app is helpful?	When given the app how much time do new visitors use the app to get around and/or learn about their surroundings.

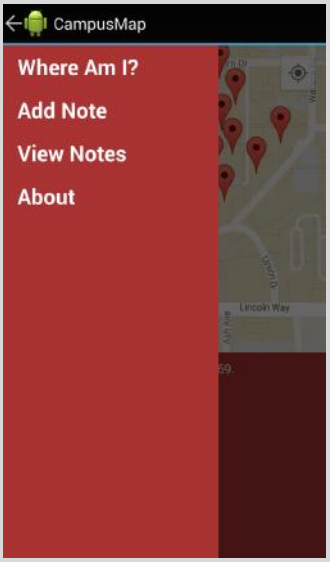
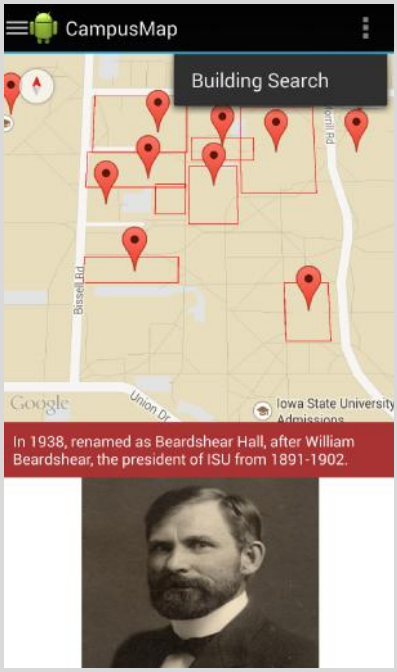
Economics

Personnel	Weekly Cost	Weeks	Total
Project Manager	1,000	3	3,000
Systems Engineer	1,250	3	3,750
Architect/Developer	1,400	3	4,200
Tester/Integrator	900	3	2,700
Grand Total	**4,550**	**-**	**13,650**
Estimated Price of Application to be Sold:			$2.00
Estimated Number of Applications Sold:			10,000
Estimated Price of Application Production:			13,650
Potential Profit for Universities/Schools:			$6,350



Prototype(s)

The following prototypes were originally designed to both follow the original layout and format of the app but with a new more consistent design following the Android design standards. The colors were chosen to match with ISU colors and the future app icon is in development and will reflect this new Material design.

Drawer Prototype	Search Interface Prototype
	
Map and Info View Prototype	**Splash Redesign Prototype**
	

Extensions and Requirements

While the previous iteration did a great deal to establish the groundwork of the application, several rather useful requirements were left incomplete, if not absent. Our proposed goal is to flesh out the planned extensions listed below.

1. Add more buildings to add notes to
2. Add a search bar to search for information on buildings
3. Make the application more intuitive
4. Add user authentication to be able to specify if user is Admin or General

Functional Requirements

1. Admins will be able to add buildings to the existing set of buildings
2. User will be able to add notes for each buildings on campus.
3. Users will be able to search for buildings and see the notes associated with designated building.
4. App will agree with the IEEE code of ethics

Non-functional Requirements

1. Users will be able to navigate the app easily
2. App will be able to handle over 100 buildings and information without compromising speed of the app
3. App will be able to be maintained from within the app itself
4. App will have state of the art security measures to protect against attackers

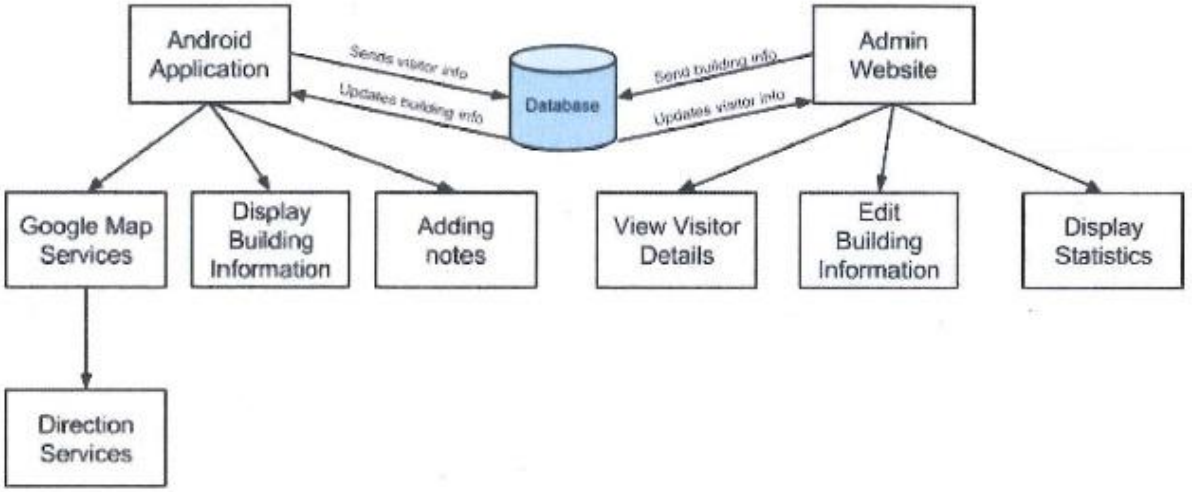
For further details, please see Appendix A: Use Cases.

Design and Modular Structure

Our design for our system incorporates the following attributes: distinctiveness, beauty, utility, features, safe, maintainable, and buildable. Each of these terms are described as follows. This project is distinct as there are no similar applications that provide aid to Iowa State tours. Our application incorporates utility by displaying map and building information as soon as the application is started. This provides the user with the main use of the application immediately. One feature that may delight the user is the ability to take notes within the application.

From a development stance, this system is safe, maintainable, and buildable. Only administrators may manipulate data within our database as a form of security. For location services, safety and security of user's location are not logged within our system, leaving that up to Google exclusively. For maintainability, our classes of code are grouped with high cohesion making it easier to make repairs or add any new features. Finally, the entire system is buildable as this is the second iteration and both have produced working products.

Below, Figure 1 is the modular structure design for our system. It describes our system from a high level how the application interacts with our website. The android application gets all of the building information from the database which is updated by an administrator on the website. Below both of these subsystems are various features or modules that a provided within that module.



Module Interface Specifications

Each of these modules have services provided and services needed. The Google Map Services provide location services as well as direction services. The Android application needs these services to satisfy a few of the highest priority requirements. In addition, the application provides displays of building information and the ability to add notes.

The test cases are defined mostly with operational profile. Every time we implemented something, we run through a few test cases of what we think a typical user would do. In addition, we plan to have a small focus group test our application while being on a simulated tour. This would ultimately test the reaction time of our application to be able to keep up with the tour. If the user or system cannot keep up, ultimately changing the design of the application or implementation would change the response time to make this process more efficient.

Some design decisions focused on administrator privileges. We discussed initially that only administrators should have access to visitor statistics, specifically how many people visit a. ha building. To extend this, our team decided that they need to be able to manipulate data in the database that directly impacts what is displayed in the application. This administrator can be anyone from a developer to an Iowa State Faculty that would want to control what is posted and displayed for various building data. It came down to what information should actually be displayed. To keep it simple, each building has a set of picture and description pairs to help

describe the building. Administrators can edit all of this data, as well as the name of the building in case- it is changed. Lastly, a major design decision focused on the flow of the Android application and where things should be placed. Ultimately we reached that the screen should only display the map and information initially, and more options can be displayed in menu form if the user swipes from left to right.

Module Implementation For Current Release

Link to the application's android apk:
https://www.dropbox.com/s/farqzxfk6e2orqq/app-release.apk?dl=0

Link to Repository
https://github.com/wzs21/Tour-Buddy.git

WBS
See Appendix B

V&V (includes measures and results)
See Appendix C

Retrospective report

History:

Team Falcon Squad was formed between April 2-7, 2015, comprised of 4 members from different teams, none of which have worked together. One of the members has previous experience on this project. Meeting times were decided upon and the team met up every school week on Tuesdays at 4:00pm in the TLA. The original requirements were decided upon by April 7th and approved on that same date. The first week was dedicated to getting used to the project and analyzing previous iterations and additions. The next 2 weeks was dedicated to development and meeting with potential users to get feedback. The remaining time was dedicated to testing to ensure that the product was ready for release.

Successes:

Despite several significant setbacks, such as scheduling conflicts, Team Falcon Squad was able to accomplish a lot in the limited amount of face to face time. Not all of members had a lot of experience with Android Studio and Git so the face time was crucial in helping others with issues and we were very successful in that sense. All team members did a good job of responding to questions proposed by other members. This built strong team chemistry and confidence to ask for help when needed. Due to the limited amount of time, all members took on the roles they were most comfortable. In doing so, we were able to stay on time with our WBS and stay focused on what is ahead rather than focusing on doing damage control from falling behind.

Shortcomings:

The project's progression through the next iteration was somewhat hampered due to several setbacks. First, with it being the third iteration of the project, there has been a lot of work done on the project already. This made it hard to get a full understanding of the application in the allotted time. Second, as mentioned earlier, there were many scheduling conflicts in meeting times so face-to-face time was limited. Although we did a good job of communicating through email, face-to-face time is very valuable in these stages of the project.

Advice:

- Meet early on in the project proposal phase to discuss each team member's capabilities and make a decision on what project to take over.
- Discuss team goals as early as possible to ensure that everyone know what is expected of them.
- Create a team schedule and deadlines early on and heavily enforce deadlines so that the team does not fall behind schedule.
- Plan for delays in production due to team members having tests and projects in other classes as well.

Appendix A: Use Cases

1 Use Case: User login

1.1 Description

This Use Case describes when the user wants to log in to use the application.

1.2 Actors

User
Database

1.3 Triggers

User is logged out, automatically asked to log in

1.4 Flow of events

1.4.1 Basic Flow

1. User opens the application
2. User inputs their info for log in
3. Database validates that the info is correct

1.4.2 Alternative Flows

1. User logs out when using the application
2. Requests user login again

1.4.3 Preconditions

1. User is not logged in
2. Credentials match those in the database

1.4.4 Postconditions

1. User is logged in and can use the application

1.4.5 Exceptional Conditions

1. Input is not valid

1.5 Issues

1. What if the user does not have a valid account? They will need to create a valid Gmail.

2 Use Case: User searches for building

2.1 Description

This Use Case describes when the user wants to find a building and the notes associated with that building.

2.2 Actors

User
Database

2.3 Triggers

User clicks on “Search” tab to search for buildings
User submits their search

2.4 Flow of events

2.4.1 Basic Flow

1. User clicks on the “Search” box
2. User inputs a designated building
3. User submits search
4. Database finds all buildings with specified key in the name
5. User selects the designated building

2.4.2 Alternative Flows

1. Same as above except user cannot find building and restarts search

2.4.3 Preconditions

1. User is logged in

2.4.4 Postconditions

1. User can view the building information
2. User can view building notes.

2.4.5 Exceptional Conditions

1. User does not have internet connection.

2.5 Issues

1. No known issues.

3 Use Case: Admin adds a building

3.1 Description

This Use Case describes when the user who is an admin wishes to add a building to the database and google maps.

3.2 Actors

User (Admin)
Database

3.3 Triggers

Admin clicks on “add building” button on homepage

3.4 Flow of events

3.4.1 Basic Flow

1. Admin clicks on the “add building” tab
2. Admin enters sufficient information needed to create building
3. If data is not sufficient, return to 2
4. Admin uploads pictures for building
5. Admin submits building to database

3.4.2 Alternative Flows

1. No alternative flows

3.4.3 Preconditions

1. User is logged in
2. User is an admin

3.4.4 Postconditions

1. Users can view the new building and its information
2. User can submit TourNotes to new building

3.4.5 Exceptional Conditions

1. Building is outside of Iowa State University bounds.

3.5 Issues

1. What if the Admin is a hacker submitting fake buildings with fake pictures?

Appendix B: WBS

There was a 4 week deadline for this project that was broken down as follows…

Work Assignment	Assigned To	Duration	Dependencies
1. Figure out which project to extend.	All	1 day (April 2-3)	None
2. Figure out project requirements for current iteration	All	2 days (April 3-5)	1
3. Write proposal for current iteration	Wan	2 days (April 5-7)	2
4. Create modules	Tyler	3 days (April 7-10)	2
5. Implement Modules	All	14 days (April 11-24)	4
5.1 Implement Extension 1. User Authentication	Tyler, Alberto	7 days (April 11-18)	4
5.1.1 Test Extension 1	Tyler, Alberto	7 days (April 18-24)	5.1
5.2 Implement Extension 2. Add new buildings	Wan	7 days (April 11-18)	4
5.2.1 Test Extension 2	Wan	7 days (April 18-24)	5.2
5.3 Implement Extension 3. Building Search	DJ	7 days (April 11-18)	4
5.3.1 Test Extension 3	DJ	7 days (April 18-24)	5.3
6. Overall Testing: Stress, Regression, User, etc.	All	3 days (April 24-27)	5
7. Write Presentation, Report	All	4 days (April 27-31)	6

Appendix C: V&V

I. Purpose

The purpose of this section is to provide information for the verification and validation plan for the TourBuddy application that Team Falcon Squad has been assigned.

II. Methods

The list provided below describes a variety of validation and verification techniques that were used for this project.

1. Requirements reviews between the clients and the architects, developers and testers of Team Falcon Squad
    a. Review the Concept of Operations by Potential Clients.
    b. Review the Requirements and Specifications by Team Falcon Squad's architects, developers, and testers within the team.
    c. Review the Requirements and Specifications by Team Falcon Squad's architects, developers, and testers with clients.
    d. Testing different scenarios in which this application would be used to find if the quality of use fits the given requirements stated.
2. Review the Design
    a. Ensure that the UI for the client is easy to navigate.
    b. Module and Structural process reviews.
3. Code Review
    a. Review the modular architecture to make sure it has not been compromised with the addition of the new code.
    b. Review code documentation to ensure it is adequate for future developers.
4. Testing
    a. Test the most probable methods / cases in which the application would be used.
    b. White Box testing of code to ensure sound code coverage.
    c. Stress test the servers for the web application. Determine current maximum tolerance limits.
    d. Regression tests to make sure application functionality from previous iterations has not been lost.

III. Schedule and Resources (For the Work Breakdown Structure, please see Appendix B: WBS.)

Review will be following our schedule that is shown on the WBS. The systems engineer, tester/integrator, and architect/developer will be responsible for the documentation and code. Organizational reviews and code conflict resolution meetings between these individuals are to be scheduled by the PM and SA team members, respectively.

IV. Measures

The V&V plan follows the measures described within the project plan. We'll also take great precautions in detecting and resolving any critical errors within our application. We hope to have a hard maximum of 5 minor defects within the third deployment of the TourBuddy extension application and 0 critical/major defects.

V. Criteria for Release

1. No critical errors
2. No major defects
3. At most 5 minor defects detected within the implementation
4. At most 7 errors within the specification or documentation
5. Code must be sufficiently and effectively commented for the sake of maintainability (Reviewed by developers)

VI. Results

1. Reviews
    a. Code Reviews
        i. Code Reviews were used to ensure that our code is clean and conforming to IEEE standards. These reviews helped us by receiving positive feedback on what is going well and what could use help. **Success**
    b. Module Reviews
        i. Module Reviews were used to ensure that Modular Design had not been compromised. These reviews were not as effective in this implementation because we only added 2 new modules and did not touch the existing ones. These reviews occurred weekly and were always efficient. **Success**
2. Testing
    a. Stress Testing
        i. Stress Tests were used to make sure that our non-functional requirement which stated that "at least 100 buildings must be handled" was true. We did this by creating mock buildings and making sure the application functionality did not suffer with the addition of more buildings. **Success**

    b. Regression Testing
        i. Regression Tests were used to make sure that previous functionality had not been altered. We did this by running old test cases on the new implementation of the app. **Success**

c. User Testing
    i. User Tests were done to make sure that application is more User-friendly and straight forward. We did this by asking potential users to use the application and if they were confused on how to use it. **Still work to be done**