**Development Plan**

**Version 1.2**

**Project Management App**

**Team A**

**CSC-355**

**Spring 2016**



02/10/2016

Author: Steven Gantz – Lead Developer

Submitted To: Dr. Joo Tan

**TABLE OF CONTENTS**

Revision History…………………………………………………………………………………..ii

1.0 Introduction……..….....…………………………….…...…………………………….……....1

2.0 Overview of Required Work…..………………………………………………………………1

2.1 Work Break Down ………………………………………………………………….1-2

2.2 Resources Needed……………………………………………………………………..2

3.0 Development Aspects…………………………………………………………………………2

3.1 Development Approach ……………….……………………………………………...2

3.2 Standard Followed ……………………………………………………………………3

3.3 Environments………………………………………………………………………….3

3.4 Version Control & Reviews ………………………………………………………..…3

3.5 Risk Management.…………………………………………………………………….4

3.6 Timeline..…….……………………………………………………………………...4-5

**REVISION HISTORY**

|  |  |  |  |
| --- | --- | --- | --- |
| Version | Author | Description | Date |
| 1.0 | Jennifer Li | I created the first draft. Added section 1 | 01/31/2016 |
| 1.1 | Steven Gantz | Filled out document  with relevant text for Section 2 and 3.0-3.5 | 02/07/2016 |
| 1.2 | Jennifer Li | Fixed the layout and placement of sections.  Filled in Timeline section 3.6 | 02/08/2016 |

**1.0 INTRODUCTION**

The purpose of this document is to outline the development process of implementing the Project Management App system by Team A. This system includes a variety of services that varies from choosing members to assigning and submitting tasks for review. Our objective for the system, is to provide a central location for organizing aspects of a created project. To achieve this objective, Team A has decided to create an automated system with modern application technology and relational databases. This new system should enhance the efficiency of successful projects by organizing all aspects of the project. Team A will achieve this objective by following the fundamental components described in this document and other documents.

**2.0 OVERVIEW OF REQUIRED WORK**

The implementation of the system will be done by developers Hector Richiez, Steven Gantz, and Tyler Mariano. Each developer will have their own tasks that will support the overall implementation of the project. The personnel will meet weekly to lay out a short progress plan for the following week and hold a verbal progress report about the prior week.

**2.1 Work Break Down**

The work needed to be done for implementation, will be distributed to each personnel.

Steven Gantz will be tasked:

* Setting up the developing environment
* Created and implemented a web service for the back end of database.
* Created and implemented a communication tool to communicate between the web service and database.
* F.R.4.0 Add members
* F.R.9.0 View overall progress
* F.R.13.0 View due dates
* F.R.7.0 Submit completed task
* F.R.10.0 Leader views members’ task progress
* F.R.11.0 View members’ own task progress
* F.R.14.0 Messaging

Tyler Mariano will be tasked:

* Implementing and populating the database
* F.R.1.0 Create an account
* F.R.3.0 Create a project
* F.R.12.0 Accept project invitations
* F.R.6.0 Assign task
* F.R.15.0 Set reminders
* F.R.18.0 Create custom roles

Hector Richiez will be tasked:

* Creating a implementing the User Interface for login and leader screen
* F.R.2.0 Login
* F.R.19.0 Send invites to members
* F.R.20.0 View all members
* F.R.16.0 Logout
* F.R.8.0 Review submitted task
* F.R.5.0 Remove members
* F.R.17.0 Uploading Documents

**2.2 Resources Needed**

To implement this project, our team will need specific tools and resources. The tools and resources are:

* Student owned PC or Mac capable of running Android Studio and interfacing with Android-powered devices.
* A copy of Google’s java style guide for reference.
* Copies of open source external libraries used within the project.
* An Android-powered device for app deployment testing.
* A copy of the SQLite3 database system and command line tools.

**3.0 DEVELOPMENT ASPECTS**

The system will be implemented using Google’s Android Java language and will function with a SQLite3 database. Java is an object oriented programming language developed by Oracle and forked by Google to use with their Dalvik virtual machine. The database supports object oriented functionality and is a relationship database. The user interface as well as program functionality will be supported through the Java code. Data storage and maintenance will be sustained through the SQLite database and web hosting by Vultr. The completed system and all components will be loaded and hosted on the client’s Android device.

**3.1 Development Approach**

Developers will be using the Iterative software methodology for project implementation. Developers will also be breaking the project into smaller segments to allow for ease of change during the process. General user testing by internal members will occur throughout development, which increases the likelihood of positive user acceptance testing of the final implementation.

* 1. **Standards Followed**

Development team will primarily write class structures to handle program logic within Android activities. Class structures will also be used on the server development to create a solid foundation between an Android application, and a locally hosted database.

Programming style is done in an Object-Oriented fashion. This is coupled with using Android best practices to develop readable, maintainable, and re-usable code throughout the life of the project’s development.

Naming Standard – Following the Google Java Style Guide

Indentation and spacing – Following the Google Java Style Guide, all whitespace characters will be spaces only. There will be no tabs in the application.

Function and method size – Methods will be cohesive and have low coupling. Method size will be kept within 50 lines.

File naming – All files are named after the class they contain as per Java standard style guide.

Development is being done by the team with the Object-Oriented paradigm in mind. Code is formatted based on Google’s coding conventions published on their website. These conventions outline file organization, indentations, comments, statements, and best practices.

Comments come in two flavors; inline documentation comments, and java documentation comments. These are referred to as implementation comments, and Javadoc respectively. The team will add implementation comments where necessary to explain to maintainers any out of the ordinary coding choices or constructs. Javadoc is being done at a high level, with each method and module receiving its own share of documentation as listed in Google’s Java Style documentation.

**3.3 Environments**

The team is using Android Studio’s built in debugging and automation tools to allow for effective change management within the source code. Android Studio has a refactoring interface that allows for individual changes to be resonated through the code and documentation. Bundled with Android Studio also, is an Android-Virtual-Device debugging system. This monitors deployments and reports back to the IDE any bugs or issues, allowing for quick turnaround on find software issues.

**3.4 Version Control & Reviews**

The team will be using GitHub for version control during development. Each user has full commit privileges within the repository. Version control is utilized using third party interfaces such as Git for Desktop and SourceTree. GitHub’s internal issue and milestone system is being utilized for questions, bugs, and enhancements brought forward by members of the team.

**3.5 Risk Management**

Due to each developer using a different device to test deployment, there is the possibility of compatibility issues between devices. This is mitigated by testing deployment on each device before moving forward with development.

By using the iterative software methodology, the team may create inflexible designs that have to be altered down the line. This may cause longer bug fixing times as each iteration is more focused. This can be mitigated by designing interfaces for each project feature and using those interfaces to implement the aforementioned features.

**3.6 Timeline**

The implementation of the project will be broken down into two Phases. Each phase will be broken down further into three sprints. After the first phase, a prototype will be delivered within the first week of March. The final product will be delivered after the final phase, within the last week of April.

Phase 1 delivery deadline is within the first week of March.

|  |  |  |  |
| --- | --- | --- | --- |
| Phase 1 | Task | Due | Milestone |
| Sprint 1  Design, Back end, and  Functionality  Start: 1/26  End: 2/13 | Setting up the developing environment | 1/26 |  |
| Web service | 2/8 |  |
| Communication tool | 2/9 |  |
| Database Implementation | 2/8 |  |
| UI Login | 2/8 |  |
| UI Screen Designs | 2/12 |  |
| F.R.1.0 Create an account | 2/12 |  |
| F.R.2.0 Login | 2/12 |  |
| F.R.3.0 Create a project | 2/13 |  |
| Sprint 2  Functionality  Start: 2/14  End: 2/20 | F.R.4.0 Add members | 2/15 |  |
| F.R.19.0 Send invites to members | 2/16 |  |
| F.R.20.0 View all members | 2/17 |  |
| F.R.12.0 Accept project invitations | 2/18 |  |
| F.R.16.0 Logout | 2/19 |  |
| F.R.9.0 View overall progress | 2/20 |  |
| Sprint 3  Functionality  Start: 2/21  End: 2/27 | F.R.13.0 View due dates | 2/22 |  |
| F.R.6.0 Assign task | 2/23 |  |
| F.R.7.0 Submit completed task | 2/24 |  |
| F.R.8.0 Review submitted task | 2/25 |  |
|  |  |  |
|  |  |  |

Phase 2 delivery deadline is within the last week of April.

|  |  |  |  |
| --- | --- | --- | --- |
| Phase 2 | Task | Due | Milestone |
| Sprint 1  Functionality  Start: 3/14  End: 3/31 | F.R.5.0 Remove members | 3/16 |  |
| F.R.10.0 Leader views members’ task progress | 3/18 |  |
| F.R.11.0 View members’ own task progress | 3/20 |  |
| F.R.15.0 Set reminders | 3/22 |  |
| F.R.18.0 Create custom roles | 3/24 |  |
| Reliability | 3/31 |  |
| Sprint 2  Functionality  Start: 4/1  End: 4/17 | F.R.14.0 Messaging | 4/10 |  |
| F.R.17.0 Uploading Documents | 4/15 |  |
| Usability | 4/17 |  |
| Scalability | 4/17 |  |
| Testing | N/A |  |
| Testing | N/A |  |
| Sprint 3  Functionality  Start: 4/18  End: 4/30 | Testing | N/A |  |
| Testing | N/A |  |
| Testing | N/A |  |
| Presentation | 2/20 |  |
| Poster Board | 4/25 |  |
|  |  |  |