**Development Plan**

**Version 1.0**

**Project Management App**

**Team A**

**CSC-355**

**Spring 2016**



02/12/2016

Author: Steven Gantz – Lead Developer

Submitted To: Dr. Joo Tan

**TABLE OF CONTENTS**

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

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

2.0 Personnel and Resources………………………………………………………………………2

2.1 Personnel………………………………………………………………………………

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

3.0 Development Aspects………………………………………………………………………..

3.1 Development Approach and Standards……………………………………………..

3.2 Timeline………………………………………………………………………………..

3.3 Characteristics…………………………………………………………………………

3.4 Programming Style …………………………………………………………………….

3.5 Coding Guidelines……………………………………………………………………

3.6 Comments………………………………………………………………………………

3.7 Version Control…………………………………………………………..

3.8 Performance Optimization……………………………………………………………

3.9 Refactoring and Debugging …………………………………………

3.10 Risks and Contingencies……………………………………………………………..

**REVISION HISTORY**

|  |  |  |  |
| --- | --- | --- | --- |
| Version | Author | Description | Date |
| 1.0 | Jennifer Li | I created the first draft. Added section 1 | 1/31/2016 |
| 1.1 | Steven Gantz | Filled out document  with relevant text | 2/7/2016 |

**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 PERSONNEL AND RESOURCES**

The implementation of the system will be done by developers Hector Richiez, Steven Gantz, and Tyler Mariano. 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 Personnel**

Hector Richiez

Steven Gantz

Tyler Mariano.

**2.2 Resources Needed**

* 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**

Stuff.

**3.1 Development Approach and Standards**

Developers will be using the Iterative software methodology for project implementation. Developers will break 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.

**3.2 Timeline**

The implementation of the project will be broken down into two sprints. After the first sprint, a prototype will be delivered in the week before spring break. The final product will be delivered after the final sprint during the week of April 30th

|  |  |  |
| --- | --- | --- |
| Phase | Task | Due |
| Sprint 1 | Account Integration |  |
|  |  |

|  |  |  |
| --- | --- | --- |
| Phase | Task | Due |
| Sprint 2 |  |  |
|  |  |

**3.3 Characteristics**

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.

**3.4 Programming Style**

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.

**3.5 Coding Guidelines**

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.

**3.6 Comments**

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.

**3.7 Version Control**

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.

**3.9 Refactoring and Debugging**

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.

**3.10 Risks and Contingencies**

* 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.