**CS673S16 Software Engineering**

**Team 3 - ProTeam**

**Project Proposal and Planning**



**Revision history**

|  |  |  |  |
| --- | --- | --- | --- |
| **Version** | **Author** | **Date** | **Change** |
| **1.0** | **Danny Chiu** | **2/9/2016** | **Added content** |
| **2.0** | **Danny Chiu** | **3/1/2016** | **Revised content** |
| **3.0** | **Danny Chiu** | **3/19/2016** | **Revised content** |
| **4.0** | **Danny Chiu** | **4/9/2016** | **Revised content** |
| **5.0** | **Danny Chiu** | **4/22/2016** | **Revised content** |

|  |  |  |  |
| --- | --- | --- | --- |
| Team Member | Role(s) | Signature | Date |
| Danny Chiu | -Team Leader  -QA Leader  -Requirement Leader | *Danny Chiu* | 2/9/2016 |
| Brian Kabuye | -Implementation leader  -App Developer |  |  |
| ChihYung Wu | -Design Leader  -App Developer |  |  |
| Ayodele Awoleye | -Configuration Leader  -Server Developer |  |  |
| Josh Wildey | -Environment and Integration Leader  -Server Developer |  |  |

[Overview](#h.g6igqliy7rm)

[Related Work](#h.bf21eadgjj29)

[Proposed High level Requirements](#h.rgyo4hi9stmq)

[Management Plan](#h.ts358bsdtbcv)

[(For more detail, please refer to SPMP document for encounter example)](#h.ds8oyr75pnh1)

[Process Model](#h.nfqtbpjve81e)

[Objectives and Priorities](#h.79nacltvk6xe)

[Risk Management (need update constantly)](#h.1xctjagvqeqx)

[Monitoring and Controlling Mechanism](#h.og4uaqur9s5i)

[Schedule and deadlines (need update constantly)](#h.bjdb6xshxp8a)

[Quality Assurance Plan](#h.quttytfw209u)

[(For more detail, please refer to SQAP document for encounter example)](#h.irx41slwwvdz)

[Metrics](#h.561b2z2r7ga0)

[Inspection/Review Process](#h.qqf0q1klr8qc)

[(e.g. describe what are subject to review, when to conduct review, who do the reviews and how ?)](#h.dvmuaxur53du)

[Testing](#h.1dibojfal58t)

[Defect Management](#h.5amsh8h9f0c7)

[Configuration Management Plan](#h.j5uvivmxqcsp)

[Configuration items and tools](#h.5tla0rg2sikc)

[Change management and branch management](#h.q14gxckrt74h)

[Code commit guidelines](#h.d17l9gn79d58)

[References](#h.pd9euov6m4du)

[Glossary](#h.ty3i2nqffhtc)

# Introduction

**1.1. Purpose**

This document is intended to give an outline of our project proposal. It describes the proposed high level requirements of this project. Documents the risk management, quality management, and test management plans.

**1.2 Overview**

The Project Management Tools website ([www.3blueprints.com](http://www.3blueprints.com)) was developed to give software development teams a means of managing and tracking progress of the software application development. The three main functionalities that it provides are requirements tracking, issue tracking and chat rooms. Requirements tracking provides a way to generate user stories that define the required functions of the software being developed. The issue tracker provides a way to keep track of found software bugs. The chat rooms provide an easy method of communication amongst development teams and a way to share ideas, comments and concerns.

The ProTeam Android application is a mobile application that adds an additional interface to the website that is intended to provide all the same capabilities as the website. This mobile application requires communication with the web server through a RESTful API to give it access to all the same information that the website provides through the browser. This document is therefore broken up into two major sections each describing the Android app itself changes made to the server codebase.

**1.3 Scope**

The initial design effort for the ProTeam Android App focused on the Requirement Tracker aspect of the website and therefore this document will be focused there however the dashboard of the App GUI will contain buttons for the other two functions for future development.

The App uses the RESTful API of [www.3Blueprints.com](http://www.3blueprints.com) and so the focus of the web server development will be focused on that aspect.

# Related Work

There are many project management tool mobile applications currently available. A few of them include Pivotal Tracker, Asana, Wrike, etc. They provide a way for users to keep track of the progress and the requirements of each project.

# Proposed High level Requirements

* 1. Functional Requirements
     1. Essential Features
        1. user login screen
        2. view list of different projects
        3. option to create and modify user stories.
     2. Desirable Features
        1. different user groups have different control permissions (some users can delete, edit and create, while some users can only view, etc.)
        2. user stories are ranked so the most important ones are at the top.
     3. Optional Features
        1. ability to chart progress for each project.
        2. ability to send the description of the user stories by email.
  2. Nonfunctional Requirements
     1. user friendliness - have a functional user interface for every screen. Each user should be able to log-in, view projects they are in and be able to add/modify user stories. They should be able to do all that through the user interface.
     2. application should be easily deployable through use of APK.
     3. security processes should be enforced such as minimal password length should be 8.
     4. users can access application 99% of the time (as long as server is up).

# Management Plan

# (For more detail, please refer to SPMP document for encounter example)

## Process Model

* + 1. We will use the Scrum process model which is an iterative and incremental approach to product development. There are three major deadlines after the initial planning phase. Our sprints will be about three weeks long each to make sure we meet each deadline. The focus will be on having a functional mobile application.

## Objectives and Priorities

* + 1. We have three main objectives for this project:
       1. Have a working user information system (ability to sign-up and log-in) to the mobile application
       2. Have a completed mobile application that performs the basic requirement tracker tool functions (Ability to view list of projects associated with each user, view iteration list, view icebox list, modify/add new user stories).
       3. Data/information are pulled/stored in Django server.

## Risk Management (need update constantly)

* + 1. The 12 Top Risk Associated with Android Application Project, probabilities of happening and their impacts.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| # | Risk Item | Probability (%) | Risk Priority | Impact (s) on Project |
| 1 | Loss of a team member | 15% | High | Difficult for his role to be replaced and delaying of project |
| 2 | Missing delivery deadline(s) | 14% | High | Losing customer and reputation $$ (Bad grade!) |
| 3 | Source code loss/server down | 5% | Low | Failing project |
| 4 | Failing to meet 3 main requirements | 10% | Medium | Poor quality project |
| 5 | Malware attacks | 7% | Medium | \* Application data loss/stolen or corruption  \* Unauthorized access  \* DoS (Denial of access) to application |
| 6 | Difficult integration work | 8% | Medium | \* Non-working application |
| 7 | Broken authentication and session Id | 7% | Medium | \* Unauthorized access and other security risks |
| 8 | Compatibility with all android device (hardware & software) | 4% | Low | Less customers == less money $$ |
| 9 | Customer change project requirements (+/-) | 5% | Low | additional resources (time and effort) needed |
| 10 | Overriding other people’s work by not having the latest versions of code | 3% | Low | Bad practice that could lead to non-functioning code. |
| 11 | Lack of exposure to and or experience with technologies | 7% | Medium | Poor quality application |
| 12 | Being overwhelmed by work in other classes | 15% | High | Withdraw from class/ Putting less effort on this assigned task(s) |

* + 1. Response/Action Plan for the top 12 Risks

|  |  |  |
| --- | --- | --- |
| # | Risk Item | Response/Action Plan for Risk  (Risk Management Technique) |
| 1 | Loss of a team member | -Insure any files worked on are added to Google drive and GitHub.  -Divide up the responsibilities among other team members. |
| 2 | Missing delivery deadline(s) | Always check during Scrum meeting that project is on track and plan more time for debugging/Testing |
| 3 | Source code loss/server down | Have 3 backup sources code in 3 different locations. How reliable are our storage? It host often |
| 4 | Failing to meet 3 main requirements | In each of our iterations we should make sure application the main requirements are met. |
| 5 | Malware attacks | Follow modern application security best practices and standards. Use Mobile Application Security Vulnerabilities Scanners like AppScan by IBM OR WebInspect by HP. |
| 6 | Difficulty integration work | Increase communication and integrate often |
| 7 | Broken authentication and session Id | ·Employ strong hashing and encryption algorithm when dealing with sensitive data like personal data (UN, PW, Profile information).  ·Enforce strong authentication logic , strong username and password policy  ·Do not send session id on URL header, but add it into body of POST. |
| 8 | Compatibility with all android device (hardware & software) | Test compatibility of our application with all available recent android devices (tablets/mobile) |
| 9 | Customer change project requirements (+/-) | Following Agile methodology few requirement modification should be possible. |
| 10 | Overriding other people’s work not having the latest versions of code | ·Version control program - GitHub  ·Configuration leader and team lead will make sure this risk is well managed. |
| 11 | Lack of exposure to and /or experience with technologies | Currently most members of our team are learning Android technologies since only 3 members who work with Android before. |
| 12 | Being overwhelmed by work in other classes | Communicate early when members are overwhelmed with other work, so other members can help out. |

## Monitoring and Controlling Mechanism

* + 1. Our main method of communication is through Slack.
    2. Most documents will be shared through Slack.
    3. Google Docs will store all of our documents (Weekly Reports, Meeting Minutes, SPPP, SDD, Testing documentation, and presentation slides).
    4. Pivotal Tracker will be used for tracking status and keeping track of all features that we would like to include in the mobile application.

## Schedule and deadlines (need update constantly)

* + 1. Meet the requirements listed below by each of the major deadlines:
       1. Initial planning phase: (2/10/2016)
       2. Iteration 1: (3/2/2016)
       3. Iteration 2: (3/30/2016)
       4. Final iteration: (4/27/2016)
    2. Iteration 1 Requirements
       1. User log-in interface created
          1. able to fetch and pass user information to Django backend database.
       2. Simple design layout for requirement tracker (view list of projects and list of user stories).
    3. Iteration 2 Requirements
       1. Working functions to create new stories, edit and delete user stories.
       2. The user story data are pulled/updated on the Django server. .
    4. Final Iteration Requirements
       1. Each user story can be clicked on to view more information on.
          1. Ability to modify the information when viewing the story.
       2. Each user can view icebox where it holds all the user stories that have not been put into iteration. Can also modify user stories on that same screen.

# Quality Assurance Plan

# (For more detail, please refer to SQAP document for encounter example)

## Metrics

* + 1. The quality of our application will be measured based on the number of features that the application is able to complete.
       1. Sign-in passes if username/password is correct.
       2. List of Project screen passes if projects that show are associated with the user.
       3. Project description screen passes when detail page of project shows up.
       4. Iteration screen passes if user stories that show are associated with the project.
       5. Icebox screen passes if user stories that are not in the Iteration screen shows on this screen.
    2. The number of times the application fails to do what is supposed to do.
       1. Sign-in fails if username/password is incorrect.
       2. List of Project screen fails if projects that show up are not associated with the user.
       3. Project description screen fails when detail page of project shows up.
       4. Iteration screen fails if user stories that show up are not associated with the project.
       5. Icebox screen fails if user stories that are not in the Iteration screen shows on this screen.

## Inspection/Review Process

* + 1. For every iteration, each developer will peer review each other’s code to make sure that there are no bugs/errors.
    2. Each developer will also create their own test programs to make sure their function is working.
    3. During each iteration, review of the features should be done once the features have been implemented/created.
    4. Focus is to resolve any bugs/issues before starting the next iteration.
    5. After a commit, developer should not push code if it breaks the current build.

## (e.g. describe what are subject to review, when to conduct review, who do the reviews and how ?)

## Testing

* + 1. For every iteration, we will create unit tests to make sure that the feature we are trying to implement is working.
    2. QA leader will be responsible for leading the testing
       1. Testing will consist of both black box testing and white box testing.
          1. Goal is to be able to identify bugs/issues.
          2. Make sure it does what it supposed to do (I.E - log-in only with correct information, etc.)
    3. Proper documentations of test runs (what is issue, how do we replicate this issue, what should have been correct?).

A separate document about testing result should be linked here.

## Defect Management

(e.g. describe the criteria of defect, also in terms of severity, extend, priority, etc. The tool used to management defect, actions or personnel for defect management)

* + 1. A defect is defined as something that is not working as intended. The focus is on the tasks the application is suppose to do but does it incorrectly.
    2. Similar to a bug/issue tracker, we would record any issues that we noticed while testing the application.
    3. We would category the defects and prioritize them.
       1. An example we had was the list of projects were duplicated every time a user goes to a different screen and back to list of projects.

# Configuration Management Plan

(For more detail, please refer to SCMP document for encounter example)

## Configuration items and tools

* + 1. version control: GitHub
    2. coding: Android Studio, Eclipse
    3. Backend: Django with REST framework.
    4. Documentation: Google Drive
    5. Requirement: Pivotal Tracker

## Change management and branch management

* + 1. the configuration leader should ensure that each member has created separate branches when making updates.

## Code commit guidelines

* + 1. codes should not be committed until they are fully tested.
    2. codes must not break program.
    3. codes should not affect previous functionalities.

# References

Lecture notes from Professor Zhang (<https://learn.bu.edu>)

<http://www.ibm.com/developerworks/library/se-owasptop10/>

[https://www.owasp.org/index.php/OWASP\_Mobile\_Security\_Project](https://www.owasp.org/index.php/OWASP_Mobile_Security_Project#tab=Top_10_Mobile_Risks)

<http://developer.android.com/training/articles/security-tips.html>

# Glossary