Software Requirements Specification

For Mobile Application PlanItRight

Team PlanItRight

September 21, 2021

Version 1

**Team Members:**

**Harleen, Pokai, Eric**

Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Version** | **Date** | **Name** | **Description** |
| 1 | Sept 21, 2021 | SRD V-1 | Initial SRD Document |
|  |  |  |  |

Contents

[1 INTRODUCTION 4](#_Toc83045939)

[1.1 Project Overview 4](#_Toc83045940)

[1.2 Project Goals and Objectives 4](#_Toc83045941)

[2 GENERAL DESIGN CONSTRAINTS 5](#_Toc83045942)

[2.1 Software Environment 5](#_Toc83045943)

[2.2 Hardware Environment 5](#_Toc83045944)

[2.3 Other Constraints 5](#_Toc83045945)

[3 NONFUNCTIONAL REQUIREMENTS 6](#_Toc83045946)

[3.1 Operational Requirements 6](#_Toc83045947)

[3.2 Performance Requirements 6](#_Toc83045948)

[3.3 Security Requirements 6](#_Toc83045949)

[3.4 Other Requirements 6](#_Toc83045950)

[4 FUNCTIONAL REQUIREMENTS 7](#_Toc83045951)

[4.1 Application Overview 7](#_Toc83045952)

[4.2 Users Use Cases 8](#_Toc83045953)

[Bibliography 10](#_Toc83045954)

1 INTRODUCTION

## Project Overview

This document presents a detail description of the Plan-It-Right. It will explain the purpose and features of the Plan-It-Right, the interfaces of the Plan-It-Right, what Plan-It-Right will do, the constraints under which it must operate and how Plan-It-Right will react to external stimuli. This document is intended for both the stakeholders and the developers of the Plan-It-Right and will be proposed to our course instructor Xing Liu for its approval.

## Project Goals and Objectives

In this project, our team want to create a mobile application named Plan-It-Right. Plan-It-Right will be designed to helps users to manage their projects or to-do-lists. Users can easily keep track of their tasks and deadlines in a central location. This application will give the users a sense of accomplishment and motivate the users to keep going when things get tough!

There are other project management applications in the market, but most of them demand a heavy monetary subscription. For example, Basecamp, a well-known project application, asks for a monthly subscription of $99. We want to make our project management application free to our users.

We want to make this application simple and intuitive to use for everyone. Plan-It-Right will be great for starting personal projects, as well as for organizing projects among small teams.

2 GENERAL DESIGN CONSTRAINTS

## 2.1 Software Environment

Our team will use the React-Native and Expo framework to develop the application for this project. We aim to design our Plan-It-Right application to run on Android 5.0 (API 21) or iOS 11.0 or newer (facebook/react-native, n.d.). React Native is a popular JavaScript-based mobile app framework that allows developers to build natively rendered mobile apps for both iOS and Android.

As per backend, we will use Firebase to store our application data and to authenticate application users.

## 2.2 Hardware Environment

The Plan-It-Right application will run on any devices that run Android 5.0 (API 21) or iOS 11.0 or newer.

Our mobile application will connect to a cloud-based database - Firebase. Therefore, we do not need to set up any physical server to handle API calls or data storage.

## 2.3 Other Constraints

Plan-It-Right will not support data caching. Users will not be able to use our application if he or she does not have an internet connection.

Our project will be using the free quote offered by Cloud Firestore. The free quote limits data traffic at 50,000 document reads per day, 20,000 document writes and deletes per day (Firebase, n.d.).

As for application development using the React-Native and Expo framework, there might be some native features that are available in Android may not be available in iOS. If we want to make sure our application work well in both Android and iOS, we will need to limit our use of features that will work in both platforms.

3 NONFUNCTIONAL REQUIREMENTS

## 3.1 Operational Requirements

* The application will run similar to the other apps, starting with signup or Login. The user may save the password in order to use to regularly without entering details every time.
* The signup for this application will be simple, only email and a password with be required.
* This application will need internet access at minimum 3G speed.
* Monitoring the application will be easy as most users are already familiar with to-do lists.

## 3.2 Performance Requirements

* Most of the actions will take place on same page, therefore, any updated information will be shown to the user within 5 seconds.
* If an action is directing user to another page, the request should be processed within 10 seconds.
* This application will only accommodate 300 users at one time.

## 3.3 Security Requirements

* The application will store user’s personal data, like what are the tasks and who are they assigned to, however, our application will keep your data safe and secure as the data will not be shared to anyone.

## 3.4 Other Requirements

* Users will be able to access their account 99% of the time without failure.
* For better use of this application, it will be recommended that users should try to use it solo at first (by assigning tasks to themselves) than working in a group.
* The application is user friendly and can be used by different aged groups for different tasks.

4 FUNCTIONAL REQUIREMENTS

## 4.1 Application Overview

Diagram

Description automatically generated

Figure 1 Application Overview

Figure 1 shows an overview of the PlanItRight application architecture. Our PlanItRight application provides an interface for application users to create, manage and collaborate projects. The mobile application will be connected to a cloud-based backend – Firebase, which will provide services such as data storage and authentication for our PlanItRight application.

## 4.2 Users Use CasesDiagram Description automatically generated

Figure 2 Applications Use Cases

Figure 2 shows all use cases of our PlanItRight mobile application. There are two levels of user privileges: project owner and project member. Project owner has a higher privilege than project member. For example, only project owner can assign project task to other project members. Each project task can have a status of being backlog, in process or completed. Only project owner or task assignee can modify the status of a project task.

The following provide a brief explanation of each use case in Figure 2:

1. All application users must sign up and login.
2. Only project owner can create & terminate projects.
3. Only project owner can add project description.
4. Only project owner can add project deadlines.
5. Only project owner can add project requirements.
6. Only project owner can manage project participants.
7. Only project owner can promote any project member to become a project owner. This will allow a project member to receive a higher level of privilege.
8. Only project owner can assign tasks to another participants.
9. Only project owner and the assignee of the project task can modify the status of the project task.
10. All members in a project can opt to receive notification for any project updates.
11. All members in the project can view the project progress and statistics.

# Bibliography

*facebook/react-native*. (n.d.). Retrieved from GitHub: https://github.com/facebook/react-native#-requirements

*Firebase*. (n.d.). Retrieved from https://firebase.google.com/docs/firestore/quotas: https://firebase.google.com/docs/firestore/quotas