
Software Requirements Specification

for

Payment Tracker

Version 1.1

Prepared by

**Sindhura.L (01JST17IS050)
Spoorthi.S (01JST17IS062)**

29/02/2020

Table of Contents

| | |
|--|-----------|
| Table of Contents | ii |
| 1. Introduction | 1 |
| 1.1 Purpose..... | 1 |
| 1.2 Document Conventions..... | 1 |
| 1.3 Intended Audience and Reading Suggestions..... | 1 |
| 1.4 Product Scope | 2 |
| 1.5 References | 2 |
| 2. Overall Description | 2 |
| 2.1 Product Perspective..... | 2 |
| 2.2 Product Functions | 3 |
| 2.3 User Classes and Characteristics | 3 |
| 2.4 Operating Environment..... | 3 |
| 2.5 Design and Implementation Constraints..... | 3 |
| 2.6 Assumptions and Dependencies..... | 4 |
| 3. External Interface Requirements..... | 4 |
| 3.1 User Interfaces | 4 |
| 3.2 Hardware Interfaces | 4 |
| 3.3 Software Interfaces | 4 |
| 3.4 Communications Interfaces | 5 |
| 4. System Features..... | 5 |
| 4.1 Reminder | 5 |
| 4.2 Store details in the server..... | 6 |
| 5. Other Nonfunctional Requirements..... | 6 |
| 5.1 Performance Requirements..... | 6 |
| 5.2 Safety Requirements | 7 |
| 5.3 Security Requirements | 7 |
| 5.4 Software Quality Attributes | 7 |
| Appendix A: Glossary | 7 |
| Appendix B: Analysis Models..... | 8 |

Revision History

| Name | Date | Reason for Changes | Version |
|---------|------------|-----------------------|---------|
| Authors | 27/02/2020 | Initial Version | 1.0 |
| Authors | 29/02/2020 | System Feature errors | 1.1 |

1. Introduction

1.1 Purpose

The purpose of this document is to give a detailed description of the requirements for the “Payment Tracker” app which tracks and notifies the user about the payments due, by the user or to the user. It will illustrate the purpose and complete declaration for the development of system. It will also explain system constraints, interface and interactions with other external applications. This document is primarily intended to be proposed to a customer for its approval and a reference for developing the first version of the system for the development team.

1.2 Document Conventions

| Term | Definition |
|------|---|
| User | Someone who interacts with the mobile phone application |
| GUI | Graphical User Interface |
| PTS | Payment Tracking System |
| ADT | Android development Tools |
| IDE | Integrated Development Environment |
| AVD | Android Virtual Device |

1.3 Intended Audience and Reading Suggestions

This document is to be read by the development team, the project managers, marketing staff, testers, and documentation writers. The software engineer/Developer

and project managers need to become intimately familiar with the SRS. Others involved need to review the document.

Testers need an understanding of the system features to develop meaningful test cases and give useful feedback to the developers. The developers need to know the requirements of the software product they need to build.

1.4 Product Scope

The scope of this product is to help people who have to manage a lot of periodic payments from various sources. It also aids to track the payments due by the user. This product avoids the problem of repetitively setting reminders for various tasks. The application takes details about the payment and periodic interval after which the user will be notified. It also stores the related images such as bond in encrypted form for the user's reference.

1.5 References

- IEEE Software Engineering Standards Committee," IEEE Std 830-1998, IEEE Recommended Practice for Software Requirements Specifications"
- <https://krazytech.com/projects/sample-software-requirements-specificationsrs-report-airline-database>
- <https://www.scribd.com/document/204773379/Srs-Example-2010-Group2>
- <http://utdallas.edu/~chung/CS4351/syllabus.htm>

2. Overall Description

2.1 Product Perspective

In the existing android applications, we generally have to set different reminders in the mobile phone for our schedule. These reminders only notify the user once on the intended date but for payment related activities periodic reminders are necessary. So, we have to set the remainder again and again. This is a tedious task. Our product avoids all these repetitive tasks and helps the users to keep track of their payments in an organized way.

2.2 Product Functions

- The app requests login credentials of the user to access the contents of the app providing a layer of security.

- After logging in the user can set periodic reminders regarding various payments/events or view details of the previously set reminders.
- It gives notifications to the user along with the details entered by them on the day of the event.
- The app also allows the user to save photos related to the payment/event like bonds along with the reminder in encrypted format. Thus, keeping sensitive information like account number, policy number, etc safe.
- The details regarding the payment along with the image is stored in a server thus the data will be safe from any system malfunctions.

2.3 User Classes and Characteristics

- Insurance policy holders can use the app in order to set periodic reminders, which notify them about the amount and date of premium payment.
- Fixed deposit holders can use the app to set the reminder about the maturity amount and date.
- Users who need to pay or collect rent can also use the app to set reminders. This helps users who need to collect rent/payments periodically from large number of parties.
- Users can set even other general-purpose reminders.
- App also provides a facility for the users to store the images such as bonds etc related to reminders in encrypted format safely.

2.4 Operating Environment

Hardware Requirements:

- The absolute minimum for Android were originally a 200 MHz processor, 32 MB of RAM, and 32 MB of storage.

Software Requirement:

All versions of Android which are compatible,

- Android 10
- Android 9.0 "Pie"
- Android 8.0 and Android 8.1 "Oreo"
- Android 7.0 and Android 7.1 "Nougat"
- Android 6.0 "Marshmallow"
- Android 5.0 "Lollipop"
- Android 4.4 "KitKat"
- Android 4.3, Android 4.2, Android 4.1 "Jelly Bean"

2.5 Design and Implementation Constraints

- The internet connection is a constraint for the application. Since the application fetches data from the database over the internet, it is crucial that there is an internet connection for the application to function.
- Mobile application is constrained by the capacity of the database.

2.6 User Documentation:

Detailed document describing the usage of the product shall be delivered along with the product after the completion of the product.

2.7 Assumptions and Dependencies

- One assumption about the product is that it will always be used on mobile phones that have enough performance. If the phone does not have enough hardware resources available for the application, for example the users might have allocated them with other applications, there may be scenarios where the application does not work as intended or does not work at all.
- If the app is not well supported, for example: not providing required permissions then it may not work as intended.
- Another assumption is that the product is installed in an android operating system. Other operating system example: iOS doesn't support the app
- It is assumed that the user will have reasonable internet connection facility, since app uses firebase, internet connection is necessary.

3. External Interface Requirements

This section provides a detailed description of all inputs into and outputs from the system. It also gives a description of the hardware, software and communication interfaces and provides basic prototypes of the user interface.

3.1 User Interfaces

A first-time user of the mobile application should see the sign-in page when he/she opens the application. If the user has not registered, he/she should be able to do that on the sign-in page. If the user is not a first-time user, he/she should be able to login using his registered password and username.

After logging in the user's dashboard page appears where all the previous reminders are displayed. This page has a new reminder button using which user can set new reminders. The button navigates to a new page where the new reminders details are taken such as date, type of reminder, photo (optional) etc. The submit button in this page creates new reminders as per the details provided.

3.2 Hardware Interfaces

Since Mobile application does not have any designated hardware, it does not have any direct hardware interfaces. The System clock in the mobile phones and hardware

connection to the database server is managed by the underlying operating system on the mobile phones and the server.

3.3 Software Interfaces

- Operating System ---- Android operating system for its best support and user-friendliness.
- Database ---- Firebase, used to save user's information
- IDE ---- To implement the project we use android studio3.5 framework (ADT), which provides integrated android developer tools for development and debugging.
- Language ---- Java

3.4 Communications Interfaces

When app connected to Firebase, it does not connect through normal HTTP. It connects through a WebSocket. WebSocket's are much, faster than HTTP. We don't have to make individual WebSocket calls, because one socket connection is plenty. All of your data syncs automatically through that single WebSocket as fast as your client's network can carry it.

4. System Features

4.1 Remainder

4.1.1 Description and Priority:

The users should be able to set reminders according to their needs. They should also be able to view, update, delete the reminders that they have previously set.

4.1.2 Stimulus/Response Sequences:

The system shall request for the following details regarding the payment, from the user to set the reminder.

- Name
- Description
- Image
- Date
- Period

The system should notify the user on the specified date. It should also notify the user repeatedly, separated by a time interval specified by the user till the user deletes the reminder.

4.1.3 Functional Requirements:

- Add reminder
The users should be able to set reminders according to their needs by providing required details.

- **View reminder**
The users should be able to view all the reminders previously set by them after logging in. The system should display the details of the reminders like the name, description, image, etc.
- **Update reminder**
The users should be able to update details specified in a particular reminder. The system should allow the user to perform operations like changing the description, image, date etc.
- **Delete reminder**
The users should be able to delete reminders when they no longer want to be notified regarding certain payments. Only the intended reminder should be deleted completely from the firebase.

4.2 Store details in the server

4.2.1 Description and Priority:

The details entered by the users should be stored in the server so that they can access their data from any device.

4.2.2 Stimulus and Response:

The details entered by the user should be stored in the server and the user should be able to access them whenever required.

4.2.3 Functional Requirement:

- Access the data from any device.
- Even if the user uninstalls the app data should not be lost.

5. Other Nonfunctional Requirements

5.1 Performance Requirements

System

The application will run on all Android devices running 4.1 (JellyBean) or Later. It will be around 11mb in size. The application will respond to the size of the screen and/or window the application is running in.

Response Time

The application should take less than 4 seconds when running on an Android phone and less than 8 second when on an emulator or tablet. The application will run fine until the user begins to multi-task between 3 or more processes.

Workload

The application must support approximately 10,000 users at the time of launch based on the population of the RU student body.

Scalability

The application will be able to scale to the size of the RU student body as it increases.

5.2 Safety Requirements

Since we have used firebase to store the reminders and encrypted images the user information is retained if the app is deleted from the mobile phone. Firebase provides a sense of safety to the users data. It acts like a backup for the user's data.

5.3 Security Requirements

Since the app tracks payments security is an important aspect. The user's account is authenticated with username and password such that unauthenticated users cannot view the reminder information.

5.4 Software Quality Attributes

- Reliability:
The application should have capability to maintain minimum level of performance.
- Availability:
The application should run 24x7 if internet connection is available.
- Portability:
User should be able to install applications easily using apk.

Appendix A: Glossary

Definition:

Android

Mobile Operating System developed by Google Inc.

Object-Oriented

A methodology that enables a system to be modeled as a set of objects that can be controlled and manipulated in a modular manner.

App

A program or piece of software designed and written to fulfill a particular purpose of the user.

Category

A category is a descriptor containing the multidimensional vocabulary items having a similar meaning, relation and/or purpose.

Reminder

A small collection of data about an important event that includes a task and time.

Appendix B: Analysis Models

Use case diagram:

