**Software Requirement Specification**

**(SRS)**

Personal Financial Management

System – Viit

Group Members

Ayushi Goel

Antra Tripathi

Abhinav Ralhan

# *Introduction*

## 1.1 Purpose

The introduction of the Software Requirements Specification **(SRS)** provides an overview of the entire SRS with purpose, scope, definitions, acronyms, abbreviations, references and overview of the SRS. This document lists the features of our application **(VITT)** so as to serve as a guide to the developers on one hand and a software validation document for the prospective client on the other

The aim of this document is to gather and analyze and give an in-depth insight of the complete **VITT application** by defining the problem statement in detail. This application basically provides a user friendly platform to manage your day-to-day expenses.

## 1.2 Scope

Primarily, the scope pertains to the usability and utility of application (i.e. Vitt) based on user interaction.

In Scope:

* Managing your personal finance on daily, weekly or monthly bases.
* Displaying graphical analysis of entered expenses.
* Customizing application according to user need.
* Option of importing data to excel if required by the user.
* Generating alerts on user demand.

Out of Scope:

* This application has no connection with user’s bank account.
* Online transaction such as online shopping cannot be linked automatically.

**1.3 Definitions, Acronyms, and Abbreviations:**

Acronyms and Abbreviations:

* **VITT :** Name of the our android application derived from Hindi word for finance.
* **SRS:** Software Requirements Specification

**1.4Overview**

The rest of this SRS is organized as follows:

Section 1: An Introduction to SRS.

Section 2: Gives an overall description of the application and describing its proficiency level.

Section 3: System requirement / Functional requirement by various use cases.

Section 4: Future Expectation

# *Overall Description*

**2.1 Product Perspective:**

VITT is aimed toward those users who would like to keep a track of their day to day personal finance. It usually happens that we forget about small expenses, so why not note it down then and there itself. This application is a user friendly platform which keeps a track of users daily expenses.

It will be easy operable, with no dependency on any other software. It is intended to be standalone product.

**2.2 Product Functions:**

VITT should support the following cases:

|  |  |  |
| --- | --- | --- |
| **Class of Use Cases** | **Use Cases** | **Description of Use Cases** |
| Use case related to system Installation | Installation | Creates and initializes working files |
| Use case related to Signup | Signup | Signup to Vitt |
| Through Gmail, Facebook |
| Use case related to system Authorization | Login | Login into Vitt |
| Password | Change Vitt Password |
| Use case related to Portfolio | Create Portfolio | It creates a new Portfolio |
| Rename Portfolio | Renames an existing Portfolio |
| Delete Portfolio | Deletes an existing Portfolio |
| Use case related to Transactions | Add Transactions | Adds expenditures |
| Edit Transactions | Edits existing expenditures |
| Deleted Transactions | Deletes expenditures |
| Use case related to Computations | Compute Net Expenditure | Compute Net Expenditure of all products/services |
| Compute Debt | System computes existing debt using Net expenditure and Income |
| Use case related to Display of Expenditure | Graphical Analysis | Shows the trend of expenditures |
| Use case related to Notifications | Notifications | Choosing the frequency of app usage |
| Use case related to Excel | Excel | Imports/Exports of files to/from Excel |
| Use case related to Shopping List | Shopping List | Helps analysing items which can be purchased according to remaining balance |
| Use case related to Addition of Categories | Categories | Allows user to add categories as per his/her needs |
| Use case related to Alerts | Set Alerts | Give date and details |
| Show Alerts | Shows pending alerts regularly |
| Delete Alerts | Delete an already set alert |

**2.3 User Characteristics:**

The user should be familiar with working of a smart phone.

**2.4 General Constrain:**

It’s a single user based application.

***3. Specific Requirements:***

**3.1 Functional Requirements:**

We describe the functional requirements by giving various use cases.

Use case related to installation:

**Use Case 1:** Installation

Primary Actor: User

Pre Condition: Internet connection available.

1. Main Scenario:
2. Install the complete setup from the give valid source
3. Run the application and customize it accordingly.

2. Alternate Scenario:

(a) Network failure.

(b) Application fails to work.

Use case related to signup:

**Use Case 2:** Signup

Primary Actor: User

Pre Condition: Internet connection available.

Main Scenario:

1. Start the application. User prompted for signup.

2. Authenticate through personal E-mail/Gmail/Facebook.

3. Alternate Scenario:

(a) Network failure.

Use case related to system authorization:

**Use Case 3:** Login

Primary Actor: User

Pre Condition: Internet connection available.

Main Scenario:

1. Start the application. User prompted for login.

2. User gives the login and password.

3. System does authentication.

4. Main screen is displayed.

5. Alternate Scenario:

(a) Network failure.

(b) Prompt the user that he typed the wrong password.

(c) Allow him to re-enter the password. Give him 3 chances.

**Use Case 4:** Change Password

Primary Actor: User

Pre Condition: User logged in.

Main Scenario:

1. User initiates the password change command.

2. User is prompted for old password, new password and confirm new password.

3. User gives the old password, new password and confirm new password.

4. System does authentication.

5. New password is registered with the system.

6.Alternate Scenario:

(a). Authorization fails

(b) Prompt the user that he typed the wrong password

(c) Allow him to re-enter the password. Give him 3 chances.

(d) New password and confirm new password do not match.

(e) Allow him to re-enter the attributes. Give 3 chances

Use cases related to portfolios:

**Use Case 5:** Create Portfolio

Primary Actor: User

Pre Condition: User logged in.

Main Scenario:

1. User chooses the portfolio version, i.e, for a Student/Professional.

2. System asks the user to enter basic information.

3. User customizes app according to given themes.

4. A user defined portfolio is created.

5. Alternate Scenario:

(a) Default Portfolio exists.

(b) Missing mandatory fields in the portfolio created.

**Use Case 6:** Rename portfolio.

Primary Actor: User

Pre-Condition: User logged in.

Main Scenario:

1. User initiates the “rename portfolio” functionality.

2. System asks for the portfolio to be renamed and the new

name.

3. User enters the new name.

4. Portfolio is renamed.

Alternate Scenario:

(a) The portfolio whose name is supposed to change does not exist.

(b) Renaming fails, the error message is displayed.

(c) Portfolio with the same new name exists.

(d) Renaming fails, the error message is displayed.

**Use Case 7:** Delete portfolio.

Primary Actor: User

Pre-Condition: User logged in.

Main Scenario:

1. User initiates the ”delete portfolio” functionality.

2. System asks for the name of the portfolio.

3. The portfolio is deleted.

4. Alternate Scenario:

(a). Portfolio does not exist.

(b) Deletion fails, error message is displayed.

Use cases related to transactions:

**Use Case 8:** Adding a transaction.

Primary Actor: User.

Pre-Condition: User logged in.

Main Scenario:

1. User initiates the “add transaction” functionality.

2. User chooses the category and item of expenditure.

3. New transaction is created.

**Use Case 9:** Editing a transaction.

Primary Actor: User.

Pre-Condition: User logged in.

Main Scenario:

1. User initiates the “edit transaction” functionality.

2. User chooses the category and item of expenditure.

3. The transaction is edited.

**Use Case 10:** Deleting a transaction.

Primary Actor: User.

Pre-Condition: User logged in.

Main Scenario:

1. User initiates the “delete transaction” functionality.

2. User chooses the category and item of expenditure.

3. The transaction is deleted.

Use case related to computation.

**Use Case 11:** Compute Net Expenditure.

Primary Actor: User.

Pre-Condition: User logged in.

Main Scenario:

1. System computes Net Expenditure of products/services.

**Use Case 12:** Compute Debt.

Primary Actor: User.

Pre-Condition: User logged in.

Main Scenario:

1. System computes existing debt using Net expenditure and Income.

Use case related to Display of Expenditure

**Use Case 13:** Graphical analysis.

Primary Actor: User.

Pre-Condition: User logged in.

Main Scenario:

1. System analyses expenditure on various items giving their graphs/charts.

2. User can analyse it with respect to their income.

Use case related to Notifications:

**Use Case 14:** Notifications.

Primary Actor: User.

Pre-Condition: User logged in.

Main Scenario:

1. User chooses the frequency of App usage.

Use case related to Excel:

**Use Case 15:** Excel.

Primary Actor: User.

Pre-Condition: User logged in.

Main Scenario:

1. User initiates the Import/Export functionality which does the same for files using Excel

Use case related to Shopping List:

**Use Case 16:** Shopping List.

Primary Actor: User.

Pre-Condition: User logged in.

Main Scenario:

1. System helps analyse items which can be purchased according to remaining balance.

Use case related to Addition of Categories:

**Use Case 17:** Categories.

Primary Actor: User.

Pre-Condition: User logged in.

Main Scenario:

1. System allows users to add categories like as per his/her needs.

Use case related to Alerts:

**Use Case 18:** Set Alerts

Primary Actor: User.

Pre-Condition: User logged in.

Main Scenario:

1. System allows users to provide date and details to set an alert.

**Use Case 19:** Show Alerts

Primary Actor: User.

Pre-Condition: User logged in.

Main Scenario:

1. System shows the user pending alerts regularly.

**Use Case 20:** Delete Alerts

Primary Actor: User.

Pre-Condition: User logged in.

Main Scenario:

1. System allows users to delete an already set alert.

**3.2 Security Requirements:**

Information given by users is kept in the system database. This information can not be reached by other users or external threats. The users in the application are all equal so there is not the need of any identity management level.

**3.3 External Interface Requirements:**

1. Hardware Interfaces

Android based smart phone with internet connection ability is the hardware interface of this system.

1. Software Interfaces

VITT runs on all Android based smart phones for API level 14 or above.

Internet connection of smart phone is also required.

***4. Future Extensions:***

VITT is intended to be single user application. A possible future extension would be to make it multi user.