**Software Design Document**

**Personal Finance Management Application: VITT**

**Ayushi Goel**

**Antra Tripathi**

**Abhinav Ralhan**

**Introduction**

**1.1 Purpose of this document**

The purpose of this document is to outline the software design of the VITT Application as described in the Software Requirement Specification(SRS). The VITT application basically provides a user-friendly platform to manage your day-to-day expenses.

**1.2 Scope of the development project**

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. **Definitions, acronyms, and abbreviations**
* **VITT:** Name of the our android application derived from Hindi word for finance.
* **SRS:** Software Requirements Specification
* **SDD:** Software Design Document
* **GUI:** Graphical User Interphase
* **Transaction:** Amount of money spent / earned

**1.4 References**

N/A

**1.5 Overview of document**

Following is the format of the SDD:

* Introduction
  + Purpose and Scope of the project
* Conceptual Architecture of the Design
  + Overview of modules
  + Structure and relationship
  + User interface issues [Screenshots]
* Logical Architecture
  + Class Diagram
  + Sequence Diagram
  + State Activity Diagram
* Execution Architecture
* Design decision and Tradeoff
* Pseudocode for components

**2. Conceptual Architecture/Architecture Diagram**

**2.1 Overview of modules / components**

This app has been designed keep in mind the constantly evolving needs of the customer. Hence, VITT by its way of helping its users also keeps changing with them, their expenses and incomes. VITT tries to analyze every aspect of an individual’s expenses and income by giving features to create a personalized profile; add, delete or edit transactions; choose the category where expenses have been made or income generated; add a category if not present; define a threshold for expenses; set alerts and notifications and quickly view the graphical analysis of daily, weekly or monthly summary.

**2.2 Structure and relationships**

**Login/Signup**

* Login Authentication (Must be a registered user)
* Sign Up and create a new account if a new user (doctor or patient)
* Using features like add/edit transaction, monthly/daily/weekly graphical analysis etc.
* Implemented using Android IDE.

**Adding and Editing Transaction**

* Selecting categories
* Adding amount
* Selecting date of transaction
* Add note if any.
* Implemented using Android IDE.

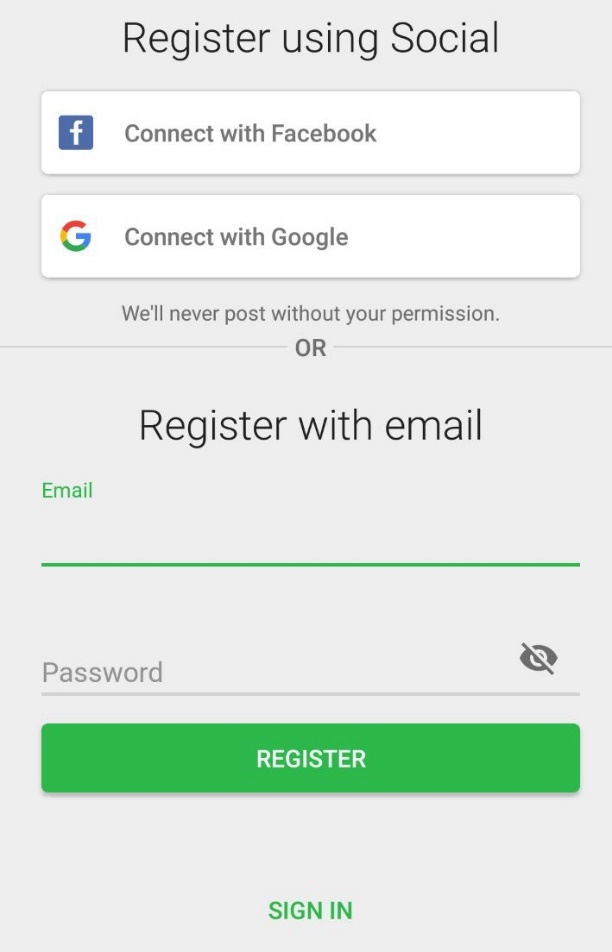
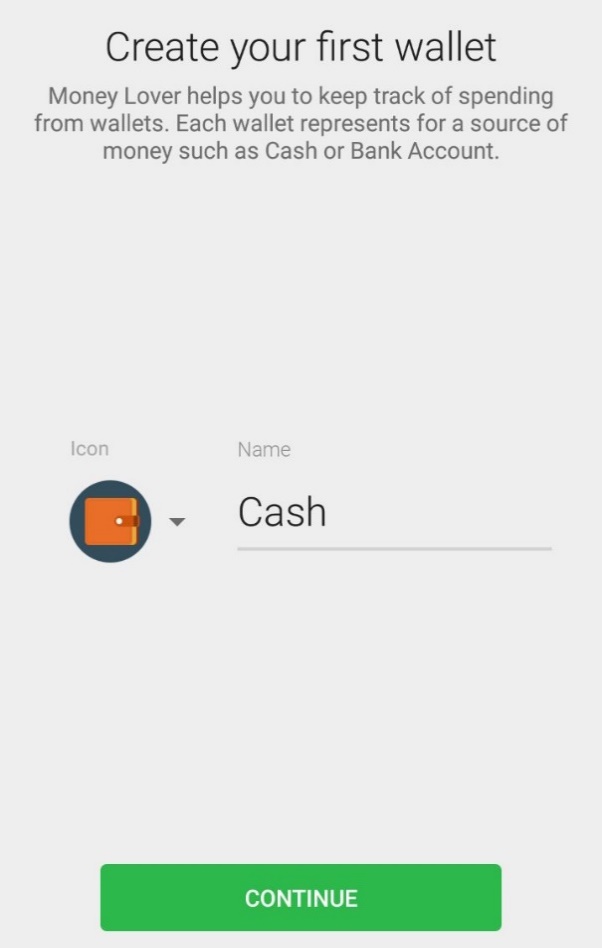
**Viewing your Expenditure**

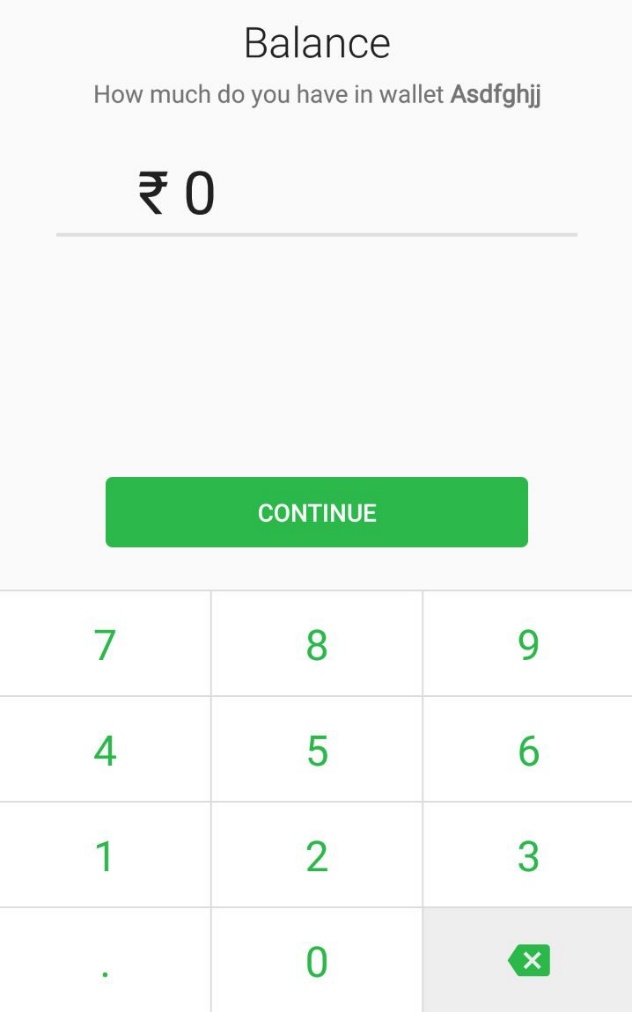
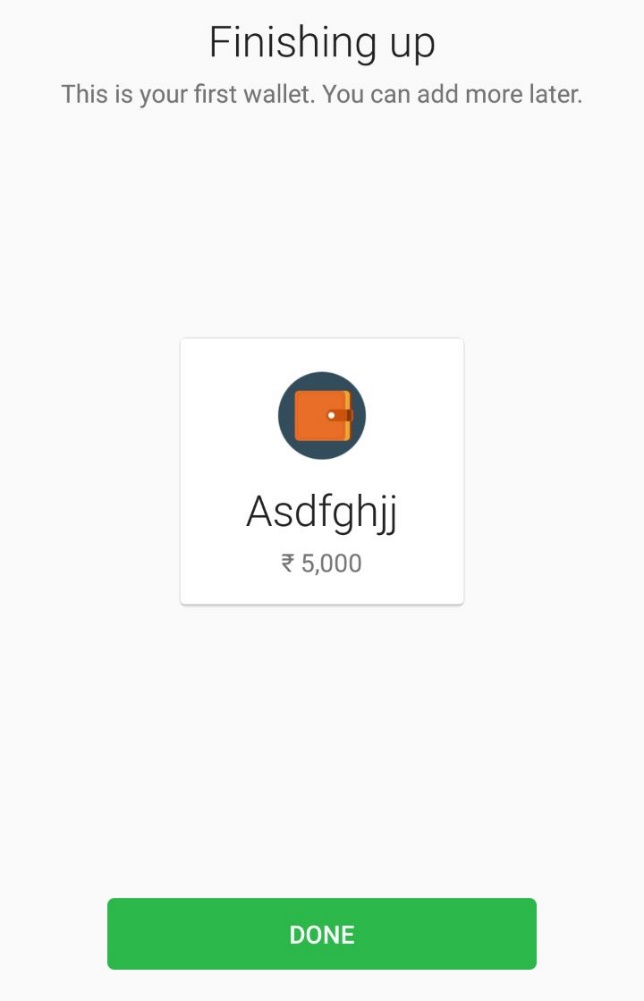
* User can see the graphical analysis of the input expenditure.
* It can be on monthly/ weekly/ daily basis.

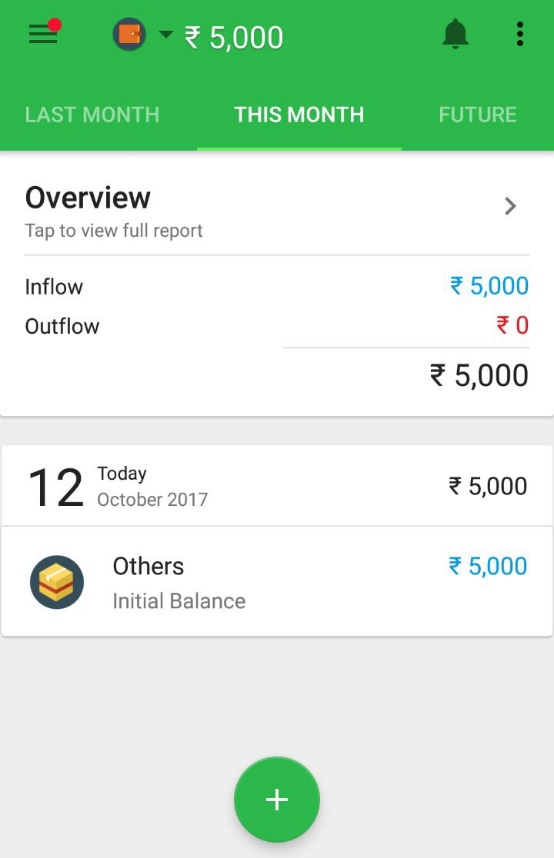
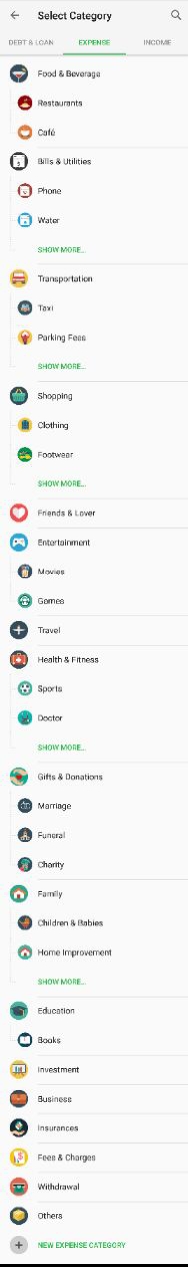
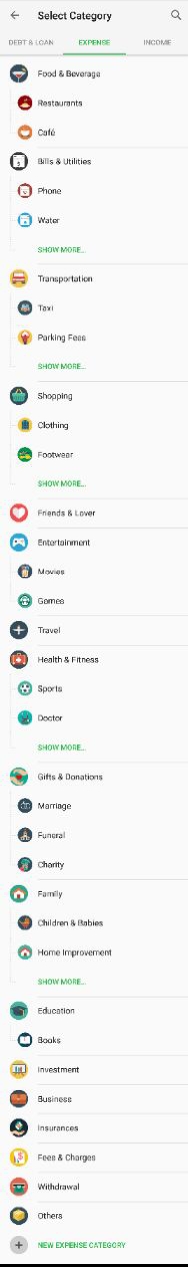
**2.3 User interface issues**

**Screen shot**

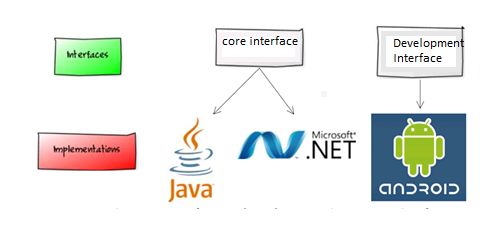
Tentative Design of the App

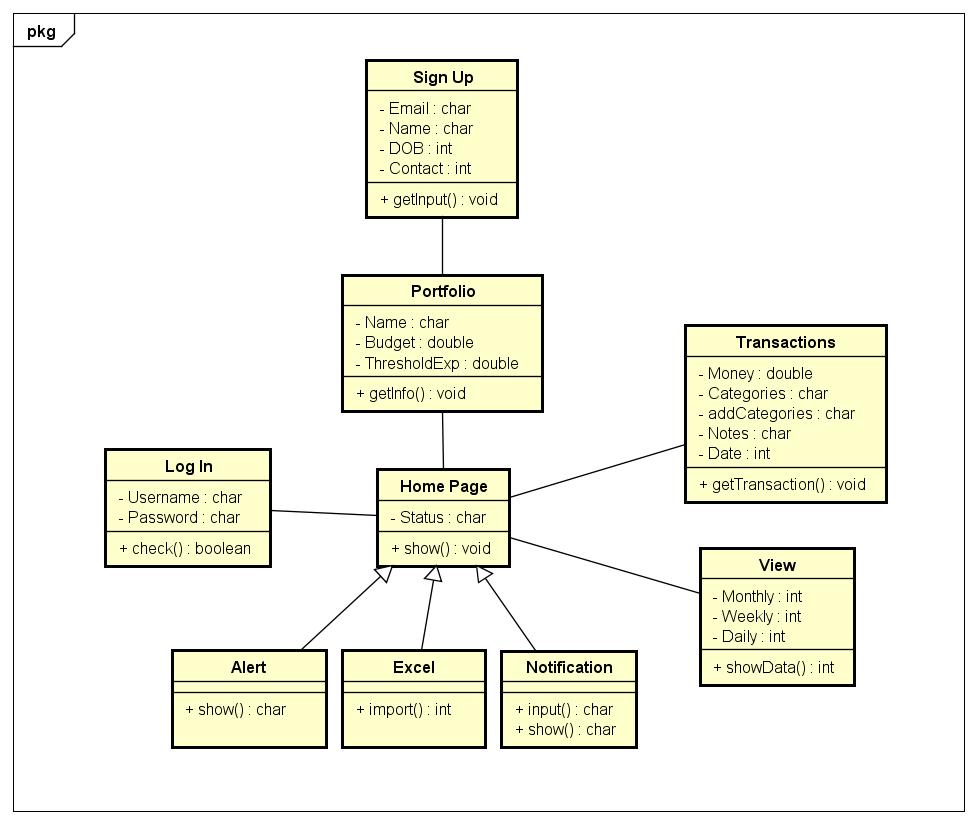
 



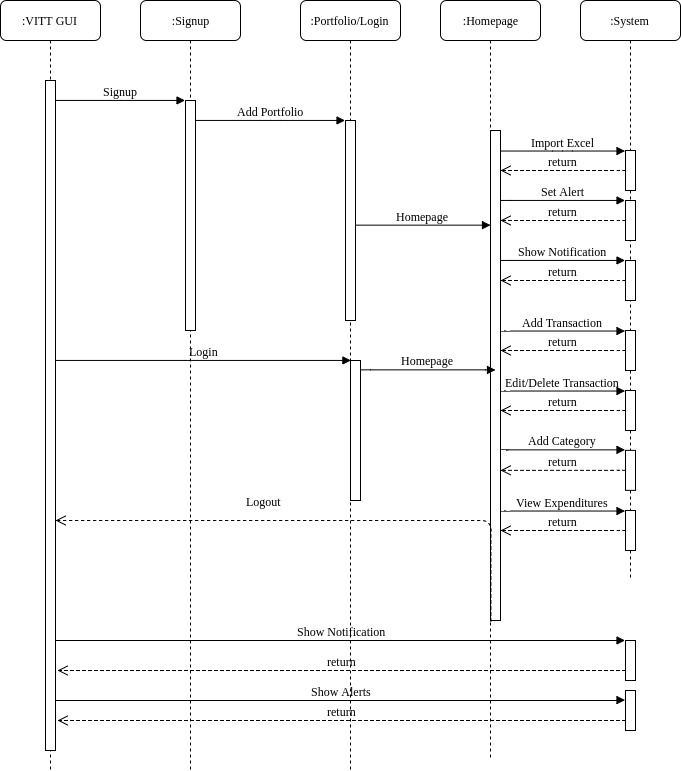
1. **Logical Architecture (Class Diagram, Sequence Diagram, State Diagram)**



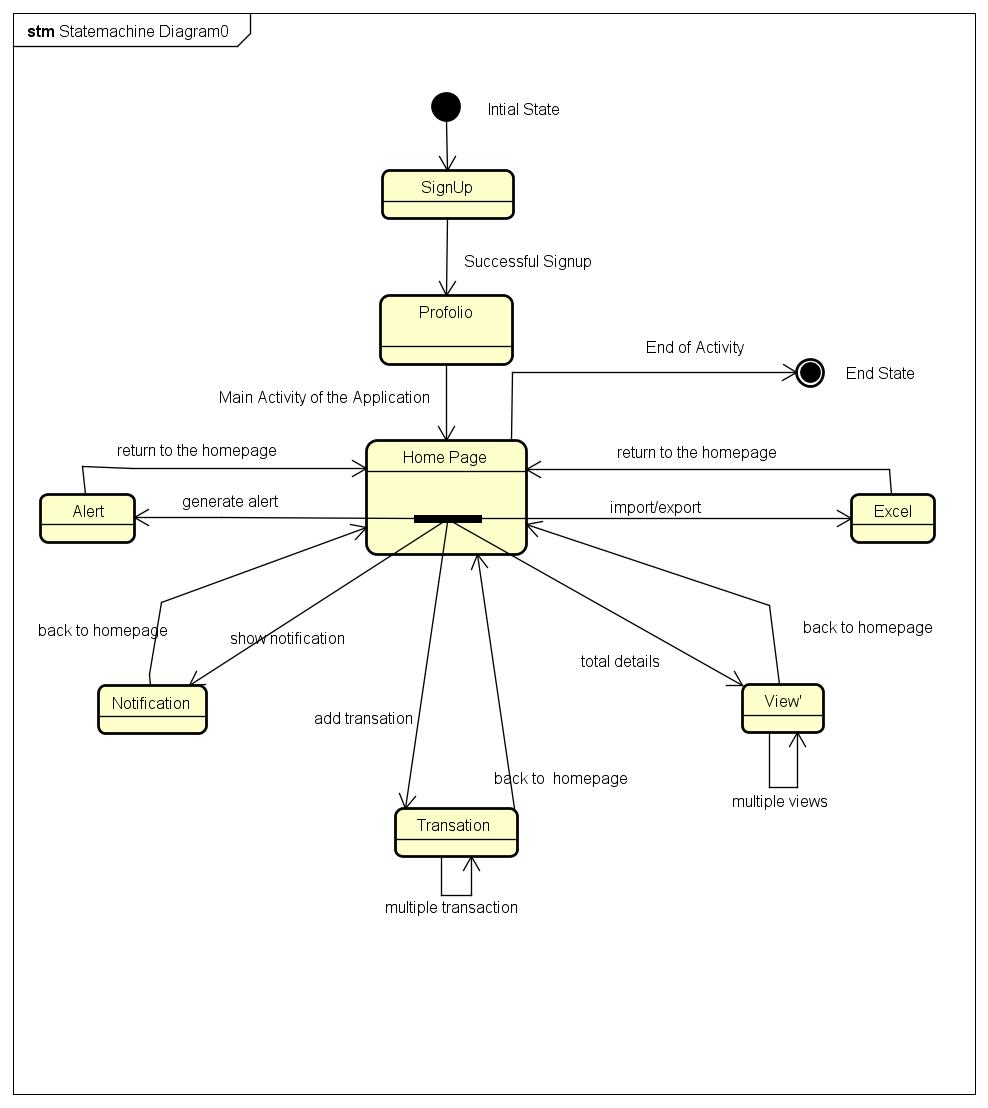
**Class Diagram**



**Sequence Diagram**

****

**State Activity Diagram**

****

**4.0 Execution Architecture**

This app will be made completely using android studio as a platform for developing and will be coded in java.

**5.0 Design decisions and tradeoffs**

The biggest decision taken in terms of the app was the platform where it should be built, whether it should be android, IOS or Windows. Finally, the team decided on android, as it is the most widely used platform and hence would benefit a lot of people. The second thing was deciding whether to integrate the app with the customer’s bank account or not. We finally decided against it due to privacy issues pointed to by many people according to the survey that we conducted.

**6.0 Pseudocode for components**

**Signup Class**

|  |  |
| --- | --- |
| void getEmail() | |
| Output | None |
| Description | Gets the e-mail id of the user. |
|
| void getName() | |
| Output | None |
| Description | Gets the name of the user. |
|
| void getDOB() | |
| Output | None |
| Description | Gets the DOB of the user. |
|
| void getContact() | |
| Output | None |
| Description | Gets the contact details of the user. |
|

**Login**

|  |  |
| --- | --- |
| void getUsername() | |
| Output | None |
| Description | Gets the username of the user. |
|

|  |  |
| --- | --- |
| void getPassword() | |
| Output | None |
| Description | Gets the password of the user. |
|

**Portfolio**

|  |  |
| --- | --- |
| void getName() | |
| Output | None |
| Description | Gets the name of the Portfolio. |
|

|  |  |
| --- | --- |
| void getPicture() | |
| Output | None |
| Description | Gets the picture of the Portfolio. |
|

|  |  |
| --- | --- |
| void getBudget() | |
| Output | None |
| Description | Gets the budget details of the user. |
|

|  |  |
| --- | --- |
| void getThreshold() | |
| Output | None |
| Description | Gets the threshold limit of the user. |
|

**Homepage Class**

|  |  |
| --- | --- |
| void showOverview() | |
| Type | Control Procedure |
| Output | Displays monthly expenses |
| Dependencies | Requires modules from Portfolio, Login, View, Transaction Classes |
| Description | Use showOverview() method to add/edit/delete porfolio, view monthly/weekly/yearly expenses, add/edit/delete transactions, see notifications, analysis, and set alerts. |
|

|  |  |
| --- | --- |
| void showMoney() | |
| Output | A bar which displays the usage of money |
| Description | Use showMoney() method to see used and available money |
|

**Views Class**

|  |  |
| --- | --- |
| void showMonthly() | |
| Output | Displays monthly expenses |
| Description | Use showMonthly() module to display the expenses and income(s) of the month. |
|

|  |  |
| --- | --- |
| void showWeekly() | |
| Output | Displays weekly expenses |
| Description | Use showWeekly() module to display the expenses and income of the week. |
|

|  |  |
| --- | --- |
| void showDaily() | |
| Output | Displays yearly expenses |
| Description | Use showYearly() module to display the expenses and income(s) during the entire year. |
|

**Transaction Class**

|  |  |
| --- | --- |
| void getMoney() | |
| Input | Takes input from user to input the amount |
| Description | The user has to enter the amount required for the particular transaction. |
|

|  |  |
| --- | --- |
| void getCategory() | |
| Input | Takes input from user to input the category |
| Description | The user has to classify the item to be purchase into a given category. |
|

|  |  |
| --- | --- |
| void addCategory() | |
| Input | Takes input from user to add a category |
| Description | The user has to enter a new category to classify his/her product. |
|

|  |  |
| --- | --- |
| void addNote() | |
| Input | Text input for the transaction |
| Description | The user enters a note for the product purchased. |
|

|  |  |
| --- | --- |
| void setDate() | |
| Input | Date Input |
| Description | The user inputs a date for the transaction made. The default date chosen is the day of transaction itself. |
|

**Alert Class**

|  |  |
| --- | --- |
| void setAlert() | |
| Output | User can set an alert. |
| Description | User can set alerts to see if the threshold budget has been exceeded. |
|

**Notifications Class**

|  |  |
| --- | --- |
| void setNotifications() | |
| Input | User can set Notifications |
| Description | User can choose products and regularity of notifications. |
|

|  |  |
| --- | --- |
| void showNotifications() | |
| Input | User views Notifications |
| Description | User receives notifications for purchases made regularly. |
|

**Excel Class**

|  |  |
| --- | --- |
| void importInfo() | |
| Output | User can upload information |
| Description | User can upload information of purchases in the past through Excel. |
|