UNIVERSITY PARTNER

UNIVERSITY OF
WOLVERHAMPTON

HERALD
COLLEGE

Project and Professionalism (6CS007)

A2: Project Report FINANCE (Artefact)

Student Id   : 2050089
Student Name   : Sajat Prasad Shrestha
Group   : L6CG5
Supervisor   : Bhanu Aryal
Reader   : Rupak Koirala
Cohort   : 5
Submitted on   : 13-11-2021

Contents

1. Functional Decomposition Diagram

The finance system when broke down into smaller systems comprises of five systems. First one is the token auth system in which basis a user will be given their unique user ID. Using Facebook and google for user credentials will require firebase dependencies for this project. Secondly, money management will be about keeping track of all the transaction added by the user and having it tracked according to date for other systems.

Expense Prediction System will have a linear regression algorithm built from the past data of the user which will predict expenses for the next month according to the target saving set by the user. Auto categorization system will be an unsupervised machine learning model which will predict which category might the transaction fall into and determine whether the category is income or an expense. Lastly a detailed analysis will be provided so the user can take an informed decision.
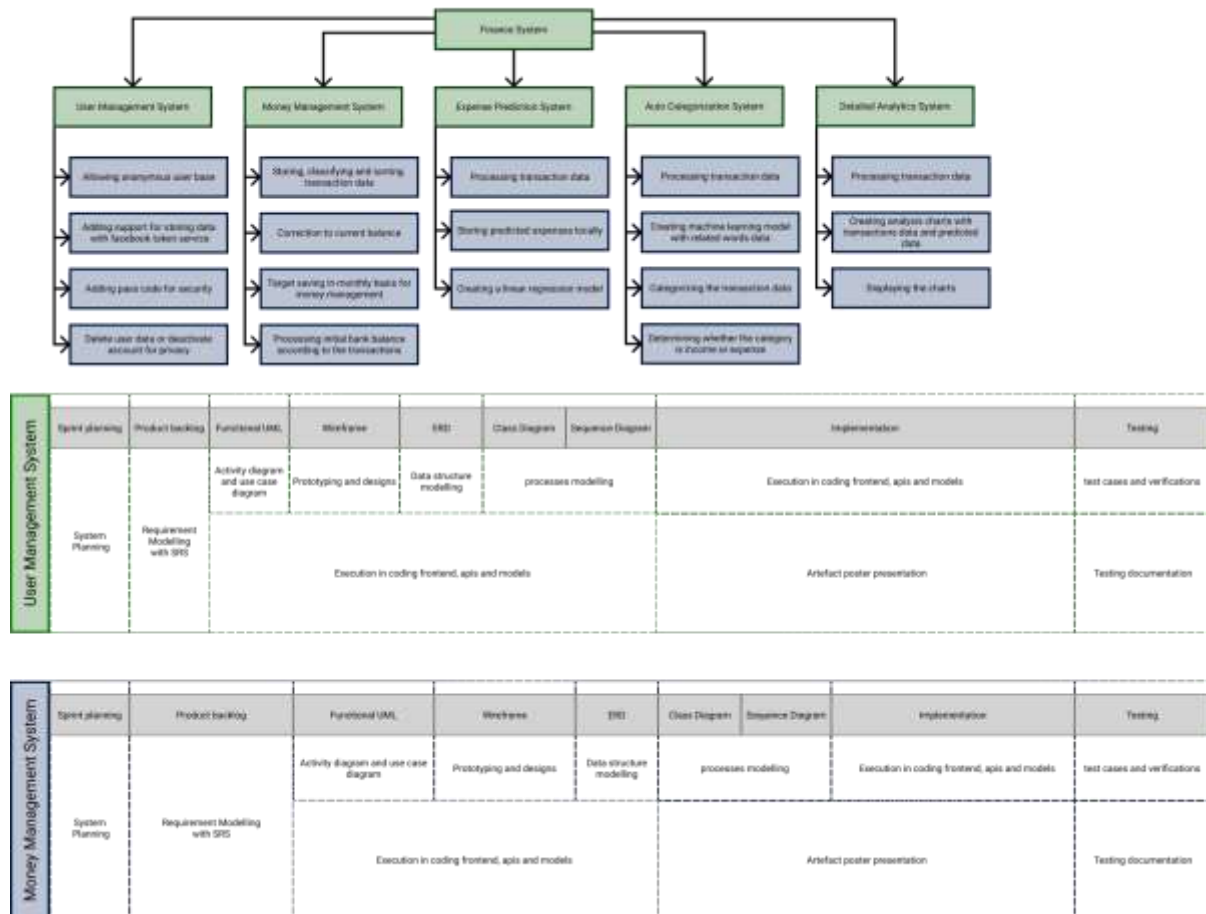

Figure 1: Functional Decomposition Diagram

1.1 User Management System (US)

Req. Code	Requirement Description	Use Case	Moscow Prioritization
US-F-1.0	Allowing anonymous user base to use the application without logging in	Anonymous user base	Must Have
US-NF-1.1	App to auto create an anonymous account to store anonymous user data right after the launch of the application	Auto Sign Up	Should Have
US-F-2.0	Support for storing data with O-Auth from Facebook or Google	Add Token Service	Must Have
US-NF-2.1	User can choose avatar that best represents themselves to recognize their min age estimation, max age estimation and gender	Add Avatar Selection	Should Have
US-NF-2.2	The User can log out of their account if they want to sign in from another account	Add log out option	Must Have
US-NF-2.3	The session id is stored locally so they don't have to sign in again unless they log out	Add session ID	Should Have
US-UR-2.1	The User can view their user data in the system from their profile page	Add profile Screen	Should Have
US-F-3.0	4-digit passcode for app security when opening the application	Passcode	Must Have
US-NF-3.1	User can change their passcode if they want to	Change Pin Code	Must Have
US-NF-3.2	User needs to retype their pin code in order to confirm their pin code while changing the password	Confirm pin code	Should Have

| US-NF-3.3	User needs to type their current pin code to confirm that they are the valid user	Ask for their current password	Should Have
US-UR-3.1	The typed pin code is obscure for better security while entering the pin code	Add obscure text	Must Have
US-F-4.0	The User can delete their personal data from the server for better privacy	Add Clear data option	Must Have
US-UR-4.1	A dialog box is provided in order to confirm if they want to clear their data	Add confirm dialog box	Should Have
US-F-5.0	The User can deactivate their account if they don't want their data to be used while the account is deactivated	Add deactivate account function	Should Have
US-UR-5.1	A dialog box is provided in order to confirm if they want to deactivate their account	Add confirm dialog box	Should Have

1.1.1 Sequence Diagram
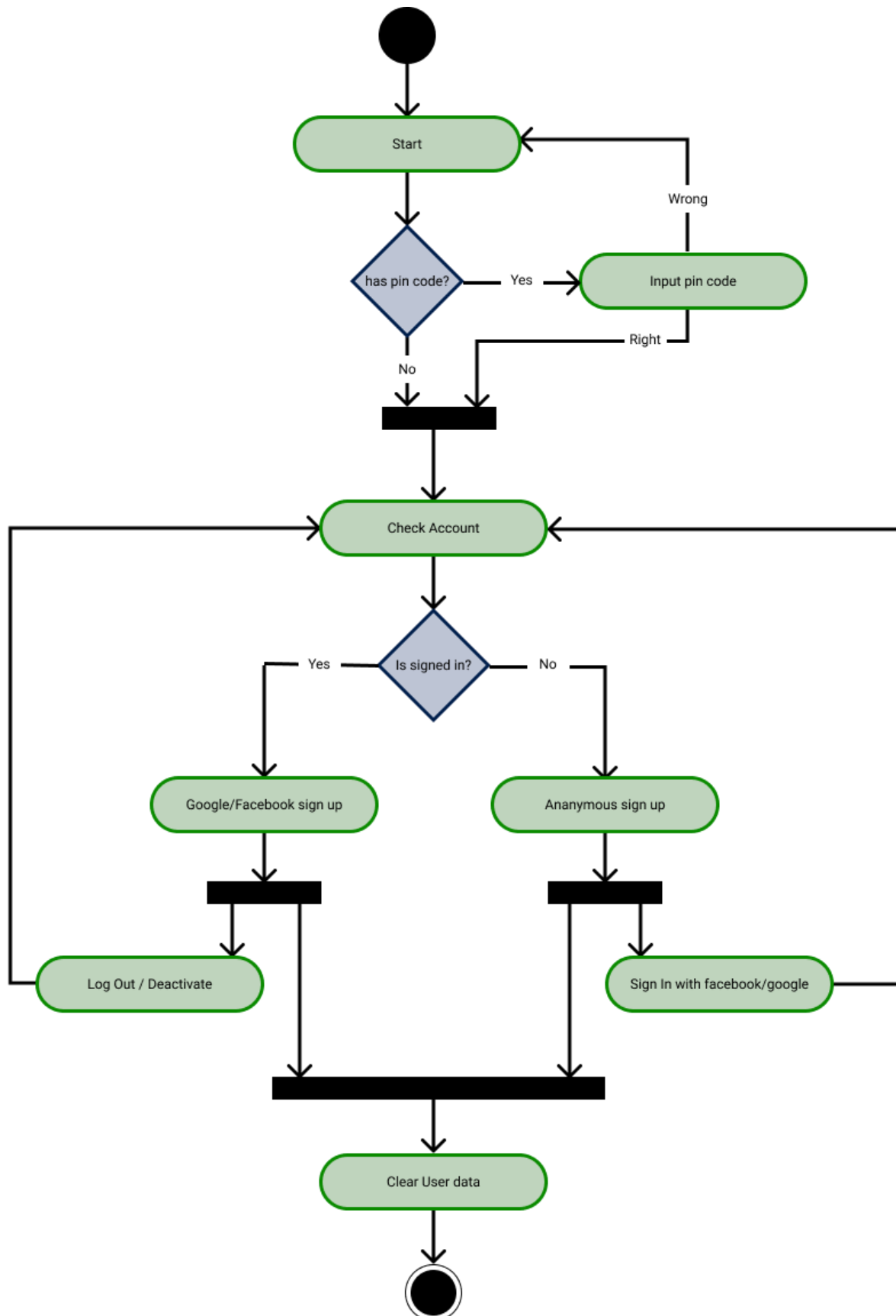
Figure 2: Sequence Diagram for user management system

1.1.2 Functional Testing

Test Cases

TC ID	REQ ID	TEST CASE TITLE	DESCRIPTION	PRECONDITION	STEPS	TEST DATA	EXPECTED RESULT	ACTUAL RESULT	PASS/FAIL
TC-01	US-F-1.0	Anonymous log in	Users should not be able to access features of app without signing in with google or Facebook synced with server	Should open the app while not logged in	If logged in log out		Logged in to anonymous account	Logged in to anonymous account	PASS
TC-02	US-F-2.0	Facebook log in	Users should be able to login using Facebook token	Sign in screen	Click on sign in with Facebook Enter Facebook credentials		Signed up with Facebook	Signed up with Facebook	PASS
TC-03	US-F-2.0	Google log in	Users should be able to login using Google token	Sign in screen	Click on sign in with Google Enter Google credentials		Signed up with Google	Signed up with Google	PASS
TC-04	US-NF-2.1	Log in screen	Choosing an avatar should define the age estimation and gender of the user	Sign in screen	Sign in with a token and select an avatar		Relative age estimation and gender	Relative age estimation and gender	PASS
TC-05	US-NF-2.1	Log in screen	User should not be able to enter without selecting an avatar	Sign in screen	Try to login without choosing avatar		Disabled button	Disabled button	PASS
TC-06	US-NF-2.2	Log out	The user should be able to log out	Settings screen	Clicking on log out button		Logged out	Logged out	PASS
TC-07	US-NF-2.3	Session id	The user should not have to login again if he has logged in before	Should open the app while logged in	Sign in and re open the app		Didn't ask for credential again	Didn't ask for credential again	PASS
TC-08	US-NF-2.3	Session id	The session id should be cleared when the user logs out	Should open the app while logged out	Log out and re open the app		Asked for credentials	Asked for credentials	PASS
TC-09	US-UR-2.1	User data	The user should be able to view their user data	Settings screen			Show user data	Show user data	PASS

1

T C-10	US-F-3.0	Passcode	The user should be able to set passcode when none is set	Settings screen	Click on add passcode Enter new passcode Confirm new passcode	1234 1234	Set new passcode	Set new passcode	PASS
T C-11	US-NF-3.3	Passcode	The user should be asked for their previous password if a passcode is already set	Settings screen with passcode set	Click on add passcode Enter previous passcode Enter new passcode Confirm new passcode	1234 4321 4321	Set new passcode	Set new passcode	PASS
T C-12	US-NF-3.1	Passcode	The user should not be allowed to set new passcode if the previous passcode wasn't correct	Settings screen with passcode set	Click on add passcode Enter wrong previous passcode	1233	Toast message which says wrong password	Toast message which says wrong password	PASS
T C-13	US-NF-3.2	Passcode	The user should not be allowed to set new passcode if the confirm didn't match the set passcode	Settings screen	Enter a new passcode Confirm with wrong passcode	1234 1233	Toast message which says passcode didn't match	Toast message which says passcode didn't match	PASS
T C-14	US-UR-3.1	Passcode	The passcode should be obscure while being typed	change passcode screen	Enter passcode	123	Obscure text	Obscure text	PASS
T C-15	US-F-4.0	Clear data	The user can clear their personal data from the server	Settings screen	Click on clear data Click on confirm		Data cleared	Data cleared	PASS
T C-16	US-UR-4.1	Clear data	A dialog box to appear to confirm if the user wants to clear data	Settings screen	Click on clear data Click on confirm		Dialog box appears	Dialog box appears	PASS
T C-17	US-F-5.0	Deactivate account	The user can deactivate their personal data from the server	Settings screen	Click on deactivate account Click on confirm		Account deactivated	Account deactivated	PASS
T C-18	US-UR-5.1	Deactivate account	A dialog box to appear to confirm if the user wants to deactivate their account	Settings screen	Click on deactivate account Click on confirm		Dialog box appears	Dialog box appears	PASS

1.2 Money Management System (MS)

Req. Code	Requirement Description	Use Case	MoSCoW Prioritization
MS-F-1.0	The user can store, update or delete their transaction data in the system	Add transaction	Must Have
MS-NF-1.1	The transaction amount is subtracted from the initial bank balance to gain new initial bank balance	Calculate new bank balance	Must Have
MS-NF-1.2	Toast message if creation editing or deletion was not successful	Catch status code exceptions	Must Have
MS-UR-1.1	The transactions are denoted by red if it is an expense and by green if it is an income	Add notation	Should Have
MS-F-2.0	User can add correction to the current balance if the user misses to input transactions. Similarly, the user can update and delete corrections.	Add correction functionality	Should Have
MS-NF-2.1	The correction is subtracted from the current bank balance on the application to obtain the actual bank balance	Calculate the actual bank balance	Should Have
MS-F-3.0	User can set a target monthly saving with which the system can analyse its goals better	Set target monthly saving	Must Have
MS-NF-3.1	The User is notified if the threshold is breached and the target saving for the month is crossed	Set threshold and notify	Must Have
MS-UR-3.1	The notification is easily dismissible	Use dismissible dialog box	Must Have

|			
MS-F-4.0	Categories can be added, edited and deleted	Add, edit and delete category	Must Have
MS-NF-4.1	The transactions can be categorized into groups	Add categorization	Must Have
MS-NF-4.2	When deleting a category, the user is asked to categorize the transactions that falls under the category	Add recategorize transactions from deleted category	Must Have
MS-NF-4.3	When adding a category, the user needs to add minimum of three related words	Add minimum related words validation	Should Have
MS-NF-4.4	Toast message if creation editing or deletion was not successful	Catch status code exceptions	Must Have
MS-UR-4.1	The add category navigation can be found in the add transaction screen	Add category function in add transaction	Should Have
MS-F-5.0	The transactions can be sorted by date or by amount for better analysis	Add sort transactions	Must Have
MS-F-6.0	The transactions can be filtered by day week, month or by year.	Add filter transactions	Must Have

1.2.1 Sequence Diagram
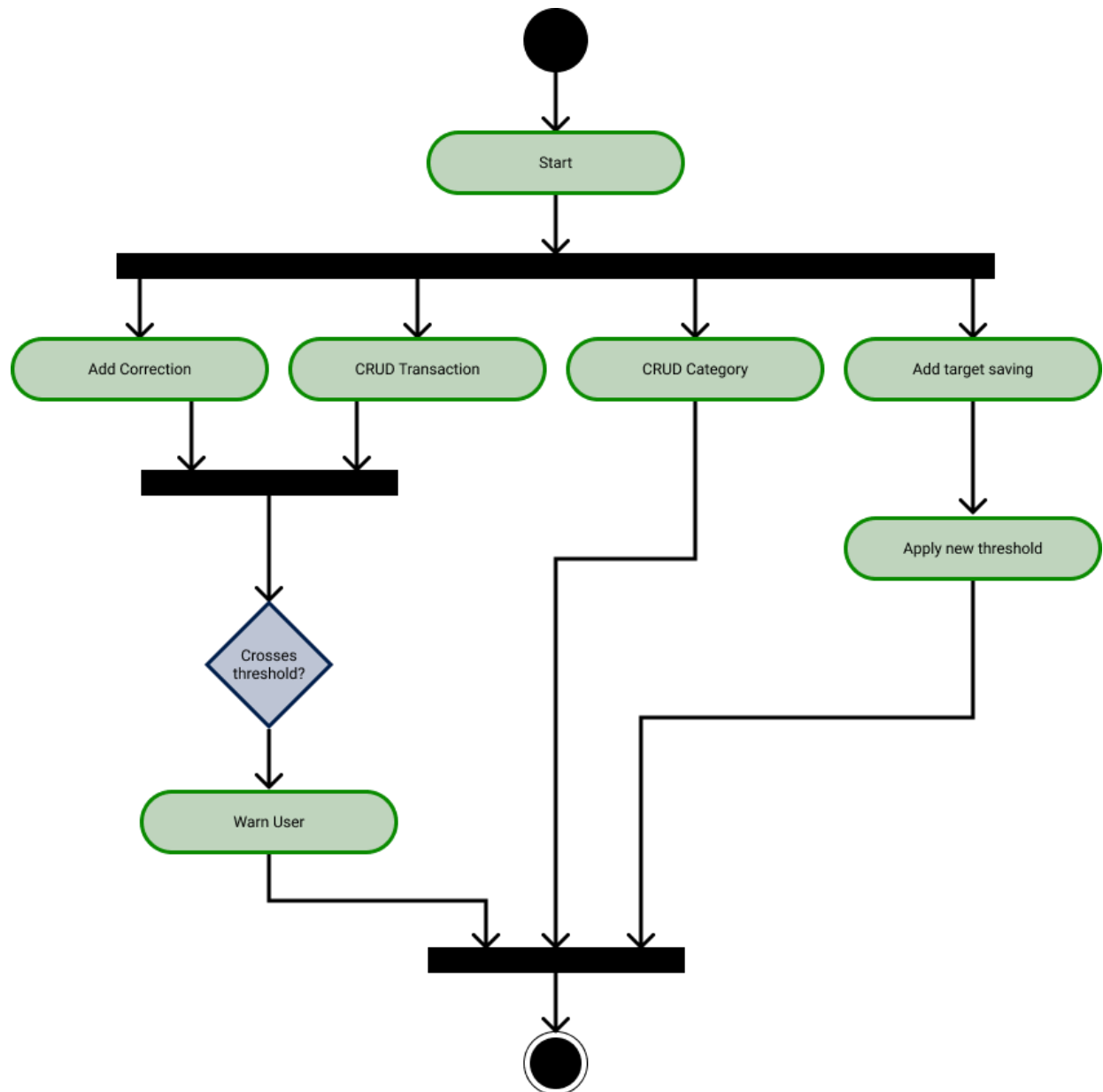
Figure 3: Sequence diagram for money management system

1.2.2 Functional Testing (add exception handling while creation update and delete)

Test Cases

TC ID	REQ ID	TESTCASE TITLE	DESCRIPTION	PRECONDITION	STEPS	TEST DATA	EXPECTED RESULT	ACTUAL RESULT	PASS/FAIL
TC-01	MS-NF-1.1	Transaction	The user can create transactions and the amount should decrease from overall balance	Add Transaction screen	Click on add transaction Fill data Click on create transaction	Remark: test Amount:20 Category: salary	Transaction added	Transaction added	PASS
TC-02	MS- F-1.0	Transaction	Proper validation for the remark field	Add Transaction screen	Click on add transaction Fill data Click on create transaction	Remark: Amount:20 Category: salary	Red outline around remark field	Red outline around remark field	PASS
TC-03	MS-F-1.0	Transaction	Proper validation for the amount field	Add Transaction screen	Click on add transaction Fill data Click on create transaction	Remark: test Amount: Category: salary	Red outline around amount field	Red outline around amount field	PASS
TC-04	MS-NF-1.2	Transaction	Toast should appear if transaction couldn't be created	Add Transaction screen	Turn off Wi-Fi Fill data Click on create transaction	Remark: test Amount:20 Category: salary	Toast message	Toast message	PASS
TC-05	MS-F-1.0	Transaction	The user can edit transactions and the amount should change in overall balance	Edit Transaction screen	Click on edit transaction Fill data Click on update transaction	Remark: test_edit Amount:200 Category: salary	Transaction updated	Transaction updated	PASS
TC-06	MS-F-1.0	Transaction	Proper validation for the remark field	Edit Transaction screen	Click on edit transaction Fill data Click on update transaction	Remark: Amount:20 Category: salary	Red outline around remark field	Red outline around remark field	PASS
TC-07	MS-F-1.0	Transaction	Proper validation for the amount field	Edit Transaction screen	Click on edit transaction Fill data Click on update transaction	Remark: test_edit Amount: Category: salary	Red outline around amount field	Red outline around amount field	PASS
TC-08	MS-NF-1.2	Transaction	Toast should appear if transaction couldn't be edited	Transaction screen	Turn off Wi-Fi Fill data Click on update transaction	Remark: test_edit Amount: 10 Category: salary	Toast message	Toast message	PASS
TC-09	MS-F-1.0	Transaction	The user should be able to delete a transaction	Add Transaction screen	Click on delete transaction Click on confirm delete		The transaction should be deleted	The transaction should be deleted	PASS
TC-10	MS-F-1.0	Transaction	A dialog box should appear to confirm the deletion	Transaction screen	Click on delete button		A dialog box should appear	A dialog box should appear	PASS
TC-11	MS-UR-1.1	Transaction	The expenses denoted by red and incomes to be denoted by green	Transaction screen			Colored similarly	Colored similarly	PASS

TC-12	MS-NF-1.2	Transaction	Toast should appear if transaction couldn't be deleted	Transaction screen	Turn off Wi-Fi Click on delete transaction Click on confirm delete		Toast message	Toast message	PASS
TC-13	MS-F-2.0	Correction	User can add a correction to their current bank balance	Correction screen	Enter an amount Click add correction button	300	Decrement in current balance	Decrement in current balance	PASS
TC-14	MS-F-2.0	Correction	User can edit the correction	Correction screen	Click on edit button	250	Correction should get edited	Correction should get edited	PASS
TC-15	MS-F-2.0	Correction	User can delete a correction	Correction screen	Click on the delete button		Correction should be deleted	Correction should be deleted	PASS
TC-16	MS-NF-1.2	Transaction	Toast should appear if correction couldn't be created	Correction screen	Turn WIFI off Enter an amount Click add correction button		Toast message	Toast message	PASS
TC-17	MS-F-3.0	Monthly Saving	User can add a target monthly saving	Settings screen	Click on edit target monthly saving	3000	Monthly Saving to be updated	Monthly Saving to be updated	PASS
TC-18	MS-NF-3.1	Monthly Saving	User should be notified when monthly target saving threshold is reached	Add transaction screen	Enter a value that exceeds the threshold	Remark: test Amount 12000 Category: expense	Alert Dialog	Alert Dialog	PASS
TC-19	MS-UR-3.1	Monthly Saving	The alert dialog box should be easily dismissible	Add transaction screen	Enter a value that exceeds the threshold Click on close		Easily dismissible	Easily dismissible	PASS
TC-20	MS-NF-4.1	Transaction	The transaction should be auto categorized according to the category defined	Add transaction screen	Enter the details and a category	Remark: test Amount:12 Category: Transportation	Transaction categorized as transportation	Transaction categorized as transportation	PASS
TC-21	MS-F-4.0	Category	User can add categories	Add category screen	Click on add category button Fill details Click on create category		New category added	New category added	PASS
TC-22	MS-F-4.0	Category	Adding category should have validation for the category name	Add category screen	Click on add category button Fill details Click on create category	Category name: Related words: hi, hello, bye	Red outline around the category name field	Red outline around the category name field	PASS
TC-23	MS-NF-4.3	Related Word	Adding category should have validation to add at least select three related words to create category	Add category screen	Click on add category button Fill details Click on create category	Category name: meeting Related words: hi, hello	Toast message to insert at least three fields	Toast message to insert at least three fields	PASS
TC-24	MS-NF-4.3	Related Word	Clicking on the add related word button should add a text field to add more words	Add category screen	Click on add category button Fill details Click on add button		New text box added	New text box added	PASS

TC-25	MS-NF-4.3	Related Word	Clicking on remove related word should remove the word	Add category screen	Click on add category button Fill details Click on create category		Related word field removed	Related word field removed	PASS
TC-26	MS-NF-4.3	Related Word	Three text fields for minimum number of related words should be already present	Add category screen	Click on add category button		Already present	Already present	PASS
TC-27	MS- F-4.0	Category	User can edit categories	Category screen	Click on an existing category edit details Click on update category		Category updated	Category updated	PASS
TC-28	MS-NF-4.3	Related Word	Editing category should have validation to add at least select three related words to update category	Edit category screen	Click on edit category button Fill details Click on update category	Category name: meeting Related words: hi, hello	Toast message to insert at least three fields	Toast message to insert at least three fields	PASS
TC-29	MS-NF-4.3	Related Word	Clicking on the add related word button should add a text field to add more words	Edit category screen	Click on add category button Fill details Click on add button		New text box added	New text box added	PASS
TC-30	MS-NF-4.3	Related Word	Clicking on remove related word should remove the word	Edit category screen	Click on add category button Fill details Click on create category		Related word field removed	Related word field removed	PASS
TC-31	MS-F-4.0	Category	User can delete categories	Category screen	Click on existing category Delete category Reassign existing transactions		Category deleted	Category deleted	PASS
TC-32	MS-NF-4.2	Category	User should be asked to recategorize the transactions that are assigned to the category	Category screen	Click on existing category Delete category Reassign existing transactions		Reassign screen	Reassign screen	PASS
TC-33	MS-UR-4.1	Category	User should be provided with add category option when they are adding transaction	Add transaction screen	Click on the green add icon beside the dropdown button		Diverted to add category screen	Diverted to add category screen	PASS
TC-34	MS-F-5.0	Sort Transaction	Sort transaction by time or amount	Transaction screen	Click on sort button	Time	Sorted accordingly	Sorted accordingly	PASS
TC-35	MS-F-6.0	Filter Transaction	Filter transaction by day, week, month and by year	Transaction screen	Click on filter button	Day	Filtered accordingly	Filtered accordingly	PASS

1.3 Detailed Analytics System (MS)

Req. Code	Requirement Description	Use Case	MoSCoW Prioritization
DS-F-1.0	The User can view the detailed analysis for the transactions made in pie chart with respect to their category	Add Pie chart view	Must Have
DS-NF-1.1	The amount of expense for specific categories are calculated for the month or year	Calculate the category wise expense	Must Have
DS-UR-1.1	The indexing is provided for colours with respect to the category	Add indexing	Must Have

2. Use Case

The app is defined with two roles in order to define the functional requirements of the application constituting the user and a database. The user is visualized to be a singular person but can be further developed for a family or even an organization. The second role played is rather a database which would constitute both local storage and a server in order to constitute the functional capabilities to store user data.

The sole functionality for the mobile application will lie upon the financial tracking of the user with correspondence to the data that is fed. Furthermore, along with keeping the data intact and categorized, the data will also be analyzed for a better perception and comparison of the user's personal incomes, savings and expenses featured with predictions and warnings in case of a potential overhead.
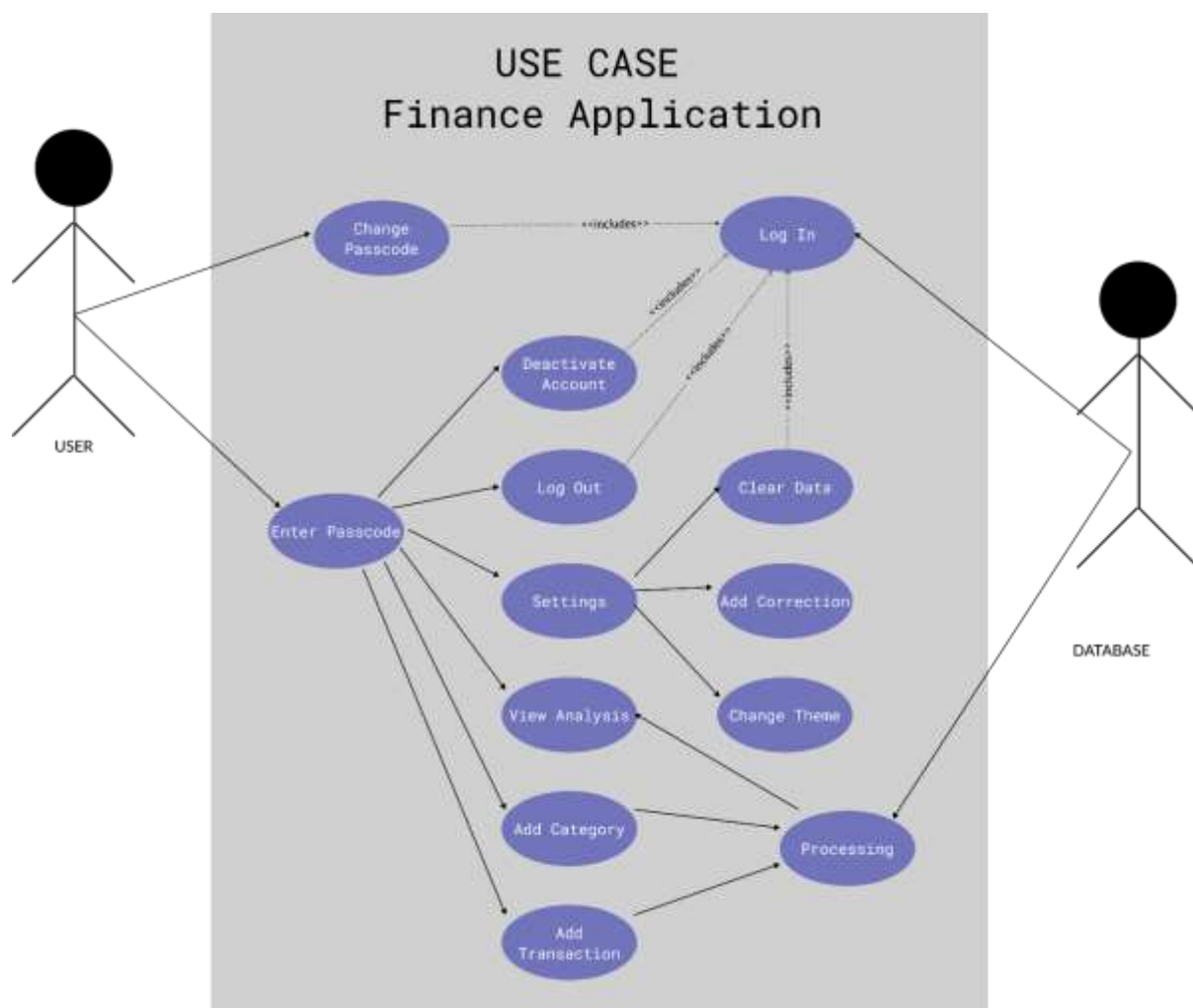
Figure 4: Functional Overview

3. Wireframe

The wireframe below is made as the basic construction of how the application would look like at its core. It constitutes of most envisioned functionalities for the application as well as all of the non-functional requirements set for the application. It represents the UI portion of the application assuring certain usability principles provided the intended functionalities.

Green lines have been drawn in order to form a connection between the initialization of the screens and similarly yellow notes have been provided in order to clarify the purpose for definitive choices for a better understanding of the non-functional terms. A more in-depth description has been provided below for major screens.
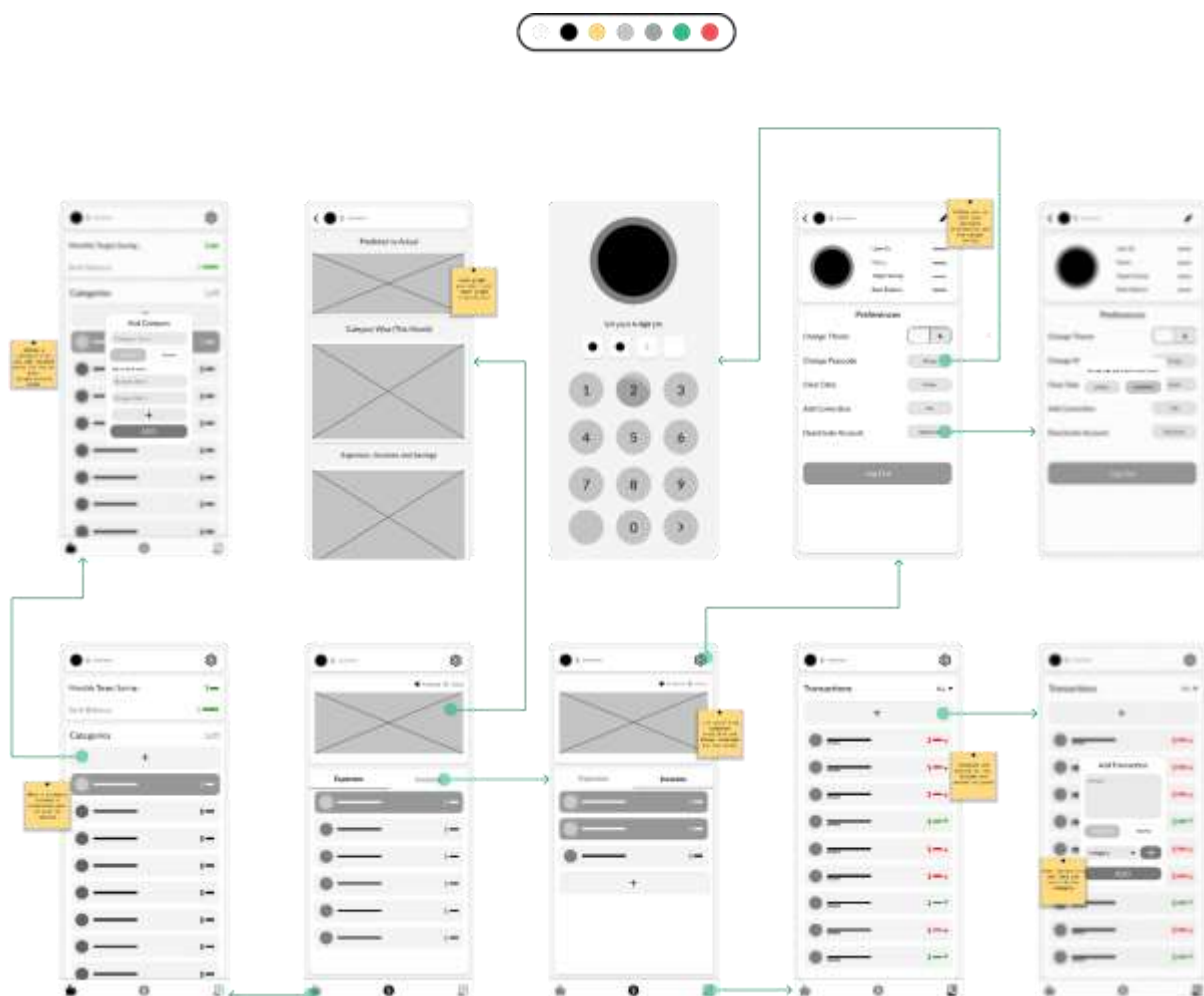
Figure 5: Connections in wireframe
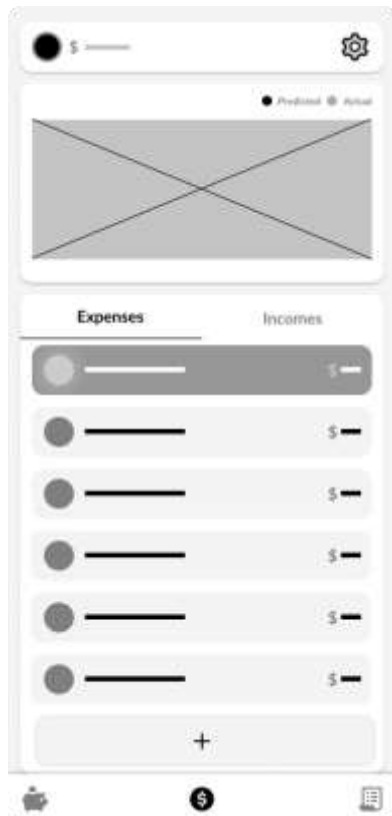
Figure 6: Home Screen

3.1 Home Screen

The home screen of the finance application is the first screen seen by a user unless a passcode has not been set. This will constitute of a double line graph on the top card that will represent the predicted expense and the other graph will represent the actual expense for the user. The graphs will be drawn solely on basis of collective data of his personal past transactions occurred for last few months.

The bottom card will consist of tab view for expenses and incomes that will constitute of a categorized dataset of transactions performed by the user.

3.2 Transaction Screen

Transaction screen is the third screen on the bottom navigation bar and it provides list of all the transactions sorted descending by the most recently added ones. This will have a drop-down button in order to filter out the list. The transactions that are allocated as expenses are denoted in red and incomes are therefore denoted in green to represent the rise or fall in the overall balance of the user along with the amount by which it acted.

The header on the top most part of the screen represents the balance of the person along with his avatar and a button to navigate to the settings screen.
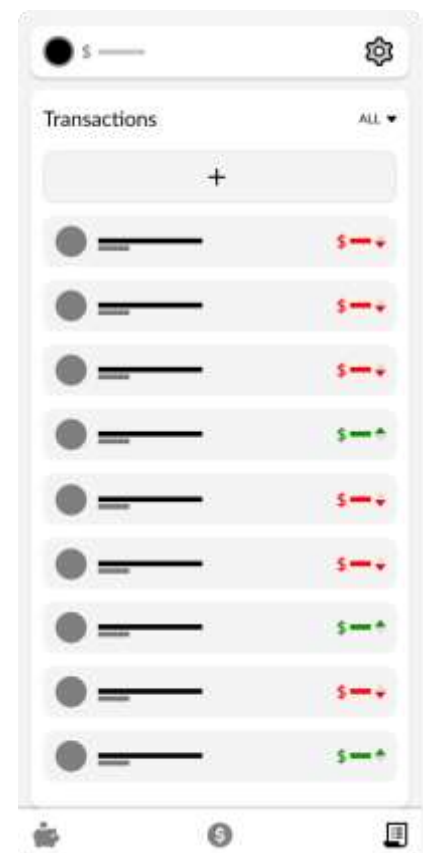

Figure 7: Transaction Screen

3.3 Category Screen

Figure 8: Category Screen

The category screen lists out all of the categories that are present. The application will categorize the natural language text entered by the user as transactions. This will be highlighted red in case an expense category overhead the prediction or highlighted green in case an income category overhead the prediction. Similarly, the user is given freedom to add a category manually by clicking the add button. This list will be sorted in descending order by the amount of the transaction.

3.4 Analysis Screen

Analysis screen can be navigated by clicking on the graph on the home screen and it consists of three graphs that will help the user analyze the personal expenses. The first one will consist of the double line graph that helps the user compare his actual expenses with the predicted expense.

Similarly, the second card will have a pie chart that will represent the expenses category wise so that he can make a more informed decision to cut his expenses.

Lastly, the third card will represent a radar chart that will help compare the user's expenses with income and savings for actual data compared with predicted data and data from last month.
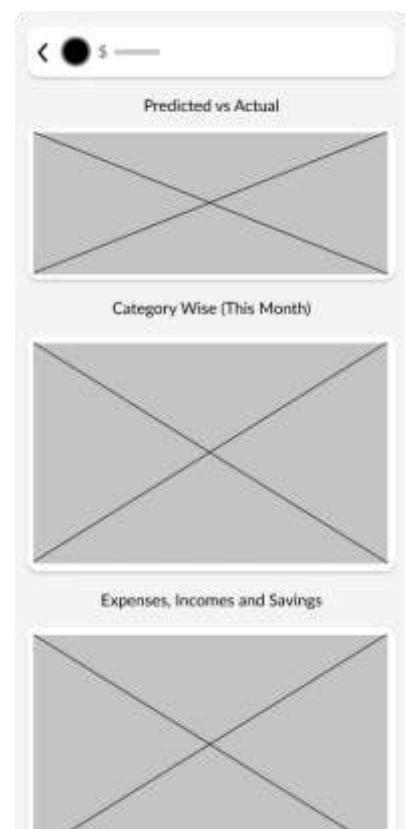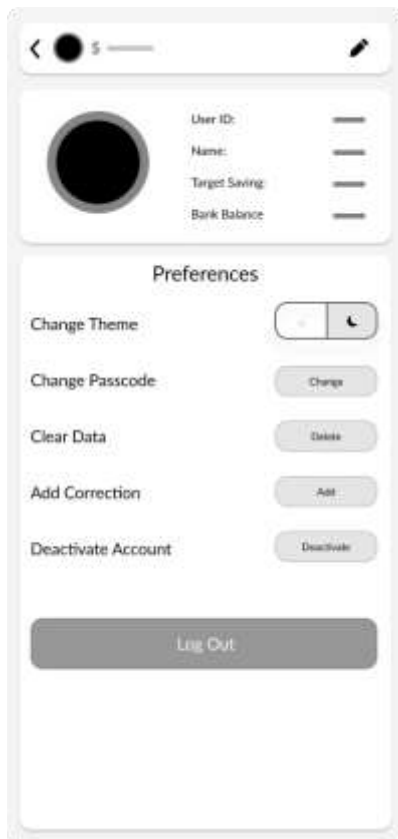

Figure 9: Analysis Screen

5

3.5 Setting Screen

The settings screen can be navigated through header and it will have personal preferences for the user like changing the theme of the app, changing the passcode, clearing the data, adding correction, deactivating the account and changing the avatar.

The user will also be able to change personal details like setting the target saving, bank balance or name. Lastly the user will also be given an option to log out from the account.

Figure 10: Setting Screen

3.6 Lock Screen

Since the application is supposed to have delicate financial details, a lock screen is to be added for security purposes. Setting a lock screen can be navigated through the settings screen and it is protected by a 4-digit number pattern.

The user can set the application to delete it's containing in case for a brute force attempt. If a passcode is set then the screen will be shown at every launch causing it to protect the application from data theft.
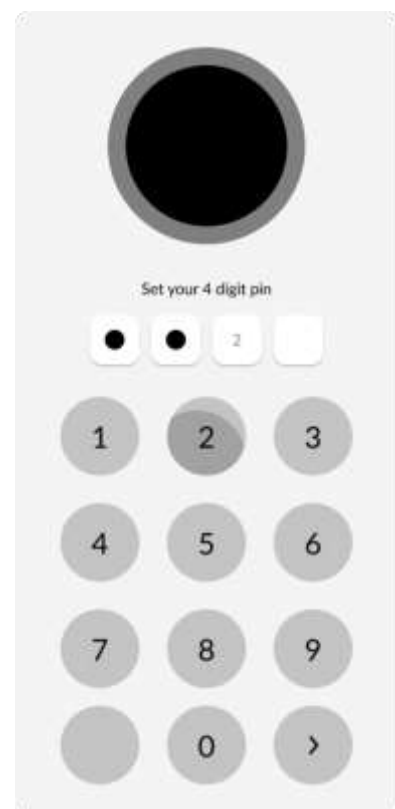
Figure 11: Lock Screen

4. Sequence Diagram

There are three objects in the system during a conventional interaction and they are the User, the database and the view that connects them which will be the application. For an example of the interaction in the application, the user sends the login credentials which is then verified by the O Auth API. If the entered credentials match the one in the database, then the app lets the user enter to use the rest of the application else the user is not let in.

Similarly, when a transaction is entered in the application, it identifies the category with the data that has already been stored in the database with respect to the past entries and the user can see that the transaction is auto categorized.
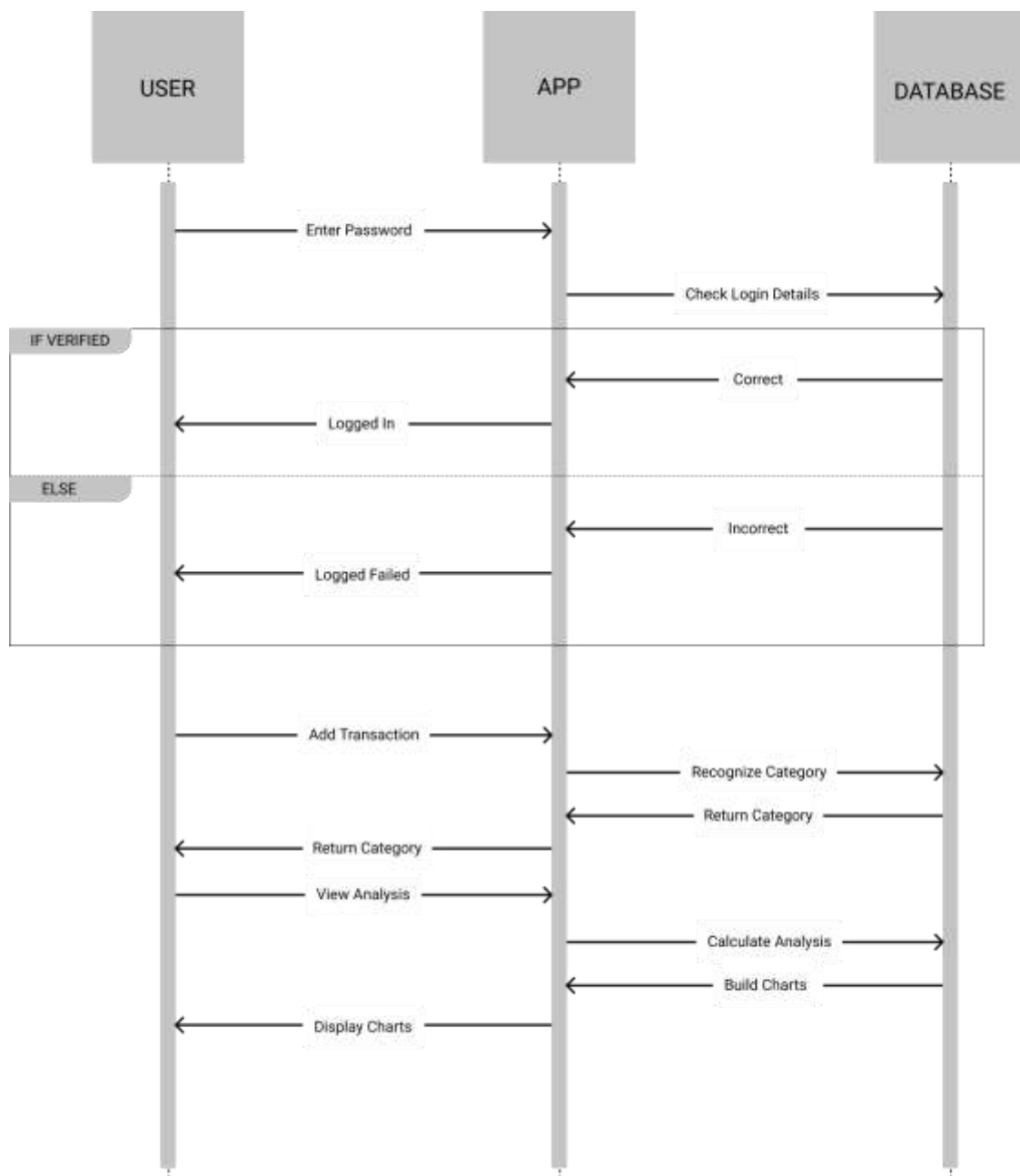

Figure 12: Interaction between objects

5. Entity Relationship Diagram

The diagram shows the construction for the envisioned database for the application. The entity user covers the information like saving, bank balance, preferences and an avatar which will be a type of an emoji which will represent the age and gender of the user. Correction entity is used when the user wants to add a correction to his bank balance because of the miscellaneous transactions which were unable to be noted in the application.

Similarly, transactions have foreign keys like category ID which will be generated programmatically and related word entity will be words that is to be used for the preprocessing section of the natural language processing used to auto categorize the raw text written as remarks when the user adds a transaction. The reason to normalize with a separate entity for theme and gender was to support scalability.

Figure 13: Entity Relationship Diagram