**CP317**

**Project Balance**

**Software Design Document**

**2022-07-12**

**Group 12**

Luke Shi 200608140

Luke Aikman 201652750

Branden Wheeler 190197360

Amirhossein Kompanizare 130783830

Haniah Khan 200687050

Ninam Rai 200619540

Naimo Yasin 201640360

**Contribution Table**

| **Section** | **Contributors** |
| --- | --- |
| Introduction and Design Considerations | Amirhossein Kompanizare |
| Architectural Design | Naimo Yasin |
| Data Design | Branden Wheeler |
| Class Diagrams | Ninam Rai, Branden Wheeler |
| Sequence Diagrams | Haniah Khan |
| Human Interface Design | Luke Aikman, Luke Shi |

Table Of Contents

1.0 Revision History…………………………………….. 3

2.0 Introduction………………………………………….. 3

2.1 Purpose……………………………………….. 3

2.2 Scope…………………………………………. 3

2.3 Definitions and Acronyms …………………… 3

2.4 References……………………………………. 3

2.5 Overview……………………………………….. 3

3.0 Design Considerations………………………………. 4

3.1 Assumptions/Constraints/Dependencies……… 4

3.2 Risks…………………………………………… 4

3.3 Operational Environment……………………… 4

4.0 Architectural Design…………………………………. 5

5.0 Data Design………………………………………….. 6

5.1 Data Description……………………………… 6

5.2 Data Dictionary……………………………….. 7

6.0 Component Design………………………………….. 8

6.1 Class Diagrams………………………………. 8

6.2 Sequence Diagrams………………………….. 12

7.0 Human Interface Design……………………………. 16

7.1 Sign In / Create Account ………………………16

7.2 User Information for Signing up / Creating Account … 17

7.3 Home Page / Upload and Add Receipt / Settings page … 18

7.4 View Receipts ………………………………..20

7.5 - View receipt and categorize receipt …………21

**1.Revision History**

**1.1** The software requirements specification is updated to include Use Case 3.2.11: Receive Share. Document revised by Branden Wheeler on July 10, 2022

**1.2** No revisions have been made to the original project proposal document

**2.Introduction**

Balance is a mobile application that transfers the receipt from the Point-of-sale to the mobile devices that support NFC. It uses a programmable NFC tag that can be reused by the retailer. Balance let the user log in, record the receipts, organize them and later access them. This design document outlines the many aspects of the design of Balance.

**2.1 Purpose**

This design document contains the details of the architecture and data design of our app Balance.

**2.2 Scope**

The scope of this document is to help developers and designers with steps they need to take to develop and maintain the Balance app.

**2.3 Definitions and Acronyms**

**User:** The person who will be using our product.

**Seller:** A user with a seller account who is given the extra functionality of being able to create receipts for sending

**Buyer:** A user with a buyer account who does not have the extra functionality of being able to create receipts for sending

**Receipt Data:** Data of purchase receipt

**API:** Application programming interface; which allows two applications to communicate with each other

**NFC:** The near-field communication is what enables the communication between two electronic devices; this is the technology used for tap payments.

**POS:** Stands for Point of Sale

**OS:** Stands for Operating System

**2.4 References**

CP317 Group 12. *Balance Software Requirements Specification.*

IEEE. *IEEE Std 1016-1998 IEEE Recommended Practice for Software Design Descriptions*. IEEE Computer Society, 1998

**2.5 Overview**

This document is written by following IEEE recommended practice for software Design document. It contains the data model, data description and other requirements.

**3. Design Considerations**

**3.1 Assumptions/Constraints/Dependencies**

* Assumptions include users have to be able to read NFC tags by having NFC support on their phone.
* One of the most important constraints we are facing is the limited space available on an NFC tag which is about a KB.
* Another important constraint is that since we have used relational databases it restricts the way we are able to communicate with the database server.
* Dependencies include Python API for NFC and managing to run python in an android app. Which is not one of the default languages supported by Android Studio.

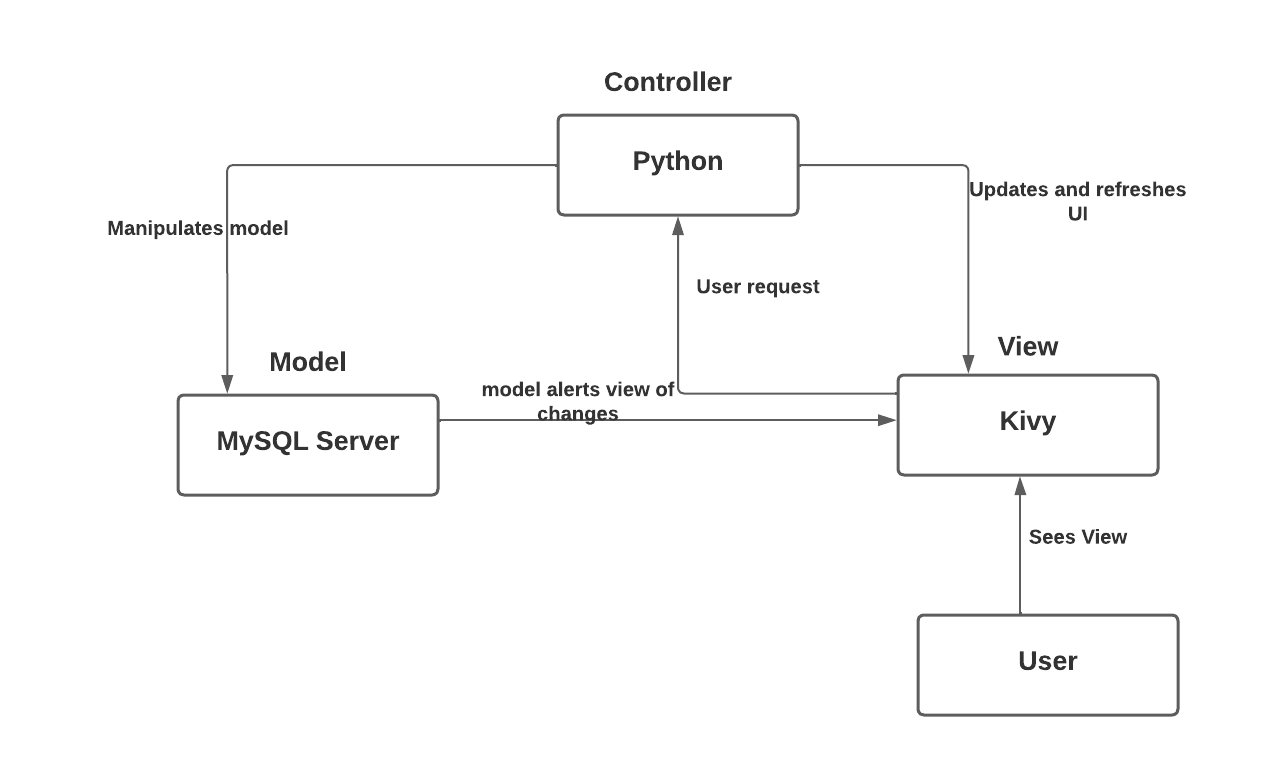
**3.2 Risks**

* In terms of risks the main one would be if there is a newer technology that phones use for transferring data remotely and no longer use NFC. In this case Balance will no longer be able to function since it functions using NFC.
* Since we are storing our data on databases there might be risk of data being breached or lost.
* Balance requires POS systems to insert a NFC transfer system if stores cannot afford to add that to their POS then Balance will not be able to operate.

**3.3 Operational Environment**

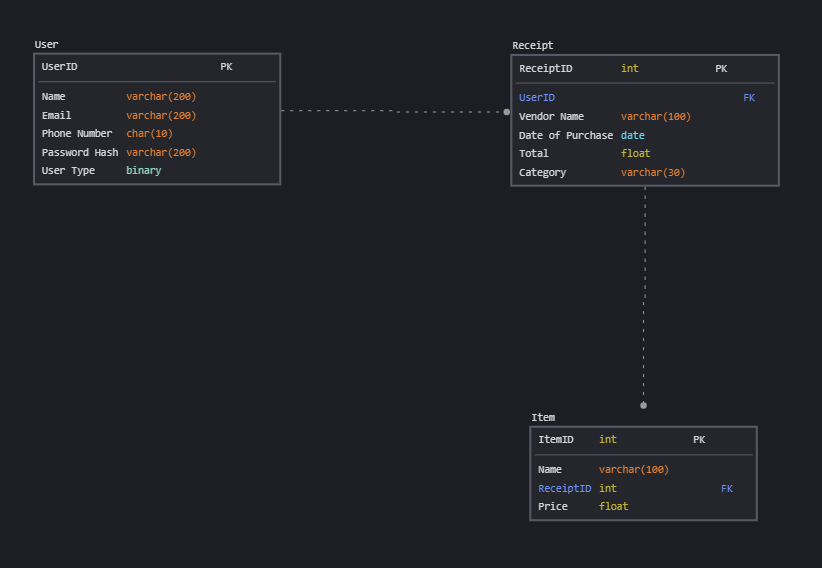
Balance is a multi-platform app that is using Python Api to convert the NFC tag data to digital copy of the receipt.

**4.0 Architectural Design**



The figure above shows the model-view-controller pattern for Balance. This design pattern ensures rapid development and testing as it allows each developer to work on a particular section. Our architecture uses Python with the Kivy framework. Kivy handles user interaction and presents data to the user. In our case, python acts as the controller and creates between the Model and the View components. The model is the database where all the data such as receipts, users and its related logic is stored. The user interacts and sees the view. Once the user makes a request, the controller is alerted. The controller then retrieves information from the database and the database updates itself. The model alerts the view of the changes and the response is presented to the user.

**5. Data Design**



**Balance Application Relational Database Diagram**

**5.1 Data Description**

There are several major entity types for this application that will be represented by tables within the relational database system. Namely, users, receipts, and items will each be represented with their own table. Each item in a table will have a unique ID to be used as the primary key for the table.

The User table will store all information relating to any users who have created an account using the application in order to authenticate and authorize users for further use of the application.

The Receipts table will store all information relating to a receipt that has been received either from another user or from a POS terminal at a store. It will be linked to the User table by using the user ID as a foreign key within the table.

The Items table will store all information relating to a single line item that would typically appear on a receipt. It will be linked to the receipt table using the receipt ID as a foreign key as well as the user table using the user ID as a foreign key.

**5.2 Data Dictionary**

User Data Dictionary

| Column | Attribute | Data Type | Example | Constraints |
| --- | --- | --- | --- | --- |
| 0 | User ID | UNSIGNED INT | 12345 | Primary Key |
| 1 | Name | VARCHAR(200) | ‘John Smith’ | NOT NULL |
| 2 | Email | VARCHAR(200) | ‘john@smith.ca’ | NOT NULL |
| 3 | Phone Number | CHAR(10) | ‘5195550212’ | NOT NULL |
| 4 | Password Hash\* | VARCHAR(200) | ‘$2y$10$J644jwJTJM4wckSc34TfUO84jvDQS5vZbybat.mJrBLMl1CqRtddm’ | NOT NULL |
| 5 | User Type | BINARY(1) | 0 for buyer type, 1 for seller type | 0 or 1 ONLY, NOT NULL |

\*In order to make the system more secure, the user passwords cannot be stored in plaintext. Instead, the password will be hashed using a predetermined function and the hashes will be compared so that even if an unauthorized person is able to gain access to the user table they will not get access to anyone’s account.

Receipt Data Dictionary

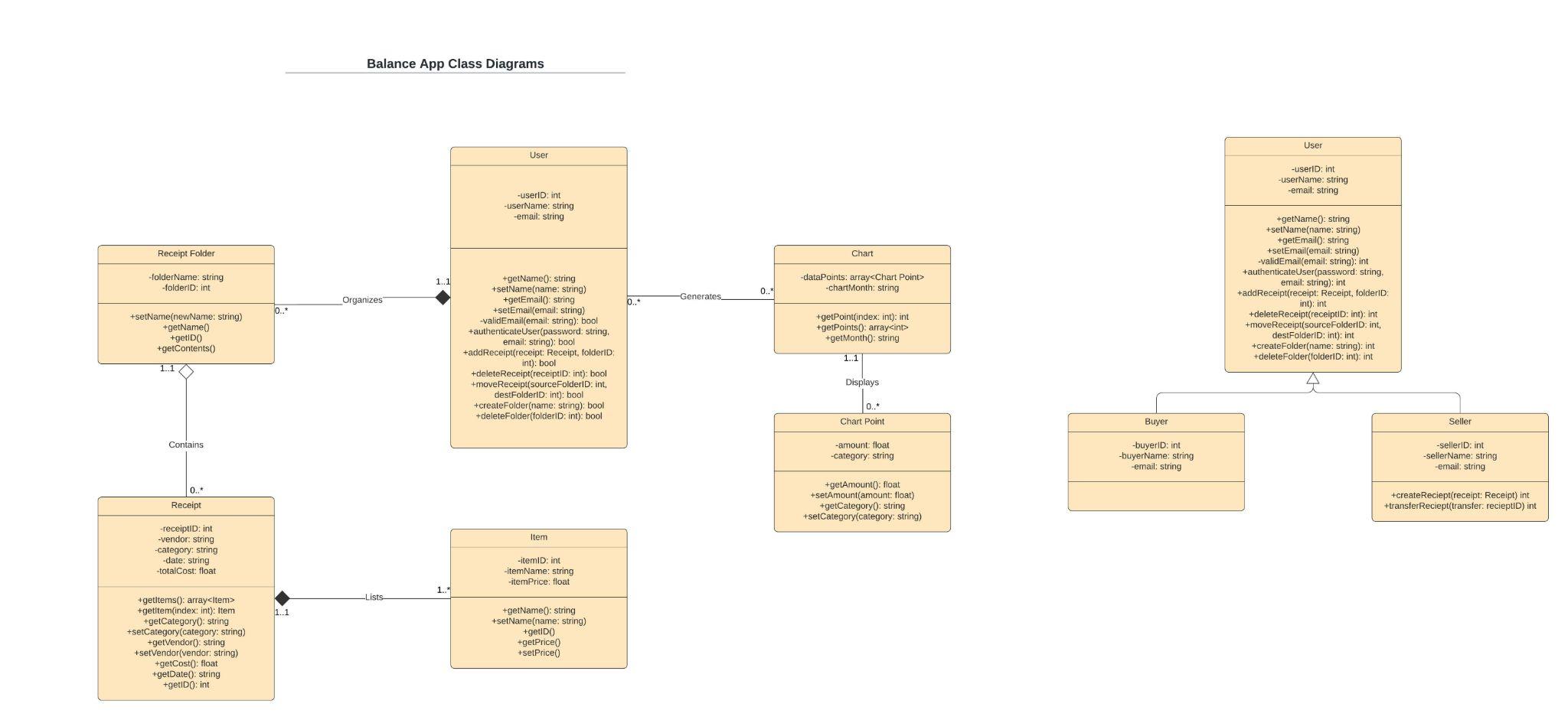
| Column | Attribute | Data Type | Example | Constraints |
| --- | --- | --- | --- | --- |
| 0 | Receipt ID | UNSIGNED INT | 99999 | Primary Key |
| 1 | User ID | UNSIGNED INT | 12345 | Foreign Key |
| 2 | Vendor Name | VARCHAR(100) | ‘Wal-Mart’ | NULL OK |
| 3 | Date of Purchase | DATE | ’14-04-2019’ | NOT NULL |
| 4 | Receipt Total | FLOAT(7, 2) | 123.45 | NOT NULL |
| 5 | Category | VARCHAR(30) | ‘Grocery’ | NULL OK |

Item Data Dictionary

| Column | Attribute | Data Type | Example | Constraints |
| --- | --- | --- | --- | --- |
| 0 | Item ID | UNSIGNED INT | 67890 | Primary Key |
| 1 | Receipt ID | UNSIGNED INT | 99999 | Foreign Key |
| 2 | Item Name | VARCHAR(100) | ‘Tide Pods’ | NOT NULL |
| 3 | Item Price | FLOAT(7, 2) | 24.99 | NOT NULL |

**6. Component Design**

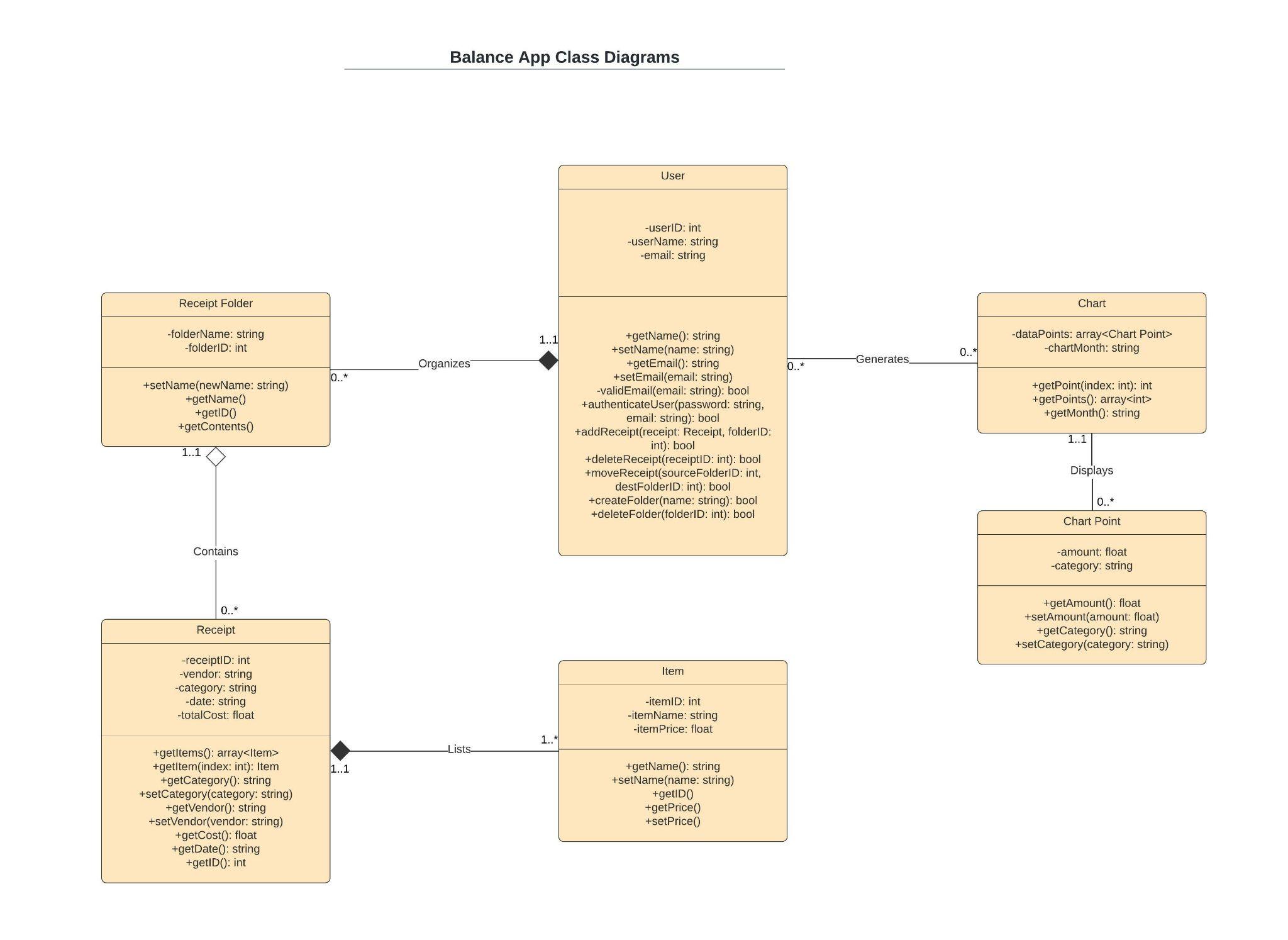
**6.1 Class Diagrams**

****

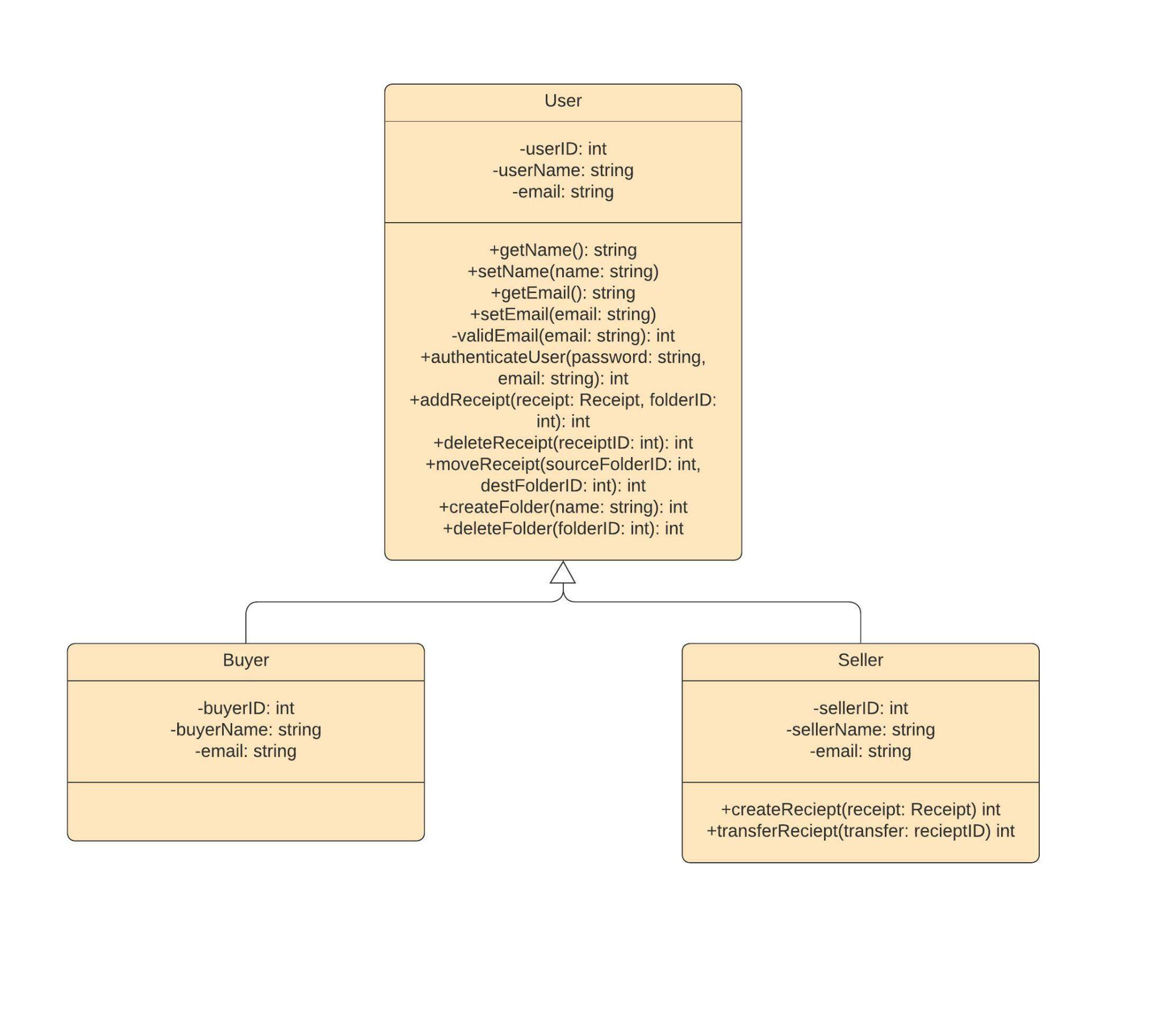
A hyperlink for a better resolution of the image is provided below.

(You will need to make an account to be able to access it)

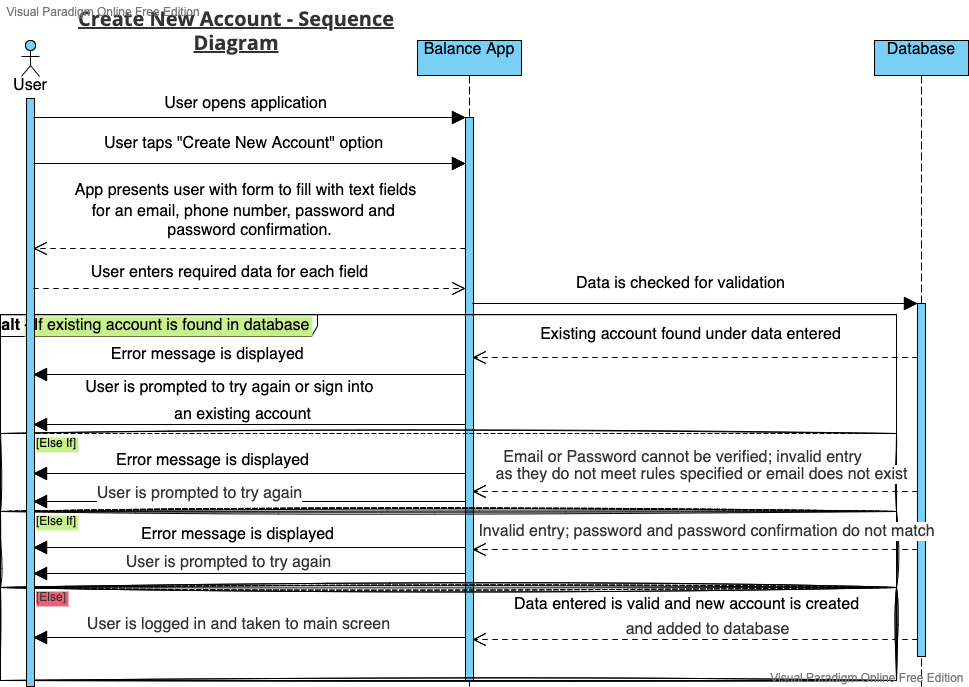
<https://lucid.app/lucidchart/7146d00f-5360-467b-bc22-487f66ebc745/edit?viewport_loc=-148%2C80%2C2219%2C1108%2C0_0&invitationId=inv_9fb57444-3151-4919-bacb-b2dc36ad3bc3#>

Main Class Diagram

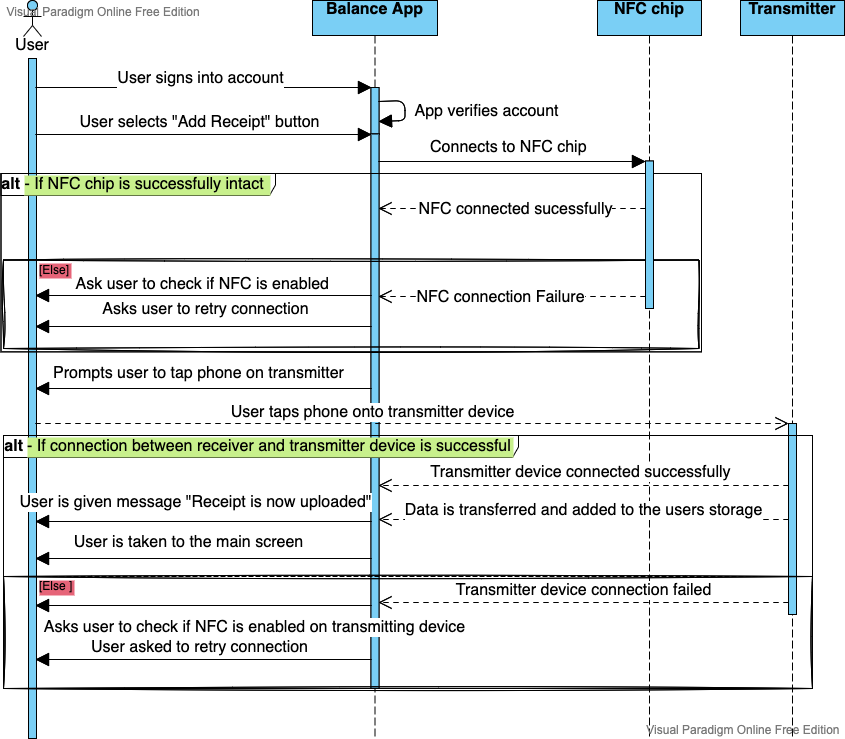
User Class Diagram

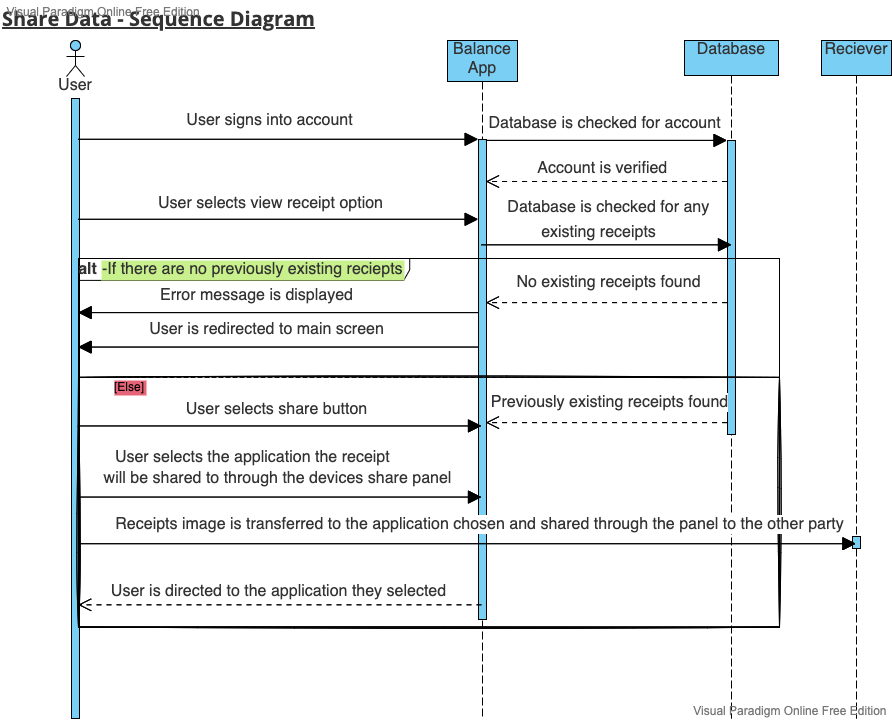
****

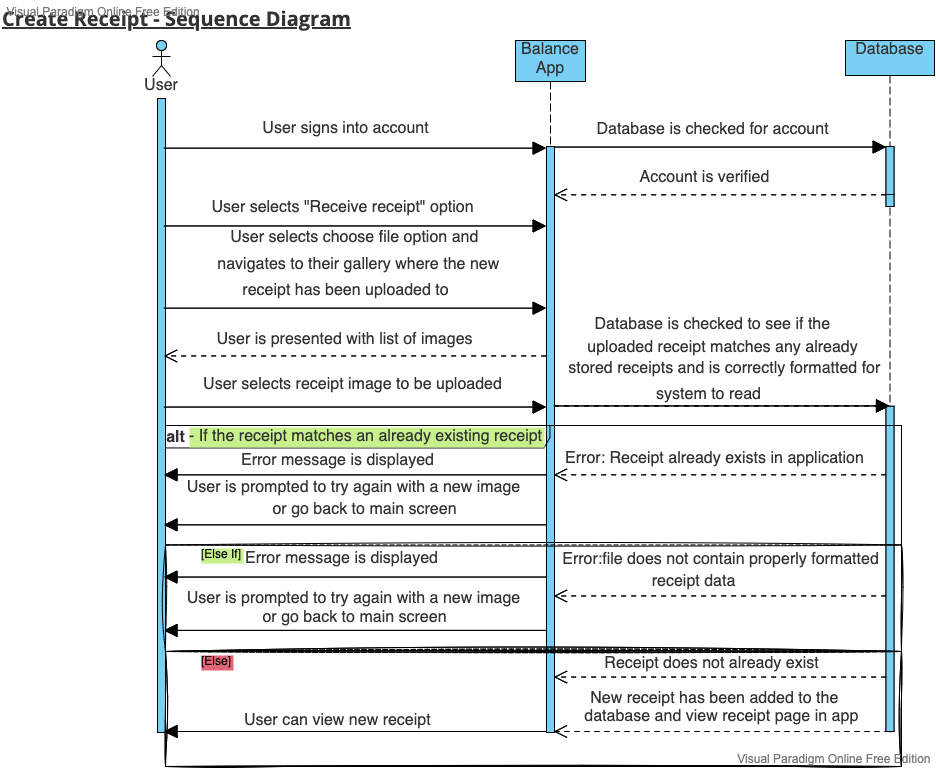
**6.2 Sequence Diagrams**

****

**Tap to Upload - Sequence Diagram**

****

****

****

**7. Human Interface Design**

**7.1 - Sign in / Create Account**

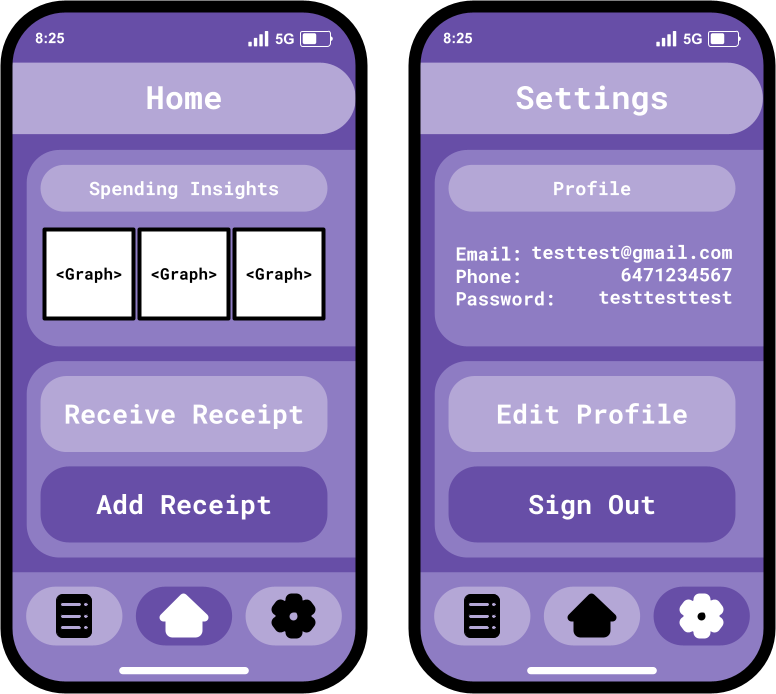
| **Field** | **Type** | **Description** |
| --- | --- | --- |
| Buyer Field | Input button | If the user is a buyer (customer) they will click this button |
| Seller Field | Input button | If the user is a seller (store owner) they will click this button |
| Sign In - (buyer and seller are the same) | Input button | If the user has an existing account, they will use this button to sign in |
| Create Account - (buyer and seller are the same | Input button | If the user does not have an existing account, they will use this button to create an account |



**7.2 - User Information for Signing up / Creating Account**

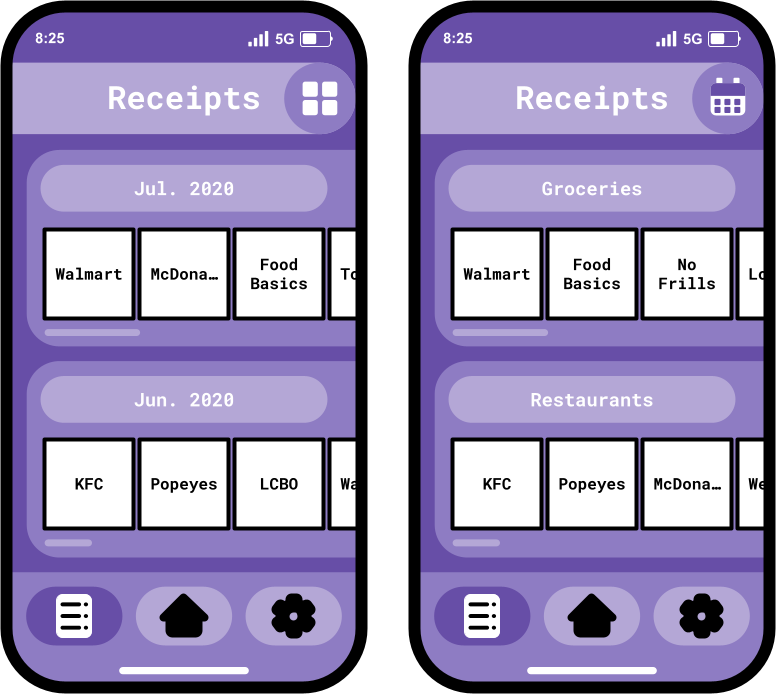
| **Field** | **Type** | **Description** |
| --- | --- | --- |
| Email Field - (same for both images) | Input text | User enters there email in this field |
| Phone Field - (same for both images) | Input text | User enters there phone number in this field |
| Password Field - (same for both images) | Input text | User enters there password in this field |
| Confirm Password Field | Input text | A new user confirms there password in this field |

**7.3 - Home Page / Upload and Add Receipt / Settings page**

****

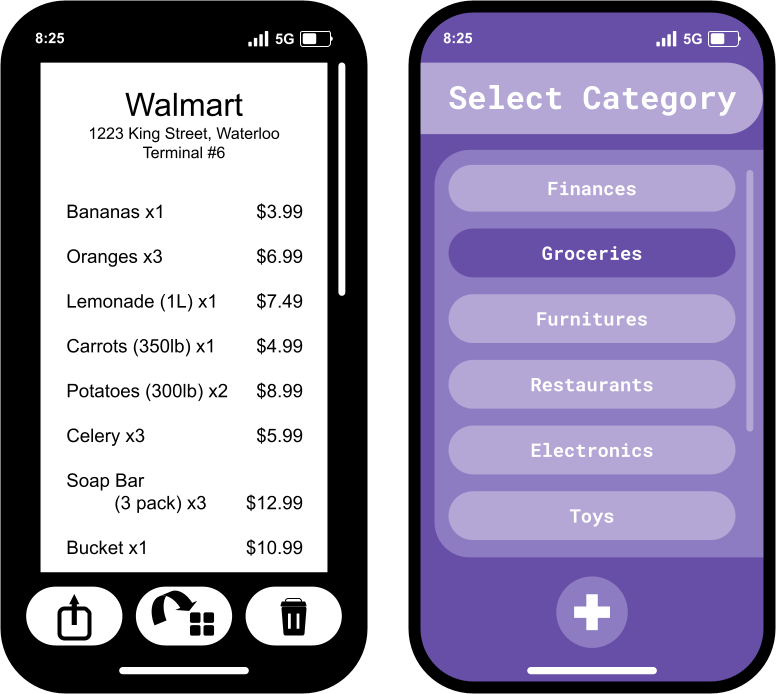
| **Field** | **Type** | **Description** |
| --- | --- | --- |
| Receive Receipt Field | Input button | A user can click this button if they wish to upload a receipt that they received |
| Add Receipt Field | Input button | A user can click this button if they wish to add a new receipt |
| Edit Profile Field | Input button | A user can click this button if they wish to edit their profile information |
| Sign Out Field | Input button | A user can click this button if they wish to sign out |
| View Receipt Field | Input button | A user can click this button if they wish to view a receipt(s) |
| Home Field | Input button | A user can click this button if they wish to go back to the homepage |
| Settings Field | Input button | A user can click this button to view there settings |

**7.4 - View Receipts**



| **Field** | **Type** | **Description** |
| --- | --- | --- |
| Sort by Categories Field | Input button | A user can click this button if they wish to sort there receipts by categories |
| Sort by Dates Field | Input button | A use can click this button if they wish to sort there receipts by date |

**Note: All buttons not mentioned perform the same task as mentioned in a previous section of 7**

**7.5 - View receipt and categorize receipt**

| **Field** | **Type** | **Description** |
| --- | --- | --- |
| Share Receipt Field | Input button | A user can press this button if they wish to share this specific receipt |
| Categorize Receipt Field | Input button | A user can press this button if they wish to change the category of this specific receipt |
| Delete Receipt Field | Input button | A user can press this button if they wish to delete this specific receipt |
| Finances Field | Input button | A user can press this button if they wish to look at finance receipts |
| Groceries Field | Input button | A user can press this button if they wish to categorize as grocery receipts |
| Furnitures Field | Input button | A user can press this button if they wish to categorize as furniture receipts |
| Restaurants Field | Input button | A user can press this button if they wish to categorize as restaurant receipts |
| Electronics Field | Input button | A user can press this button if they wish to categorize as electronic receipts |
| Toys Field | Input button | A user can press this button if they wish to categorize as toy receipts |
| Add a Category Field | Input button | A user can press this button if they wish to add a new category of receipt |