## 

## 

## **CP317 Project**

## **Software Requirements Specification**

Luke Shi 200608140

Luke Aikman 201652750

Branden Wheeler 190197360

Amirhossein Kompanizare 130783830

Haniah Khan 200687050

Ninam Rai 200619540

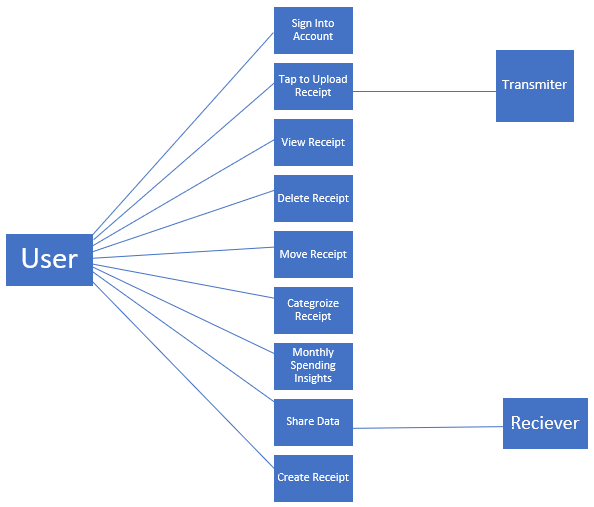
Naimo Yasin 201640360

# Contribution Table

| **Topic** | **Contributor** |
| --- | --- |
| Use Case Diagram | Luke Aikman |
| Introduction | Luke Aikman |
| Purpose | Luke Shi |
| Scope | Luke Shi |
| Definitions | Haniah khan |
| References | Branden Wheeler |
| Overview | Naimo |
| Product Perspective | Amir K |
| Product Functions | Amir K |
| User Characteristics | Ninam Rai |
| Constraints | Ninam Rai |
| Assumptions and Dependencies | Luke Shi |
| External Interface Requirements | Branden Wheeler |
| Security | Naimo |

| **Feature Description** | **Contributor** |
| --- | --- |
| Sign into account - let this be section 3.2.1 | Haniah Khan |
| Create new account | Branden Wheeler |
| Tap to upload Receipt | Branden Wheeler |
| View Receipt | Ninam Rai |
| Delete Receipt | Haniah Khan |
| Move Receipt | Luke Aikman |
| Categorize Receipt | Branden Wheeler |
| Monthly Spending Insights | Luke Shi |
| Share Data | Luke Shi |
| Create receipt | Naimo |

# Use Case Diagram

****

# 1. Introduction

Balance is a phone-based application that allows users to store all their receipts in one easy-to-access location. The idea behind this is that it makes life easier for shoppers when it comes to organizing receipts which is a huge help come tax season. Balance also helps to combat climate change by eliminating the use of paper receipts. Vendors that use Balance should see an uptake in busyness as people will like the organizational factor of Balance and the fact that it promotes a greener lifestyle.

## 1.1 Purpose

This document aims to cover the software requirements for the Balance application. It will also list and explain the features of the application. The intended users for this document are developers of the application and any users or clients looking to gain an in-depth understanding of the application.

## 1.2 Scope

Balance is a mobile application that is used to create, send, receive, store, and display receipts through a digital interface. The two primary users of Balance are the Merchandisers and Shoppers. The merchandisers are able to create the customer’s receipt through an interface. The receipt is then stored and transferred through a purchase POS’s NFC tag. The shopper’s mobile device will then receive the digital receipt and store it on the phone. This digital receipt can then be displayed through the purchases tab in the Balance application.

The application will be accessible to the shoppers via the free app on both Android and iOS and the merchandiser’s terminal will be available through paid partnerships. The application will distinguish between users through an account system that requires the user to create a username and password.

## 1.3 Definitions

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

**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

## 1.4 References

IEEE. *IEEE Std 830-1998 IEEE Recommended Practice for Software Requirements Specifications*. IEEE Computer Society, 1998

*Specifications*. (n.d.). NFC Forum. Retrieved June 19, 2022, from <https://nfc-forum.org/build/specifications>

## 1.5 Overview

The rest of this document contains details that highlight the functionality of Balance. More specifically, it describes the details pertaining to the various features offered by the application.

# 2. Overall description

Balance helps users save time when it comes to managing their receipts. This product digitalizes the receipts that consumers get from any location that provides the service and transfers the data to mobile devices.

## 2.1 Product Perspective

This product does not perform independently, it requires one NFC tag that contains the receipt’s details. The NFC tag could be reused and reprogrammed by the point-of-sale device at the store. Therefore, Balance can transfer the receipt to the user in moments without the use of any cables or extra hardware. This results in much less paper consumption on a large scale and reduces clutter.

## 2.2 Product Functions

## Balance is a phone-based application. For users to access their data they obviously have to have the app downloaded on their mobile device. Once the app is opened the user will be prompted to sign in if they are a current user. Otherwise, they are prompted to create a new account. The functionalities for the user are as follows, create a new account, sign in to their account, upload receipt, move receipt, and manage receipt. For more details refer to the later sections of this document.

#### **2.3 User Characteristics**

Balance is meant to be used by any and every kind of shopper. Anyone who has ever bought something and needed to track their spending can find our product useful. Our app does not need any specific type of skills or knowledge to be able to easily navigate through it. Although, familiarity with other apps does make it easier for users to grasp the concept of Balance quicker. In addition, any experience with charts and graphs would also make it easier for users to understand the monthly overview of their spending.

#### **2.4 Constraints**

One of the major constraints that we face are the merchandisers. If we are not able to expand across several merchandisers or join hands with major corporations like Walmart, we might lose the interest of our customers. This could ruin our feature of convenience because it could be troublesome for our clients to check which store our product is associated with. Another limitation to our product comes in the form of development. Since we are only able to use object-oriented languages to develop our product, it blocks several other paths we could have taken to make our product superior and more efficient.

#### **2.5 Assumptions and Dependencies**

It is assumed that all user devices with this application installed have Near-Field Communication (NFC) and that the NFC component is enabled. The user must also have sufficient storage on their device for the application to run and store the digital receipts.

Furthermore, for the transfer feature to work, it is assumed that the shopper’s mobile device and the merchandiser’s terminal have a direct line of contact for the digital transfer to occur.

### **3. Requirements Specification**

#### **3.1 External Interface Requirements**

##### **3.1.1 User Interfaces**

###### **3.1.1.1**

The user shall interact with the application via a graphical front-end interface. The application will be built for mobile devices (tablets, and smartphones) so the actions will be dictated by the user by taps on various functional components of the interface. These taps will then trigger a specified function.

##### **3.1.2 Hardware Interfaces**

###### **3.1.2.1**

The application shall receive and send data externally with the NFC chip embedded in the mobile device via an API

###### **3.1.2.2**

The application shall receive data from the NFC transmitter responsible for sending the receipt information to the mobile device

##### **3.1.3 Software Interfaces**

###### **3.1.3.1**

The application shall store and retrieve data from a relational database system.

###### **3.1.3.2**

No create, update, read, or delete operation on any of the tables within the database shall take longer than 5 seconds from request to response.

##### **3.1.4 Communications interfaces**

###### **3.1.4.1**

The application shall utilize the near-field communication (NFC) wireless standard for transmitting receipt information between devices.

###### **3.1.4.2**

The application shall successfully transmit information between compatible devices at a range of direct contact to 4 centimeters of separation between the devices.

#### **3.2 Functional Requirements**

##### **3.2.1 Sign into account**

| **Use Case ID** | BL-1 |
| --- | --- |
| **Use Case Name** | Sign into account |
| **Actors** | User (Primary) |
| **Use Case Overview** | Allows the user to sign in to an existing account to access their previously-stored data and add new receipts. |
| **Preconditions** | The user has the application installed and has previously created an account. |
| **Postconditions** | After the use case is complete, the user is authenticated, the user is taken to the main screen. |
| **Use Case Associations** | None |
| **Normal Flow** | 1. The use case begins when the user opens the application 2. From the welcome screen, the user taps the “Sign into Account” option 3. The app then presents the user with a form with two text fields for their email/phone number and password. 4. The user then taps the ‘Sign in’ button 5. The user waits a few seconds while the application validates the data entered 6. The user is logged in and taken to the main screen of the app |
| **Alternative Flow** | **A1: Account does not already exist**  If in step 5 the application finds no existing account with the email/phone entered by the user then  1. An error message will be displayed  2. The user is prompted to try again or create a new account  **A2: Invalid email/phone number formatting**  If in step 5 the application finds that the email/phone number provided does not meet the requirements specified then  1. An error message is displayed  2. The user is prompted to try again  **A3: Incorrect password**  If in step 5 the application finds that the provided password not match the password on record with that corresponding email/phone number  1. An error message is displayed  2. The user is prompted to try again |
| **Priority** | High |

##### **3.2.2 Create New Account**

| **Use Case ID** | BL-2 |
| --- | --- |
| **Use Case Name** | Create New Account |
| **Actors** | User (Primary) |
| **Use Case Overview** | Allows the user to create a new account in order to use the application’s functionality |
| **Preconditions** | The user has installed the Balance application to their phone |
| **Postconditions** | After the use case is complete, the user is authenticated, the user is taken to the main screen, and the new user data has been added to the database |
| **Use case associations** | None |
| **Normal Flow** | 1. The use case begins when the user opens the application  2. From the welcome screen, the user taps the “Create New Account” option  3. The app presents the user with a form with text fields for an email, phone number, password, and password confirmation  4. The user enters the required information into each field  5. The application validates the data  6. The user is logged in and taken to the main screen of the app  7. The user data is added to the database |
| **Alternative Flow** | **A1: Account already exists**  If in step 5 the application finds an existing account with the email entered by the user then  1. An error message is displayed  2. The user is prompted to try again or sign into an existing account  **A2: Invalid email or password formatting**  If in step 5 the application finds that either the email or password provided does not meet the rules specified then  1. An error message is displayed  2. The user is prompted to try again  **A3: Mismatched passwords**  If in step 5 the application finds that the provided password and password confirmation do not match then  1. An error message is displayed  2. The user is prompted to try again |
| **Priority** | High |

##### **3.2.3 Tap to Upload Receipt**

| **Use Case ID** | BL-3 |
| --- | --- |
| **Use Case Name** | Tap to Upload Receipt |
| **Actors** | User (Primary), Transmitter (Secondary) |
| **Use Case Overview** | Allows the user with the receiver device to take in receipt data from the transmitter device via NFC |
| **Preconditions** | The user with the receiver device has opened the application and signed into their account successfully. |
| **Postconditions** | After the use case is complete, the user is returned to the main screen, a new receipt record has been added to the user’s data, and the NFC communication channel is closed |
| **Use Case Associations** | None |
| **Normal Flow** | 1. The use case begins when the user opens the application  2. From the main screen of the app, the user taps the “Add Receipt” option  3. The app connects to the phone’s NFC chip and prompts the user to tap their phone to the transmitting device when the connection is successful  4. The user taps their phone to the device transmitting the receipt data  5. The data is transferred between the two devices and is added to the user’s storage  6. The NFC communication channel is closed by the application  7. The user is returned to the main screen of the app |
| **Alternative Flow** | **A1: Failure to connect to NFC**  If in step 3 the connection to NFC cannot be made within 5 seconds then:  1. The application will notify the user of the failure and tell them to check that NFC is enabled on their device  2. The user will be asked to retry their connection  3. The use case resumes at step 3  **A2: Failure to connect to transmitter device**  If in step 4, the connection between the receiver device and the transmitter device cannot be established within 5 seconds then:  1. The application will notify the user of the failure and tell them to check that NFC is enabled on both the receiving and transmitting devices  2. The user will be asked to retry their connection  3. The use case resumes at step 4  **A3: Quit**  If the user chooses to leave at any point prior to step 5 then  1. The application will return to the main screen |
| **Priority** | High |

##### **3.2.4 View Receipt**

| **Use Case ID** | BL-4 |
| --- | --- |
| **Use Case Name** | View Receipt |
| **Actors** | User (Primary) |
| **Use Case Overview** | Allows the user to view any past receipts that were uploaded. |
| **Preconditions** | The user has downloaded the application, created an account, logged in and has at least one receipt uploaded. |
| **Postconditions** | At the end of the use case, the user is returned back to the page that displays all of their uploaded receipts. |
| **Use Case Associations** | BL-3 (Tap to Upload Receipt) |
| **Normal Flow** | 1. The use case begins when the user is logged in and can see all their uploaded receipts. 2. The user then taps on the receipt and presses the “view receipt” button. 3. The user can now be able to perceive their receipt in full screen. |
| **Alternative Flow** | **None** |
| **Priority** | Medium |

##### **3.2.5 Delete Receipt**

| **Use Case ID** | BL-5 |
| --- | --- |
| **Use Case Name** | Delete Receipt |
| **Actors** | User (Primary) |
| **Use Case Overview** | Allows the user to delete any past saved receipts. |
| **Preconditions** | The user has the application installed, is logged in to an account, and has at least one receipt in their account to delete. |
| **Postconditions** | After the use case is complete, the user is returned to the view receipt screen of the app with the deleted receipt no longer in record. |
| **Use Case Associations** | BL-4 (View Receipt) |
| **Normal Flow** | 1. The use case begins when the user selects the “Delete” option on a specific receipt after opting to view the receipt via BL-4 (View Receipt) 2. A warning message pops up to alert the user that once the receipt is deleted it will no longer be in the database and will not be able to be retrieved. The user will have an option to choose whether to continue with deleting the receipt or returning to the ‘View Receipt’ screen instead. 3. The user selects either option presented. 4. The user is returned to the View Receipt screen. |
| **Alternative Flow** | **None** |
| **Priority** | Medium |

##### **3.2.6 Move Receipt**

| **Use Case ID** | BL-6 |
| --- | --- |
| **Use Case Name** | Move Receipt |
| **Actors** | User (Primary) |
| **Use Case Overview** | Allow customer to move receipt to a specific file / folder |
| **Postconditions** | At the end of the use case the user will be taken back to the main screen where they can either sign off or choose another case. |
| **Use Case Associations** | None |
| **Normal Flow** | 1. Use case begins when the user clicks on the option to move the receipt.  2. Users are asked if they want to move the receipt to an existing file, create a new file or go back to the main menu.  3. User selects an option.  4. Once a file is moved, the user is returned to the main screen. |
| **Alternative Flow** | **A1: Invalid File**  In step two of normal flow if the user selects a file that does not exist or trys to create an already existing file then,  1. An error message is displayed  2. Users are prompted again to select an existing file, create a new one or go back to the main menu. |
| **Priority** | medium |

##### 

##### **3.2.7 Categorize Receipt**

| **Use Case ID** | BL-7 |
| --- | --- |
| **Use Case Name** | Categorize Receipt |
| **Actors** | User (Primary) |
| **Use Case Overview** | Allows the user to sort the items listed on a particular receipt into categories to better track their spending in monthly insights |
| **Preconditions** | The user has the application installed, is logged in to an account, and has at least one receipt in their account to categorize |
| **Postconditions** | After the use case is complete, the user is returned to the view receipt section of the app and the category data is updated in the database |
| **Use Case Associations** | BL-4 (View Receipt) |
| **Normal Flow** | 1. The use case begins when the user selects the “Categorize” option on a specific receipt after opting to view the receipt via BL-4 (View Receipt)  2. The application presents a screen with the items of the receipt listed along with their corresponding category if one has been assigned  3. The user selects an item for which they would like to assign a new category or change an existing category  4. The user is presented with a list of existing category options and a text box to enter a new category  5. The user either selects an existing category from the list or types in a new category name  6. The application updates the database to attach the category to the item  7. The user is returned to the item list screen  8. The user repeats steps 3 to 6 as many times as necessary  9. The user chooses to return to the view receipt screen and they are returned |
| **Alternative Flow** | **A1: Existing Category Name**  If the user enters an existing category name when trying to enter a new category then  1. An error message will be displayed  2. The user will be prompted to try a different name or select the existing one |
| **Priority** | Medium/Low |

##### **3.2.8 Monthly Spending Insights**

| **Use Case ID** | BL-8 |
| --- | --- |
| **Use Case Name** | Monthly Spending Insights |
| **Actors** | User (Primary) |
| **Use Case Overview** | Provides a monthly overview of the user’s purchases. This is sorted into tables and charts for each purchase category like groceries, electronics, food, entertainment, etc. |
| **Preconditions** | * User has the Balance application installed * User is logged in to an account |
| **Postconditions** | After the use case is complete, the user can exit the spending insights tab and continue to use the app or exit back to the home screen. |
| **Use Case Associations** | None |
| **Normal Flow** | 1. The Balance application sends a notification to the shopper at the end of the month 2. The shopper can view the monthly insight through a tab in the app’s menu 3. The application will present the shopper with graphs that show the distribution of purchases between different categories 4. The shopper can then close the monthly insight by clicking the back button or exiting the app 5. If the back button is used, the shopper is returned to the main welcome page of the app 6. If the home button is used, the shopper is returned to the home screen of the device |
| **Alternative Flow** | None |
| **Priority** | Low |

##### **3.2.9 Share Data**

| **Use Case ID** | BL-9 |
| --- | --- |
| **Use Case Name** | Share Data |
| **Actors** | User  Receiver |
| **Use Case Overview** | Allows the user to send their digital receipt image to others via the device’s share panel. |
| **Preconditions** | * User has the Balance application installed * User is logged in to an account * User has at least one digital receipt |
| **Postconditions** | After the use case is complete, the user will stay in the selected application that they shared to. |
| **Use Case Associations** | BL-4 (View Receipt) |
| **Normal Flow** | 1. The shopper navigates to the Receipts tab and selects a receipt 2. The shopper uses the share button, which brings up the device’s share panel 3. The user will select the application that they will share the receipt to 4. The receipt’s image is transferred to the application that the user chooses and shared through the panel 5. The user will be directed to the app they selected |
| **Alternative Flow** | None |
| **Priority** | Medium-High |

##### **3.2.10 Create Receipt**

| **Use Case ID** | BL-10 |
| --- | --- |
| **Use Case Name** | Create Receipt |
| **Actors** | User |
| **Use Case Overview** | User (Primary), Balance (Secondary) |
| **Preconditions** | Allows the user to move the receipt from the user’s gallery into the application main screen. |
| **Postconditions** | The user has the application installed, logged into the account, and has a receipt in the user’s storage device. |
| **Postconditions** | At the end of the use case, the receipt is uploaded and appears on top of the list of receipts on the main screen. |
| **Use Case Associations** | None |
| **Normal Flow** | 1- The use case begins when the user selects the “Create new receipt” option.  2- The user selects the choose file option and navigates to the directory (the gallery) where the new receipt has been uploaded to.  3- The user is presented with a list of images  4- The user chooses the latest image that was uploaded and selects the submit tab  5- The uploaded image now appears at the top of the list on the main screen. |
| **Alternative Flow** | A1: Existing receipt  If the user attempts to upload a receipt that has been already uploaded an error message will be displayed. |
| **Priority** | High |

#### 3.2.11 Receive Share

| **Use Case ID** | BL-11 |
| --- | --- |
| **Use Case Name** | Receive Share |
| **Actors** | User (Primary) |
| **Use Case Overview** | Allows the user to download a text file sent via the Share Data Use Case (3.2.9) and store it with the rest of their receipts |
| **Preconditions** | The user has the application installed and is logged in to an account. They must also have access to their phones internal file system |
| **Postconditions** | After the use case is complete, the user is returned to the home screen, and a new receipt has been added to their account |
| **Use Case Associations** | None |
| **Normal Flow** | 1. The use case begins when the user taps the “Receive Receipt” button on the home screen 2. The app calls the mobile device’s native file system for the user to select a text file containing receipt data 3. When a file is selected it will be parsed to ensure it is a file containing properly formatted receipt data 4. A new receipt object is created and its info is stored in the appropriate relational database table 5. The user is redirected to the home screen |
| **Alternative Flow** | **A1: App fails to call file system**  If the Balance applications fails to successfully call the device’s native file system interface in step 2 then   1. An error will be displayed 2. The user will be instructed to check their connection and try again 3. The use case resumes at step 1   **A2: Improper file formatting**  If the selected file is found to not contain properly formatted receipt data or is a file of the wrong type in step 3 then   1. An error will be displayed 2. The user will be instructed to pick a different file of the correct type 3. The use case resumes at step 2 |
| **Priority** | Low-Medium |

## 

## 3.3 Security

##### **3.3.1 Login Page for existing users**

A login page must be in place to allow users to log in. The only user authentication required must be a valid username and password. Once logged in, a user should be able to access all the features offered by the Application.

##### **3.3.2 General Users**

A general user must not have access to the application’s functionalities without signing up.