Split-It

Software Requirements and Design Specifications

Brian Sokol, Jacob Tucker, Logan Skinner, Tzofi Klinghoffer, Hanlin Ye

April 4, 2018

Version History

Date	Version	Description	Modifier(s)
February 4, 2018	0.01	Created initial outline	Brian
February 6, 2018	0.02	Created detailed outline for each written section	Team
February 7, 2018	0.03	Fleshed out the introduction section and created the functionality comparison chart	Brian
February 12, 2018	0.04	Wrote Product Description using outline as a guideline	Jacob
February 12, 2018	0.05	Wrote Non-Functional Requirements using outline as a guideline	Tzofi
February 12, 2018	0.06	Modified and updated Non-Functional Requirements based on team feedback	Tzofi
February 13, 2018	0.07	Wrote Functional Requirements using outline as a guideline	Logan
February 21, 2018	0.08	Added use case diagram and organized content	Brian
February 22, 2018	0.09	Added activity diagrams for: adding receipt, payment, archiving receipt, and assigning items	Tzofi
February 23, 2018	0.10	Added Activity diagrams for modifying Account, modifying receipt and managing group	Hanlin
February 26, 2018	0.11	Added activity diagrams that Logan and Brian created and put on GitHub into the document: user registration, logging in, and group creation	Tzofi
February 27, 2018	0.12	Added Class Diagram	Jacob
February 27, 2018	0.13	Added Class Diagram explanation	Jacob

March 17, 2018	0.14	Added descriptions of activity diagrams for modifying account, modifying receipt and managing group	Hanlin
March 18, 2018	0.15	Added descriptions and figures of sequence diagrams for modifying account and modifying receipt	Hanlin
March 21, 2018	0.16	Added descriptions of activity diagrams for adding receipt, payment, archiving receipt, and assigning items	Tzofi
March 26, 2018	0.17	Added descriptions of sequence diagrams, and proofed the document, making many small changes	Brian
March 31, 2018	0.18	Added sequence diagrams and descriptions for adding receipt, payment, and assigning items	Tzofi

Table of Contents

I. Introduction

Purpose

Split-It is an Android app that will allow users to easily split costs with their friends, roommates, business partners, etc. Our app will simplify the task of determining who owes who what by keeping track of it for you; all you need is your receipts and our app will do the work for you!

Scope

Our app will run on the Android platform, and we plan to support 80% of retail stores in the U.S. as well as multiple groups of consumers. The app will utilize the blinkReceipt API for processing of receipts into a format we can use. Our app may also use a Venmo API for payments, and use Twillo for user verification. Our project will require a backend server to store user data and enable users to share and edit receipt data. Furthermore, the server will need to maintain state between multiple devices in order to properly handle concurrency issues among users.

Goals

The ultimate goal of this project is to meet all the requirements mentioned in this document, and have an app that is not only functioning properly, but that brings true value to the users. Furthermore, we hope that each feature we implement is done so in the most efficient, and user friendly way possible.

Specifically our goals for this project are being able to successfully extract text from receipts using OCR, being able to perform appropriate actions given the data read from the receipt, performing accurate calculations regardless of the number of items or users in a group, implement profiles and successfully sync the data with the server, give the user the ability to edit their profile, and finally implement a clean, user friendly UI. In addition to these goals, we have a few stretch goals that we hope we have time to implement. These goals include implementing Venmo payments within the app, and possibly even implementing automatic payments at the end of the month to settle costs. Our last stretch goal is to have a highly optimized app. OCR is an intensive process, and we hope that we can optimize this as well as any server communications, so that there is little noticeable delay for the user.

Glossary

- **Consumers**: people that purchases goods or services.
- **Group**: set of users that share a receipt repository and intend to split prices among each other.
- **OCR**: optical character recognition, which is the recognition of printed or written text characters by a computer.
- **Receipt**: statement detailing and acknowledging that something has been paid for or that goods or services were received.
- **Receipt Repository**: collection of all the receipts a user has scanned into the app.
- **Server**: remote device that runs the backend of the app and maintains user data.
- **SKU**: stock keeping unit that is usually in the form of a bar code, which allows the store to track the item in inventory.
- **UPC**: universal product code that can be used to uniquely identify an item.
- **User Account**: collection of personal data that is associated with the current user that functions as the user's method of logging into the application.
- **Vendor**: person or company that sells something to consumers.
- **Virtual Receipt**: virtualized version of the paper receipt shown in the app.

II. Project Description

User Account Creation

Upon first opening the application, users are presented with a login screen having options to either login or create a new account. Once the user has registered an account, the user will be able to link their Venmo username to the account and choose the user's display name.

User Account Management

The user will have the ability to edit their account details through an options menu. Through this menu, display name, Venmo username, and password may be updated (for accounts that do not use social media accounts for login). In addition, a user will be able to delete the user's account.

Group Creation

A user will be able to create a group of users that will split cost. The creator of the group will be the group's owner. The owner names the group during the creation process. The owner may add users into the group using the user's contacts.

Group Management

A group's owner may further add or remove group users after group creation. The owner may change the group's name as well as delete the group entirely. In addition, the owner may allow other group members to add new users to the group.

Receipt Creation

Any user in a group can add a receipt to the group. To add a receipt, the user will take a picture of the physical receipt, then an OCR algorithm will convert the text from the physical receipt to a detailed digital receipt attached to the group. The receipt creator will have an opportunity to confirm the item descriptions and prices. Finally, the receipt creator indicates which group member paid for the receipt in full.

Receipt Management

By default, the owner of the group as well as the creator of the receipt will be able to alter a receipt. These users will be able to extend editing permissions to the other members of the group. Users with permissions, including the owner of the group and creator of the receipt, will be able to edit item descriptions and prices.

Price Assignment

There will be two main options for splitting the costs of a receipt among the users in a group. The first option is an even split among all members of the group. The second option is to assign each item in a receipt to a particular user. Each user then is assigned the aggregate costs of that user's chosen items. Item assignment may take place as users use the items.

User Payment

A user has the ability to pay the purchaser of a receipt through a connection to a Venmo account. If the user does not wish to use Venmo, the user may indicate that the user has reimbursed the purchaser of the receipt through alternative means. In either case, the purchaser receives a notification indicating that the user has paid. The purchaser of the receipt may confirm or deny that the payment was received. The user will then receive a notification of the purchaser's response.

Receipt Archive

Users with receipt editing privileges will be able to archive a receipt once the group is finished with it. Archived receipts will not be visible in the default view of a group; these receipts will become visible using an archive filter, but may not be edited once archived.

III. Functional Requirements

Requirement	Description
Capture image of receipt	Using the phone's camera or photo library, the app will take a picture of a receipt or bill to gather data for the functions of the application. This will require permission to access the phone's camera and photo library.
Convert image to text	Using an OCR API (Blink Receipt), Split-It will process an image create a text file containing the data for parsing. Images can be taken by the camera or use a previously captured image of a receipt.
Parse text data	Parse the text file for necessary information, including itemized prices, total price, and tax percentage (only necessary when splitting less than the total cost). The parsed data will be saved in a separate file.
Perform calculations to split prices	Calculations for price splitting will be based on user input. ● If the total price is being evenly divided between users, the cost will simply be divided by the number of people in the group. ● Paying only for certain items will require the sum of the cost of the items times the tax percentage.
Account creation and management	● Users will set up accounts and link their venmo or other payment service account of their choice. ● Users will be able to login to their account upon opening the app. ● Users can change account settings such as email address, user ID, and password.
Manage consumers and groups	● Users will be able to create groups and delete payment groups. ● Group members will have the ability to request split charges amongst each other, add/remove group members, and rename the group. ● Deleting the group will be restricted to the group administrator or creator.

Request payment application action	Split-It application will make requests to Venmo to initiate payment actions between users of an amount specified by previous calculations.

IV. Nonfunctional Requirements

Requirement	**Description**	**Priority**
Performance	Split-It app should be able to use OCR to scan, process, and virtualize receipts in under 10 seconds each to enable real-time user interaction.	High
Platform	Split-It app should be able to run on currently supported Android operating systems, such as versions 6.0 – 6.0.1, 7.0 – 7.1.2, and 8.0 – 8.1.	High
Reliability	Data in the Split-It app should remain synced between different users in a group, such that when one user is assigned as responsible for part of a receipt, other users can see this change.	High
Accuracy	Any calculations done within the Split-It app must be accurate within one-cent of precision.	High
Performance	Subtotals in the app must sum to the exact total displayed, such that when a receipt is uploaded and then virtualized, all prices of individual items add up to equal the subtotal listed.	High
Usability	The BlinkReceipt API, an API used to digitally scan receipts in real-time, should be integrated into the app to allow users to scan physical receipts.	Medium
Usability	The Split-It app is compatible with Venmo and has built-in Venmo support so users can pay each other through Venmo based on calculations done by Split-It.	Low
Performance	The Split-It app should not drain battery significantly.	Medium
Usability	Split-It app should be able to use either Wifi or data provider (3G, 4G, LTE) connections.	Medium

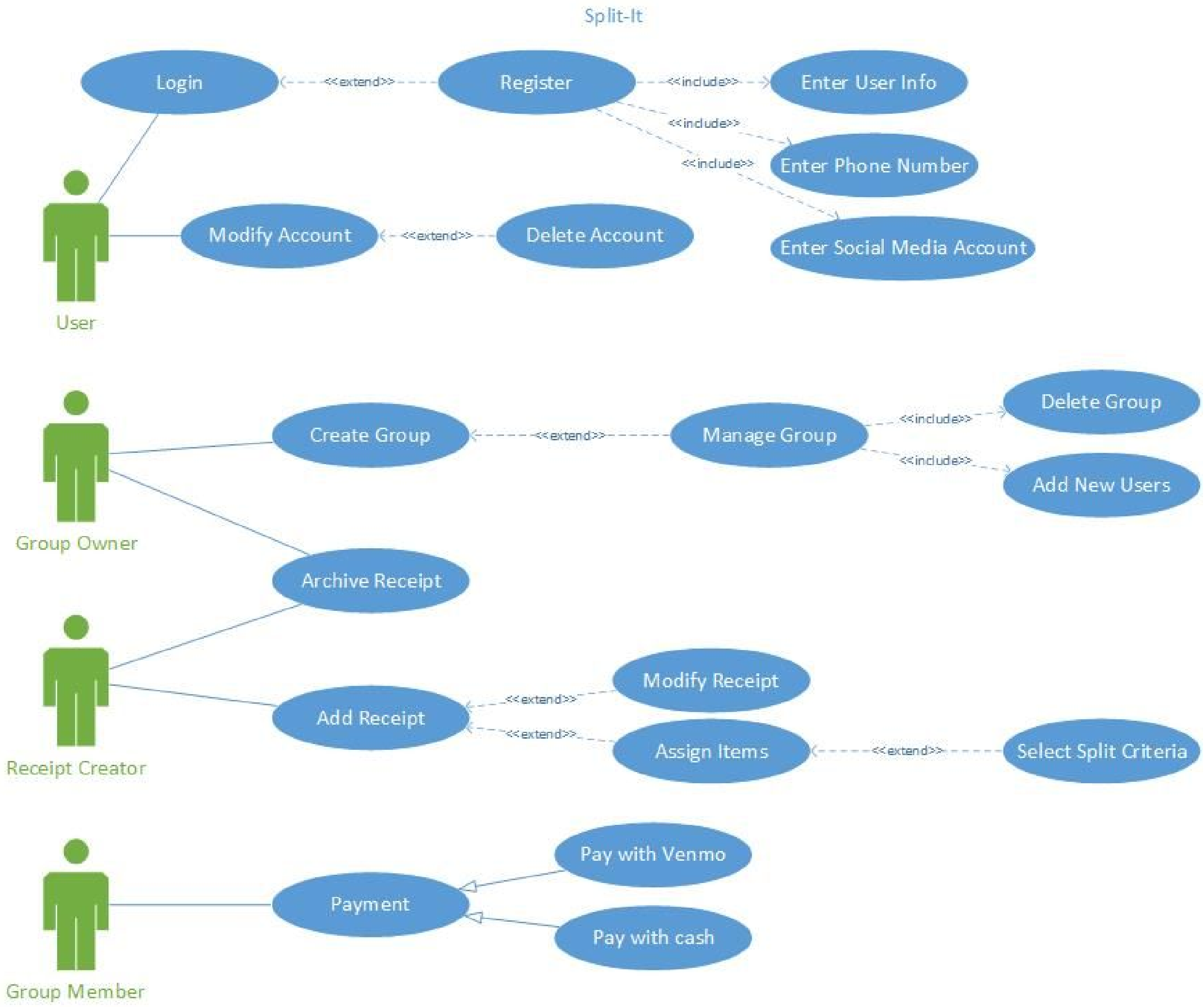
Performance	Persistent data stored in the Split-It app should be retrieved in under 5 seconds after a user makes a request.	Medium
Security	User accounts must be password protected as to ensure sensitive information, such as bank information or Venmo account information, is safe.	High
Security	Sensitive/private information should be stored in a secure database that is protected against basic cyber attacks with a method such as encryption.	High
Usability	The Split-It app functionality and usage should be able to be grasped or understood by users within 10 minutes of use and information on using the app should be included in an easy to find location.	Medium

V. Functionality Comparison

	Scan Receipt Using OCR	Create Digital Receipt	Create/Manage Consumer Groups	Split Costs	Initiate Payment
Splitwise	No	Yes, not itemized.	Yes	Yes	No
Billr	No	Yes	Yes, but just for one receipt.	Yes, but just for one receipt.	No
Divvy	Yes	Yes	Yes, but just for one receipt.	Yes, but just for one receipt.	No
Tab	Yes	Yes	Yes	Yes	No
Splittr	No	Yes	Yes	Yes	No

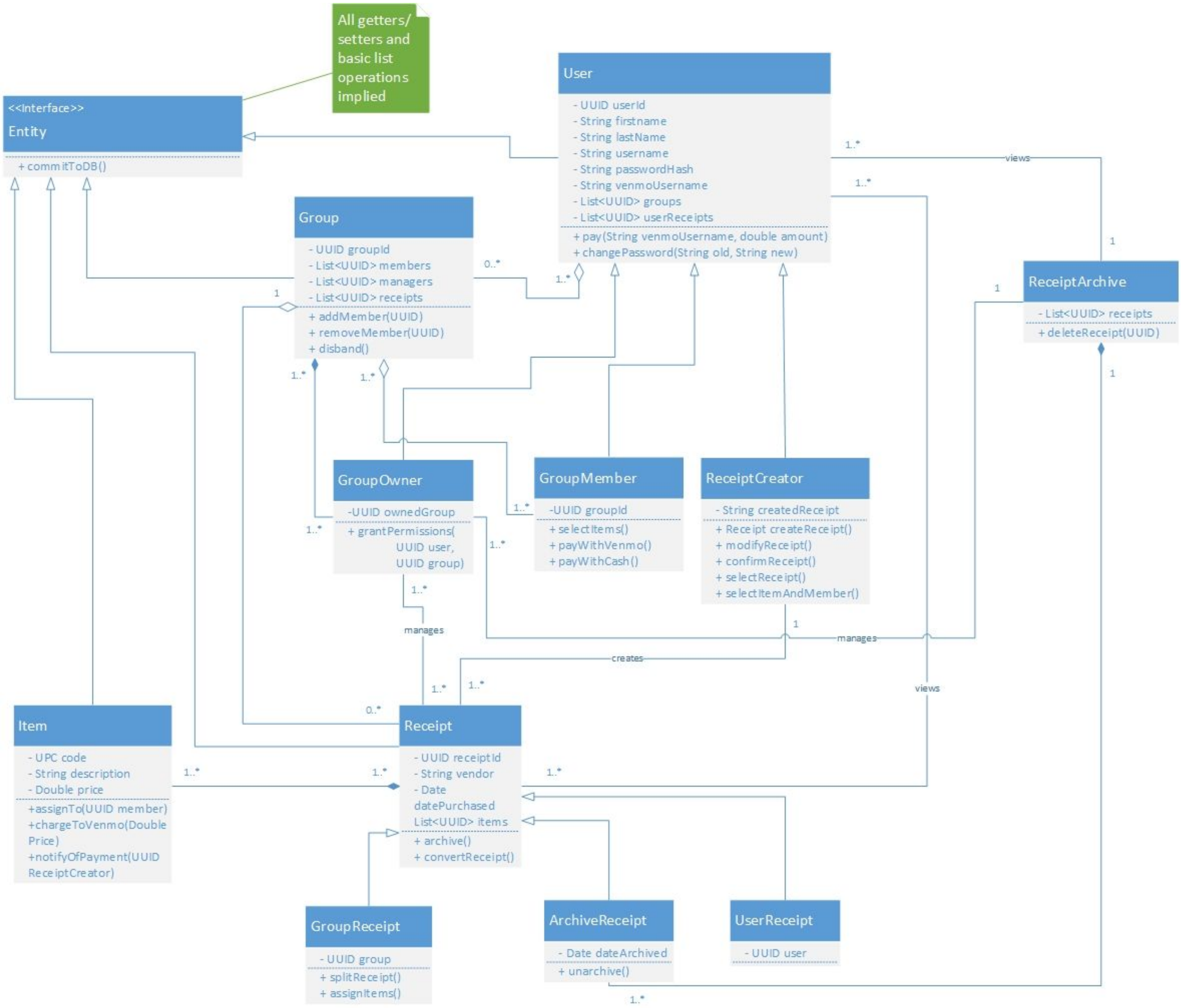
Of all the apps above Splitwise is most comparable to what we are trying to implement. While it does not provide OCR capability, it allows users to manually enter prices for items, and choose from an array of options regarding how they want to split the cost between a group of people. Each person can have their own account, and has access to the shared group. Furthermore, this app has a great UI and overall user experience that we hope to match or surpass. Moreover, our app aims to fill the gap, by providing OCR functionality in combination with the simple cost splitting functionality that you can get from an app like Splitwise.

VI. Use Case Diagram



This use case diagram shows the typical interactions for each of the different user types, or states. There exist four types of users in our App, with a single person possibly occupying one or more of these roles depending on the permissions they have been given.

VII. Class Diagram

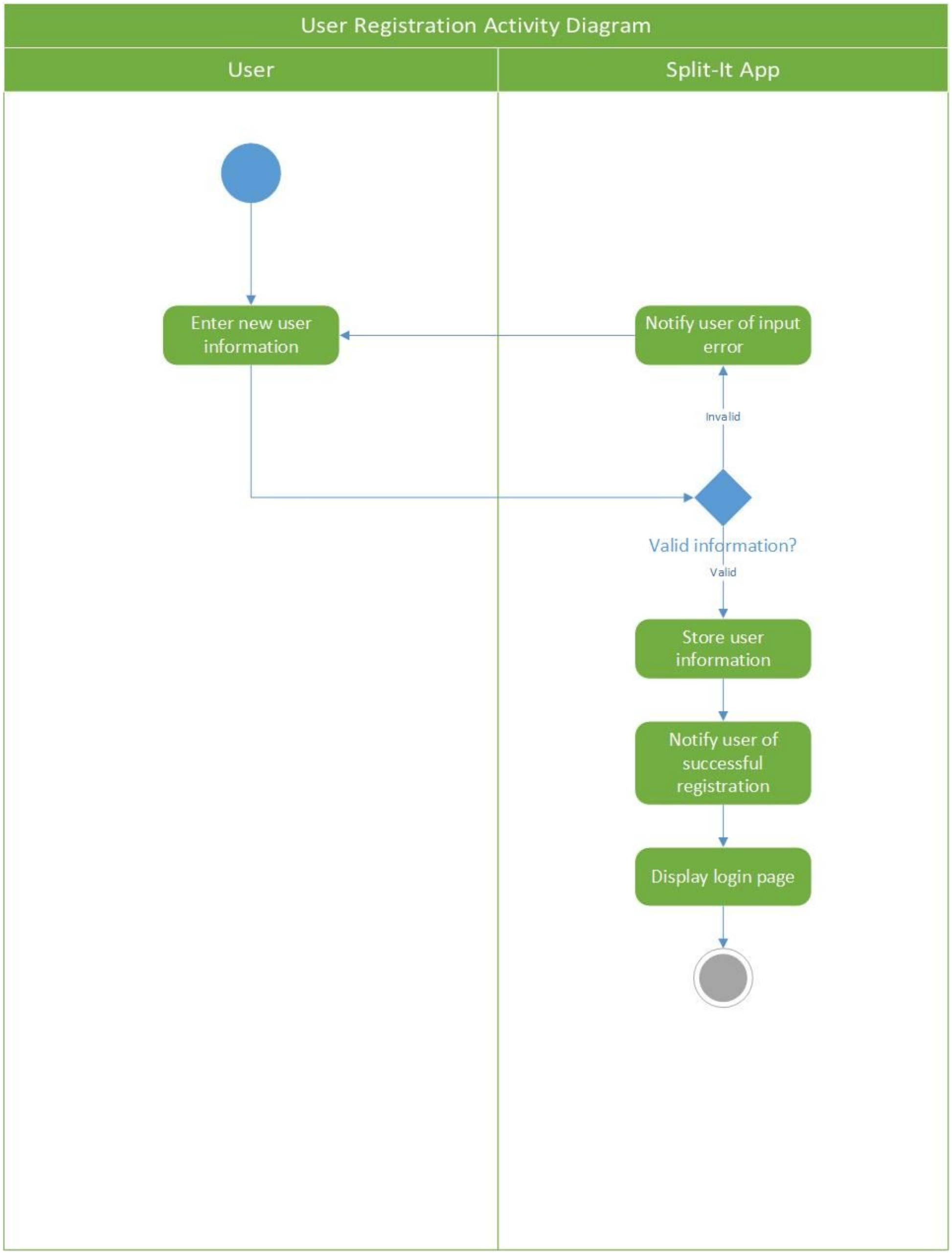


There are three primary objects that compose the application. The most important object is User. This class represents the individual operating the app, storing personal information such as name and Venmo ID as well as their login details. ReceiptCreator, GroupOwner, and GroupMember extend the User class. The first is responsible for creation of the receipt, while the second manages the group as well as the receipts within it. Users are composed into a group which also includes the receipts of that group. Finally the receipts are composed of individual items and can be in the form of an entire group’s receipt, a receipt listing items assigned to a particular user, or a receipt in the archive.

VIII. Activity Diagrams

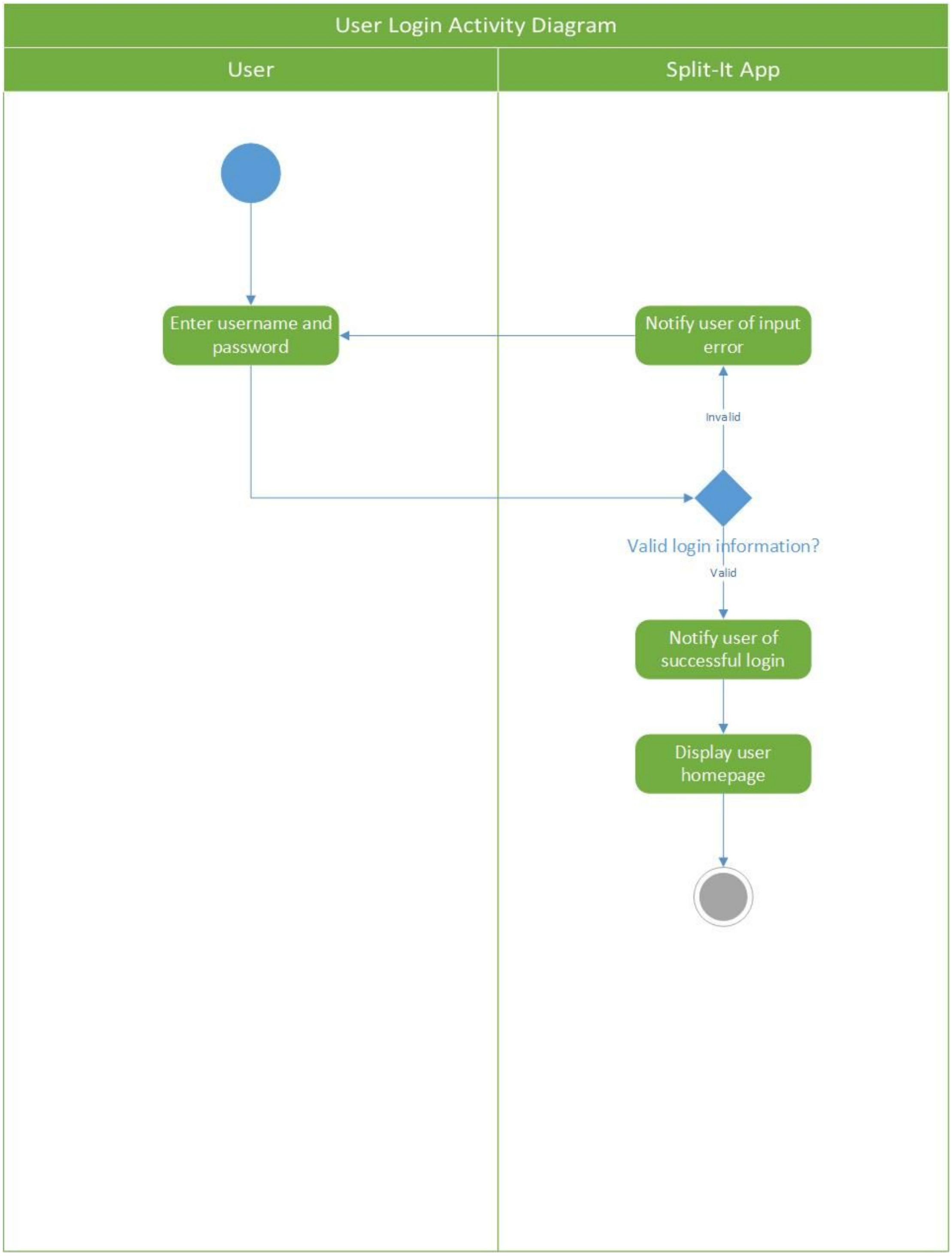
1. User Registration

User registration is the first step necessary for new users to access the app. Entering registration information will approve or deny text based on if it is valid or invalid. Valid text will be stored in the database and the user will be prompted to login.


User Registration Activity Diagram
User
Split-It App
Enter new user information
Notify user of input error
Invalid
Valid information?
Valid
Store user information
Notify user of successful registration
Display login page

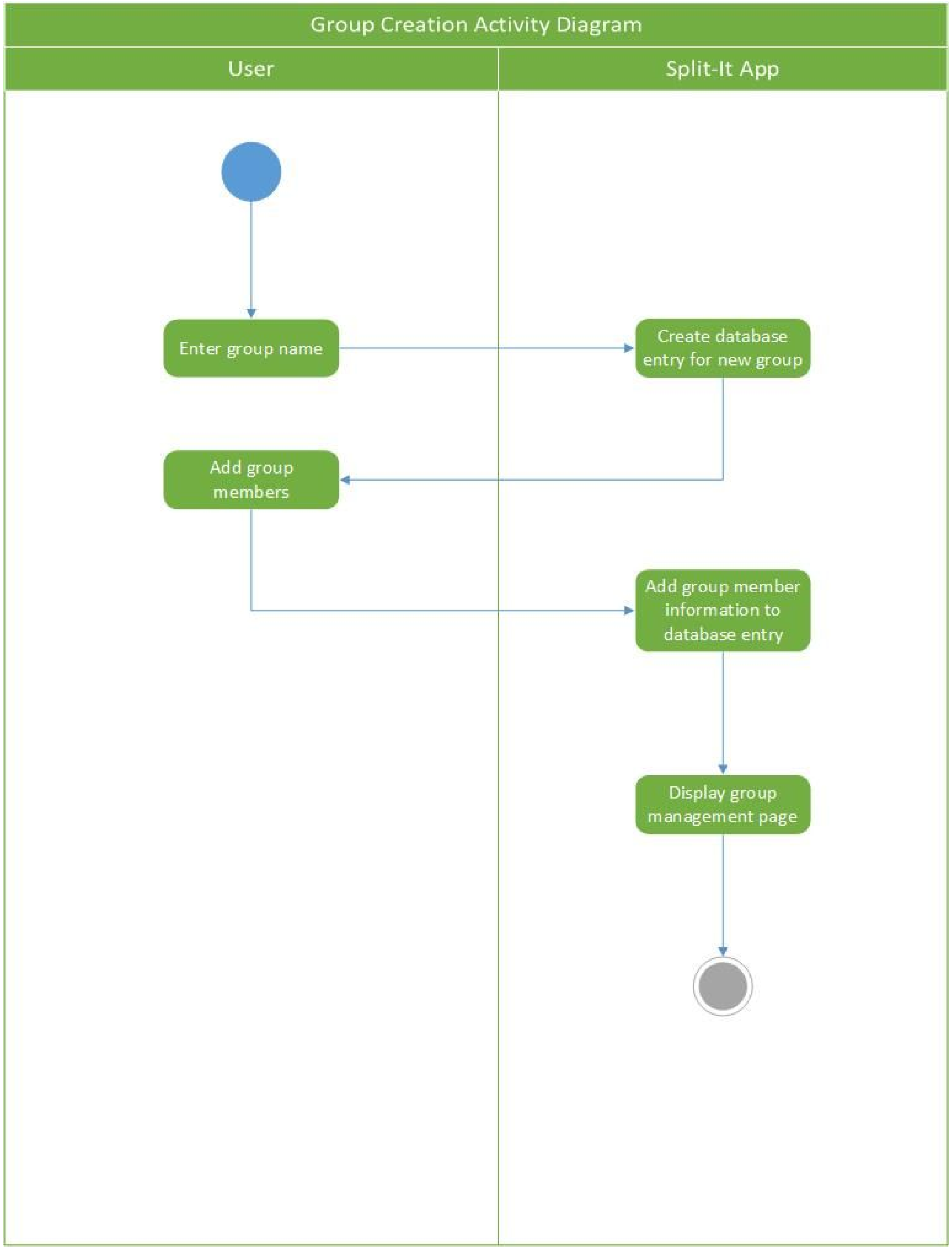
2. User Login

Users will enter their login information and the application will determine if it is valid or invalid using information stored in the application database. Upon entry of valid information, the user will be redirected to the homepage.


User Login Activity Diagram
User
Split-It App
Enter username and password
Notify user of input error
Invalid
Valid login information?
Valid
Notify user of successful login
Display user homepage

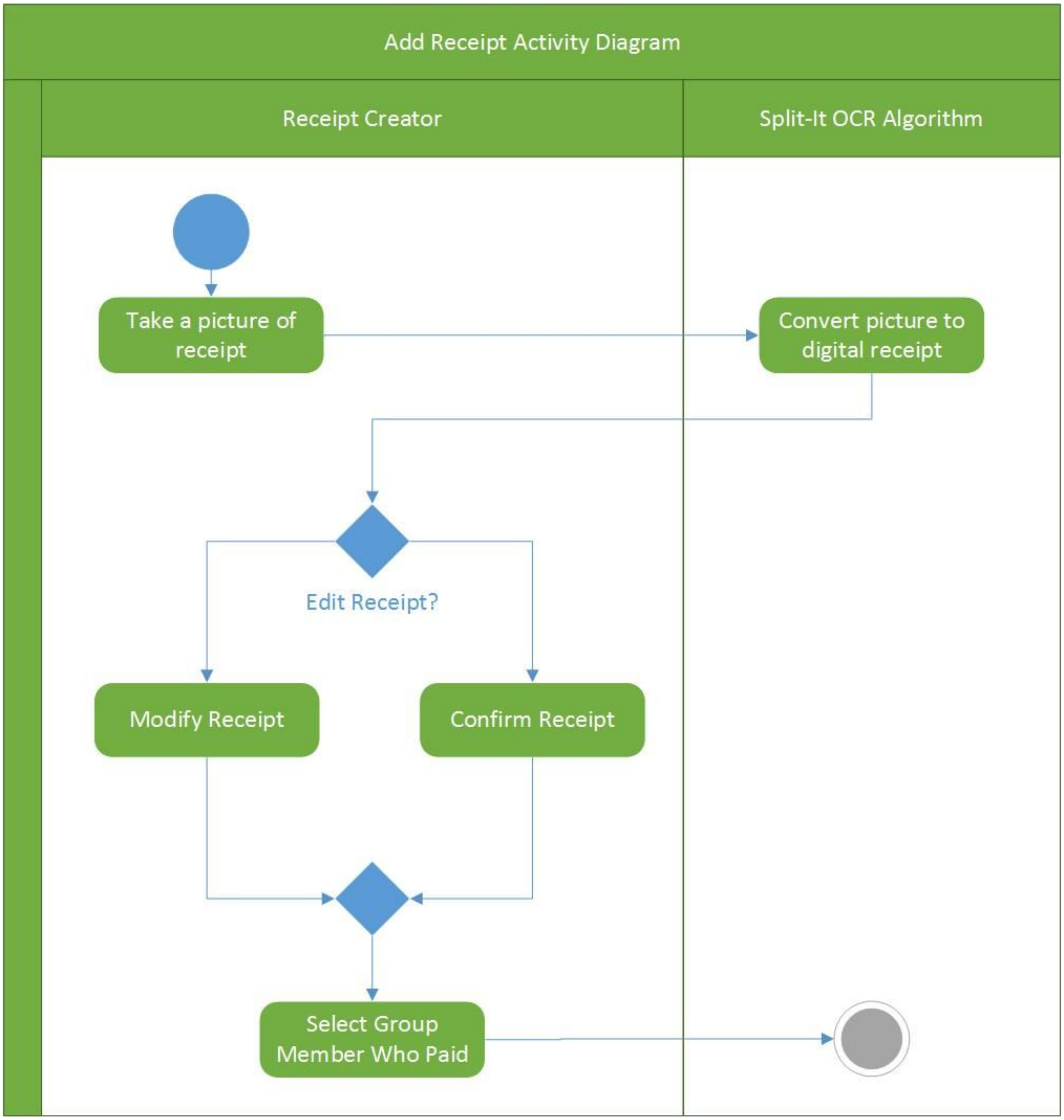
3. Group Creation

Group creation will allow users to manage a payment group. Once a group is created, group info is added to the database and the group management page is display to the user.


Group Creation Activity Diagram
User
Split-It App
Enter group name
Create database entry for new group
Add group members
Add group member information to database entry
Display group management page

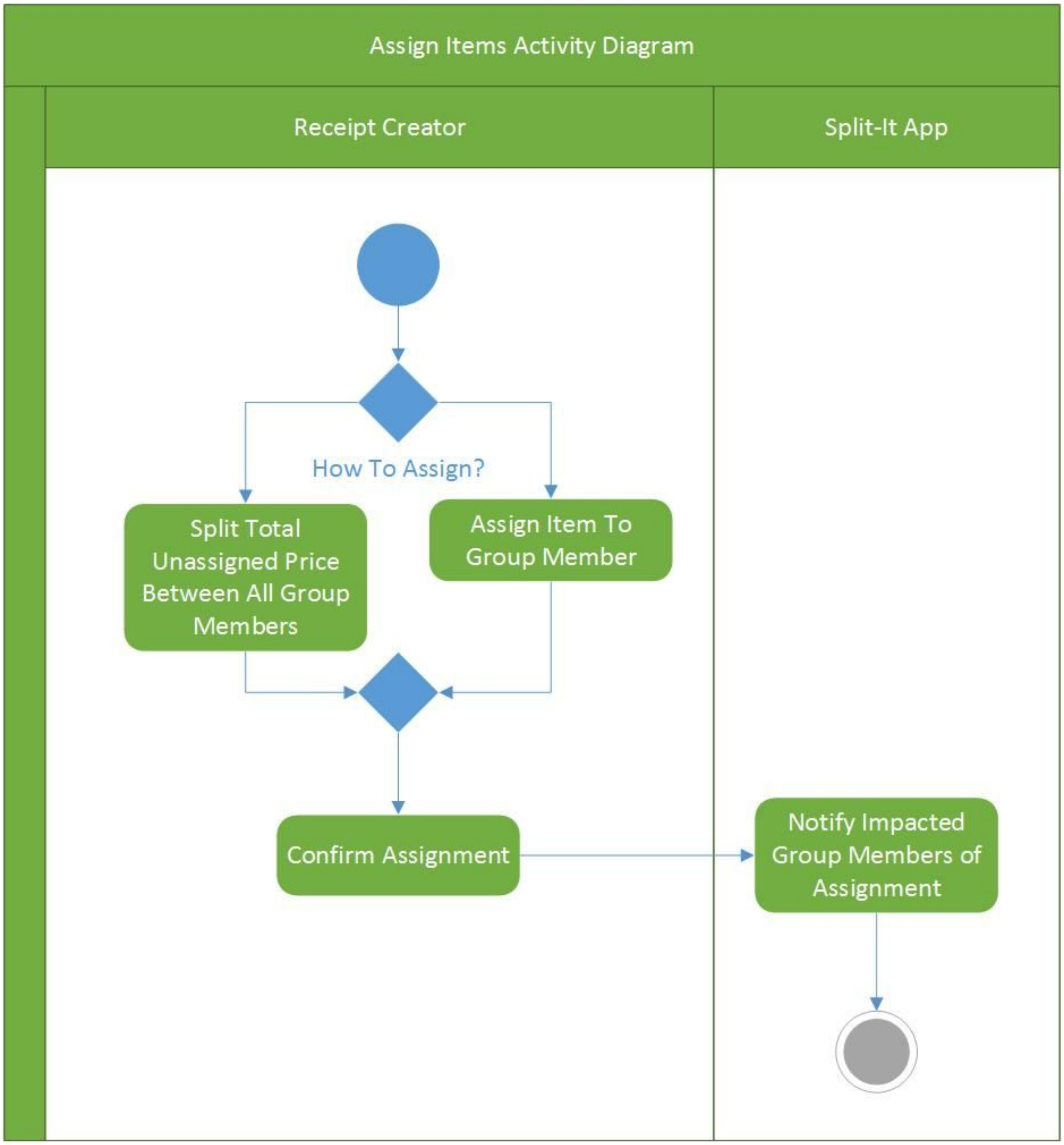
4. Adding Receipts

This activity diagram represents the process of adding a receipt to the Split-It mobile app. The user must first use the app to take a picture of a receipt. Then, the app will automatically convert the receipt into a digitized, text, and editable format via optical character recognition. At this point, the user may choose to modify or confirm the receipt and select the group member you paid for the items on the receipt.


Add Receipt Activity Diagram
Receipt Creator
Split-It OCR Algorithm
Take a picture of receipt
Convert picture to digital receipt
Edit Receipt?
Modify Receipt
Confirm Receipt
Select Group Member Who Paid

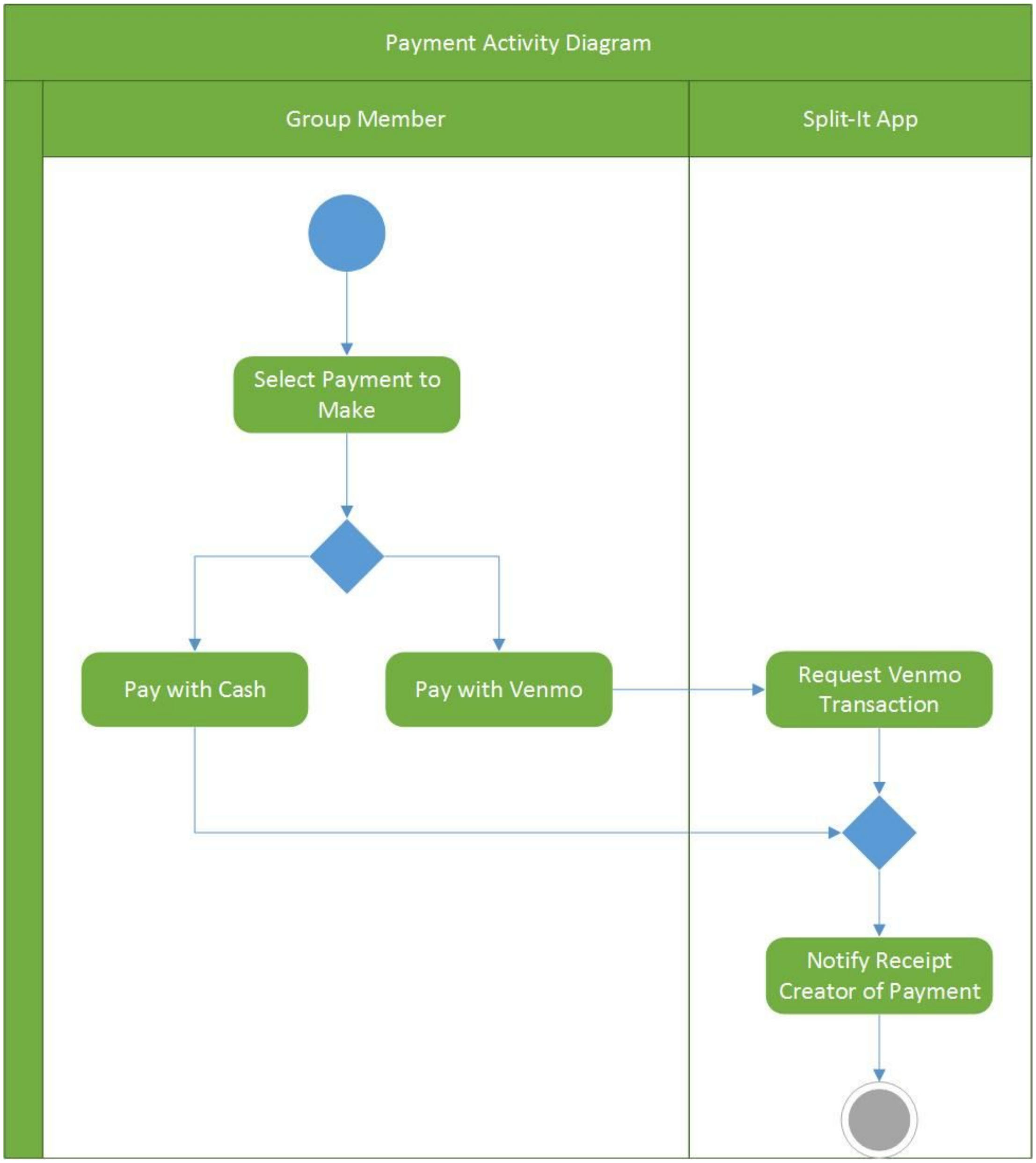
5. Assigning Items

This activity diagram represents the process of assigning items from a scanned receipt in Split-It to members who are responsible for paying for the items. The receipt creator must do this by either choosing to have the total amount of the receipt split between all users in the group or assigning each item to individual members in the group who are responsible for paying for the assigned items. The receipt creator then confirms the assignment and the impacted group members are notified.


Assign Items Activity Diagram
Receipt Creator
Split-It App
How To Assign?
Split Total Unassigned Price Between All Group Members
Assign Item To Group Member
Confirm Assignment
Notify Impacted Group Members of Assignment

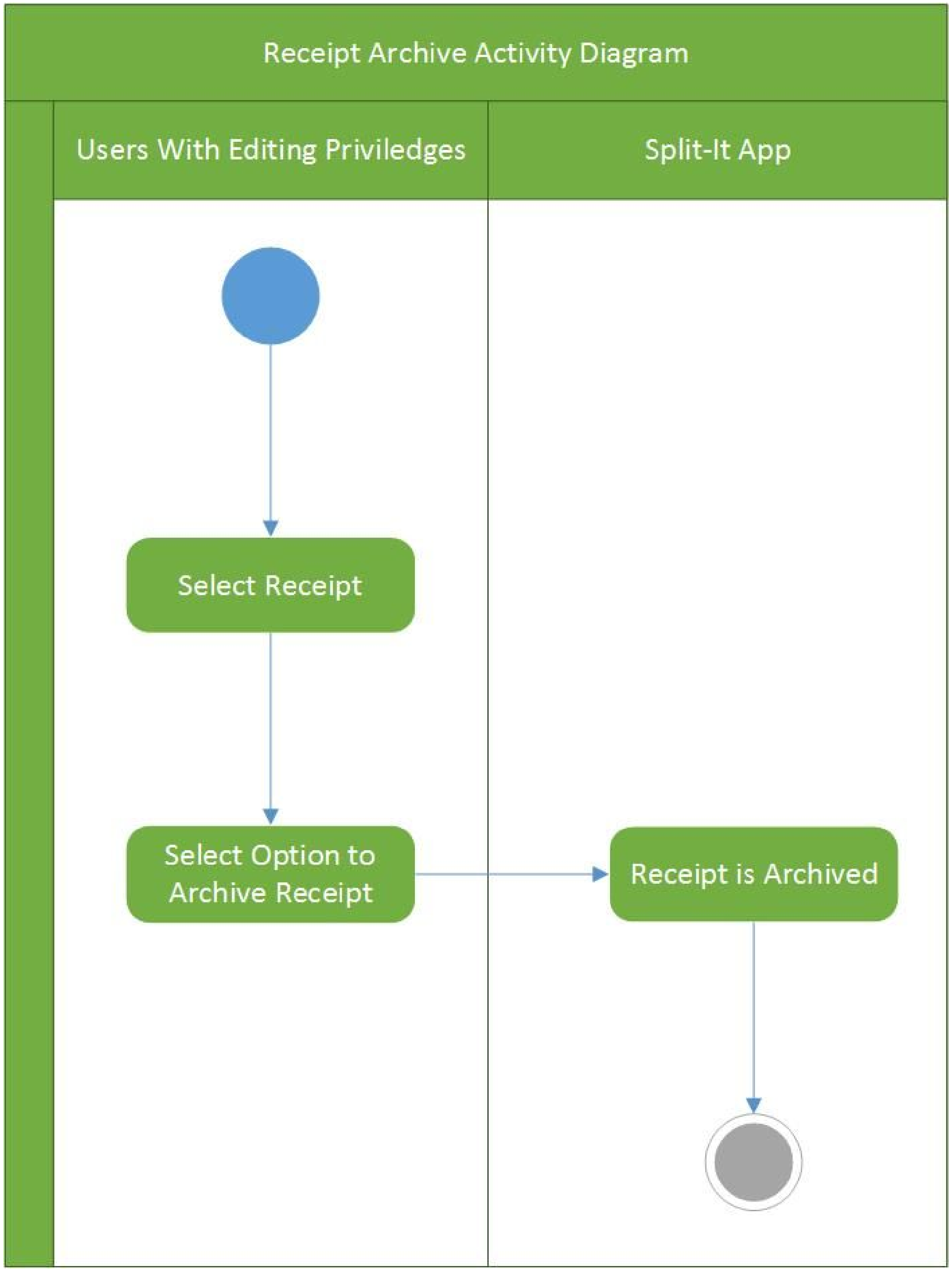
6. Payment

This activity diagram represents the process of group members paying for items from receipts. Group members must first choose which item they would like to pay for from their list of due payments by selecting the payment to make. Then they may choose to pay with cash or Venmo; if Venmo is chosen, then Venmo is opened, else the payment is done outside of the app. Then, once complete, the receipt creator is notified of the payment.



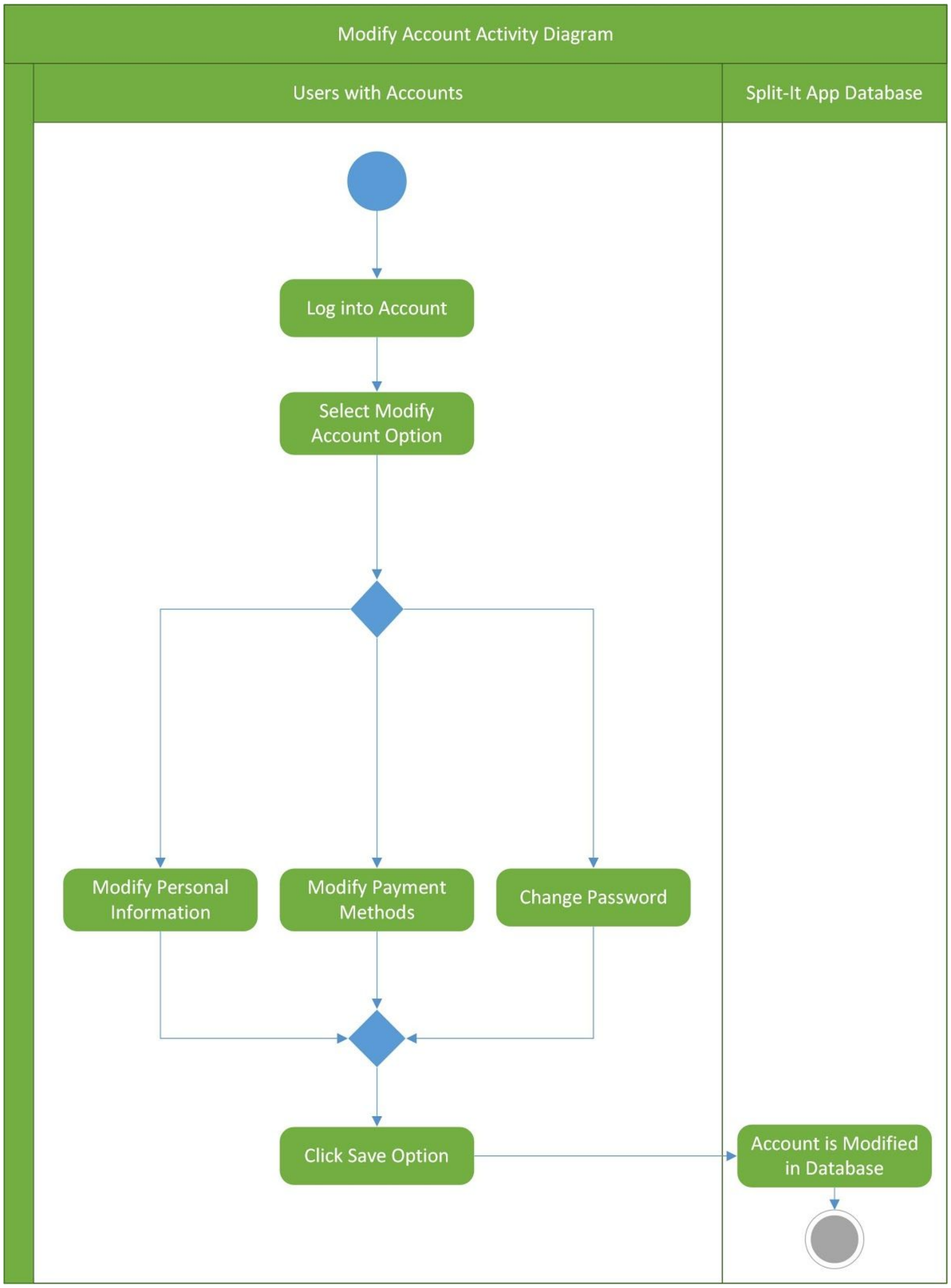
7. Archiving Receipts

This activity diagram represents the process of archiving a receipt so it can be viewed later. Users with editing privileges may select a receipt on their account, click the archive option, and the receipt will be archived in the Split-It app. Archived receipts cannot be modified.


Receipt Archive Activity Diagram
Users With Editing Priviledges
Split-It App
Select Receipt
Select Option to Archive Receipt
Receipt is Archived

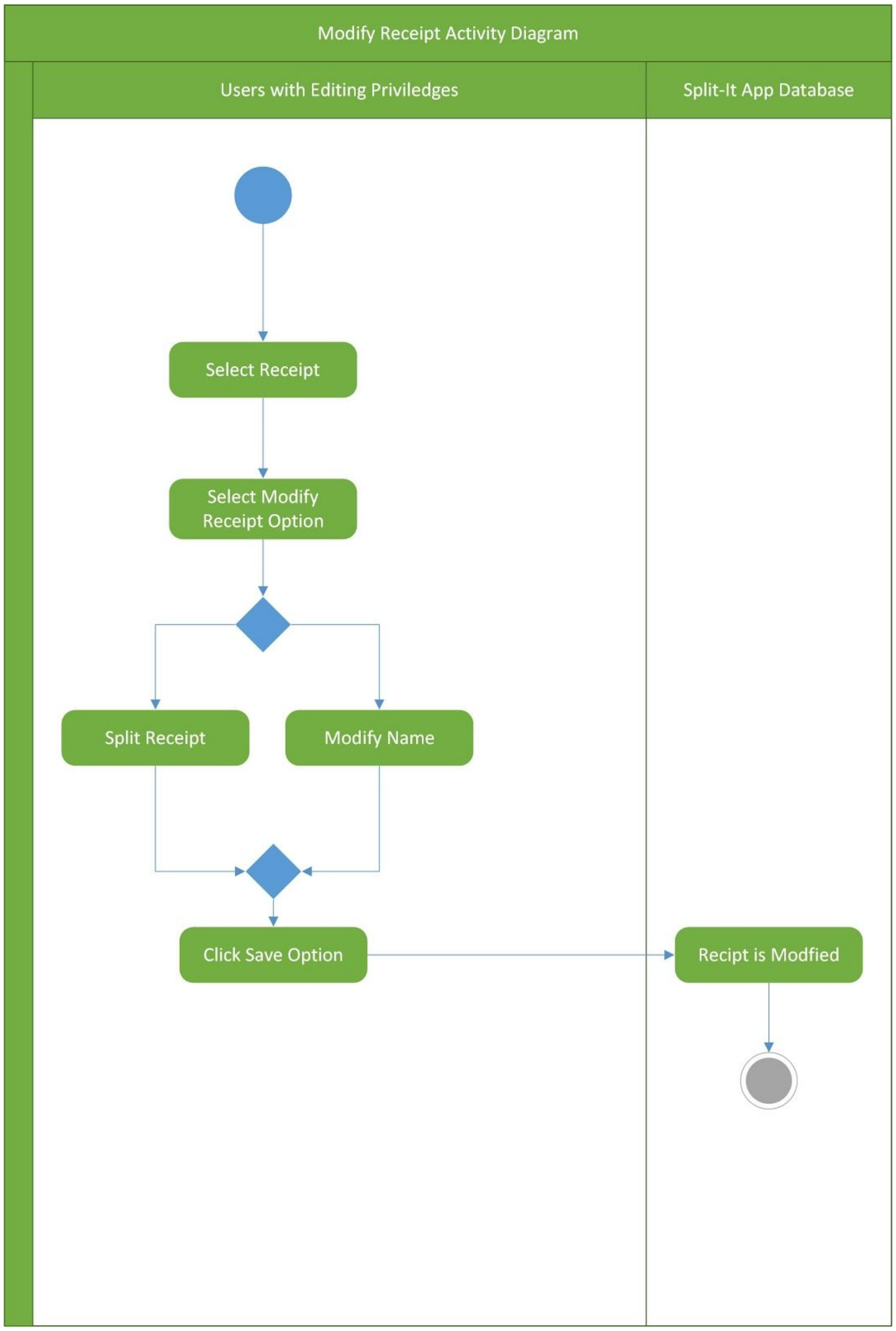
8. Modifying Accounts

This activity requires users to log in their accounts. After logging into their accounts and choosing to modify the account, users can choose to modify personal information, modify payment methods, change password, etc. The user can modify any of the settings in the account at leisure. After the user chooses to save the modification, the changes will be saved in the Split-It Database.


Modify Account Activity Diagram
Users with Accounts
Split-It App Database
Log into Account
Select Modify Account Option
Modify Personal Information
Modify Payment Methods
Change Password
Click Save Option
Account is Modified in Database

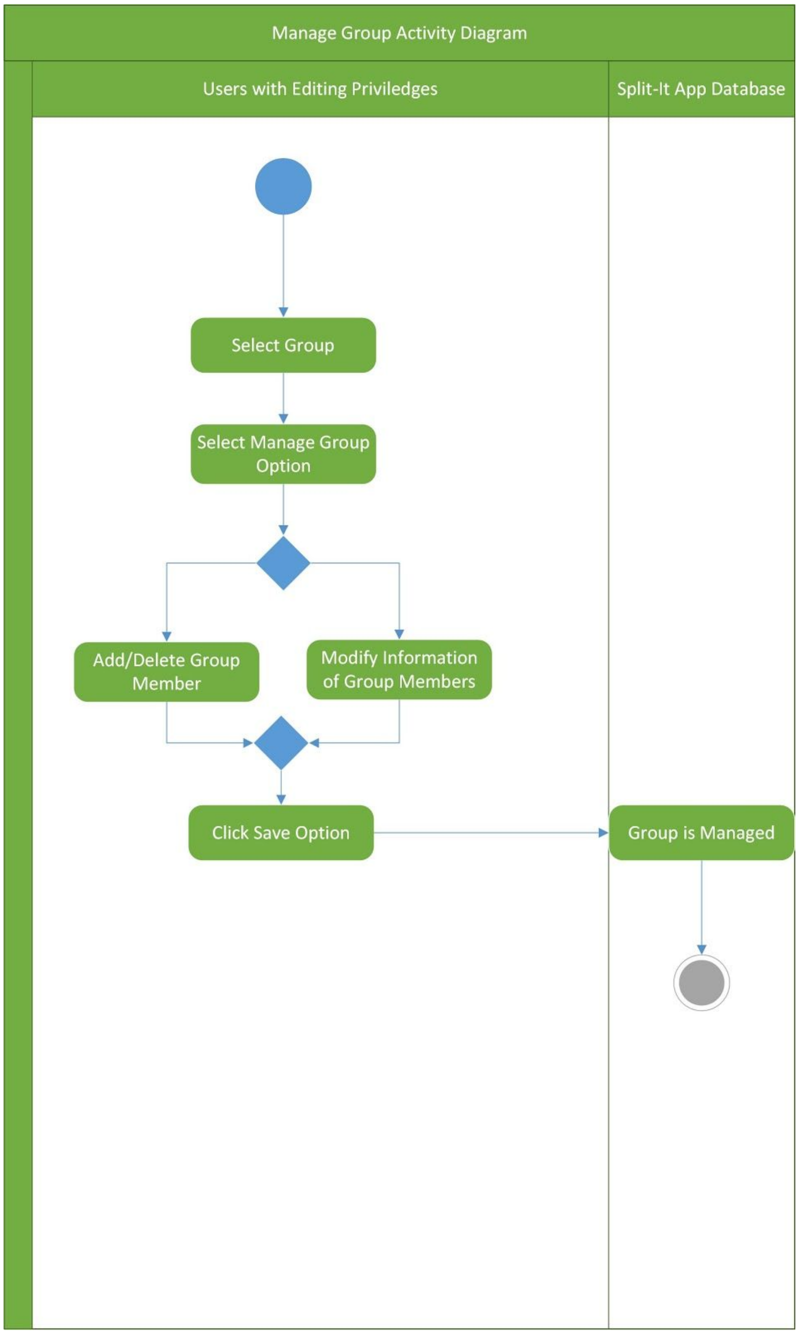
9. Modifying Receipts

Modifying a receipt displays and changes the attributes associated with the receipt. After selecting a specific receipt, the user can choose to split the receipt, modify names, etc. The user can modify any receipt at leisure. After the user chooses to save the modification, the changes will be saved in the Split-It Database.



10. Managing Groups

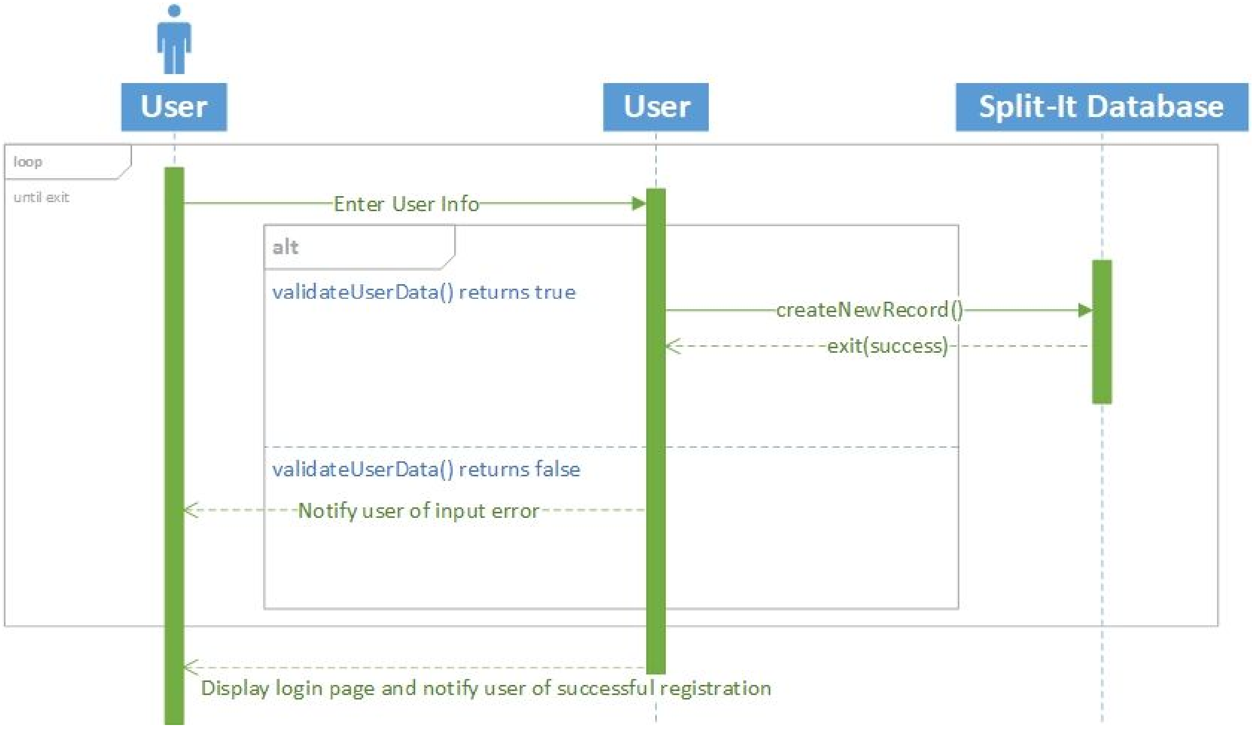
Managing a group displays and changes the attributes associated with the group. After selecting a specific group, the user can choose to add/delete a group member, manage group members information, etc. The user can manage any group at leisure. After the user chooses to save the changes, the changes will be saved in the Split-It Database.


Manage Group Activity Diagram
Users with Editing Priviledges
Split-It App Database
Select Group
Select Manage Group Option
Add/Delete Group Member
Modify Information of Group Members
Click Save Option
Group is Managed

IX. Sequence Diagrams

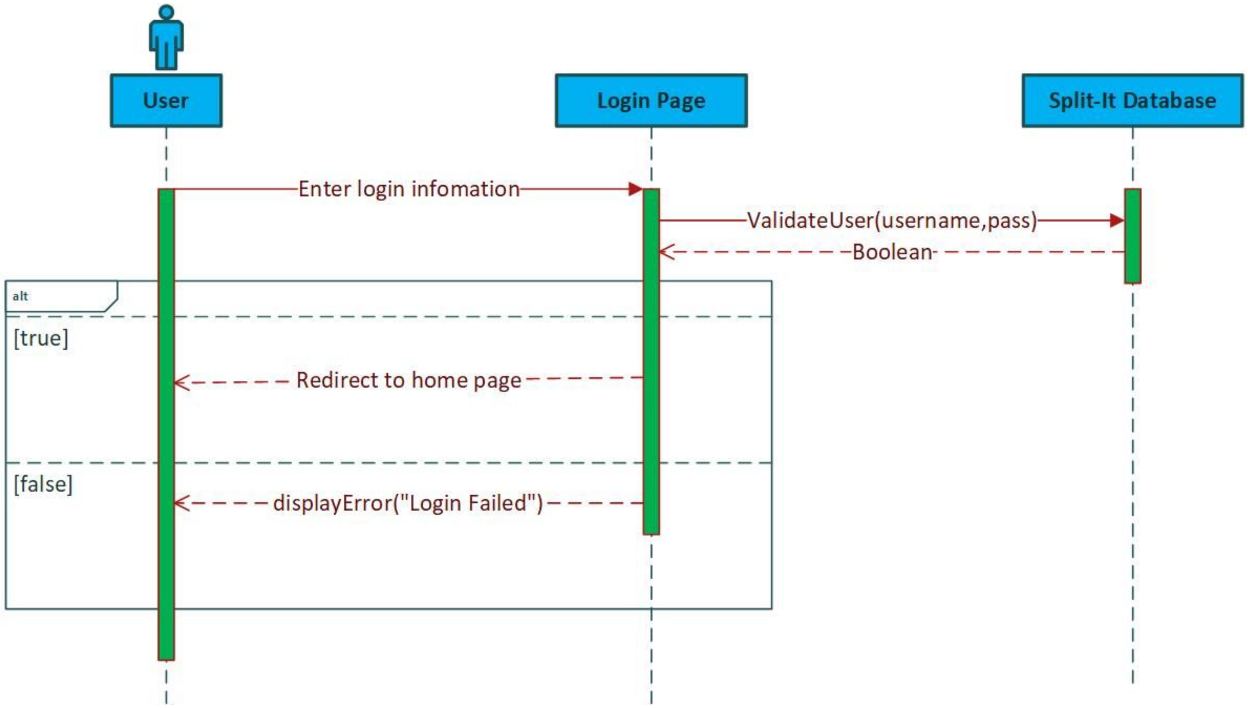
1. User Registration

When a user opens our app for the first time they will be able to enter their name and create a username and password, which will be used to login to the app in the future. Once the user has entered valid information they will be taken to the login page where they can use their new credentials to login, and the information they provided will be stored in the database. If the information that the user provides is invalid an appropriate error message will be displayed.


User
User
Split-It Database
loop
until exit
Enter User Info
alt
validateUserData() returns true
createNewRecord()
exit(success)
validateUserData() returns false
Notify user of input error
Display login page and notify user of successful registration

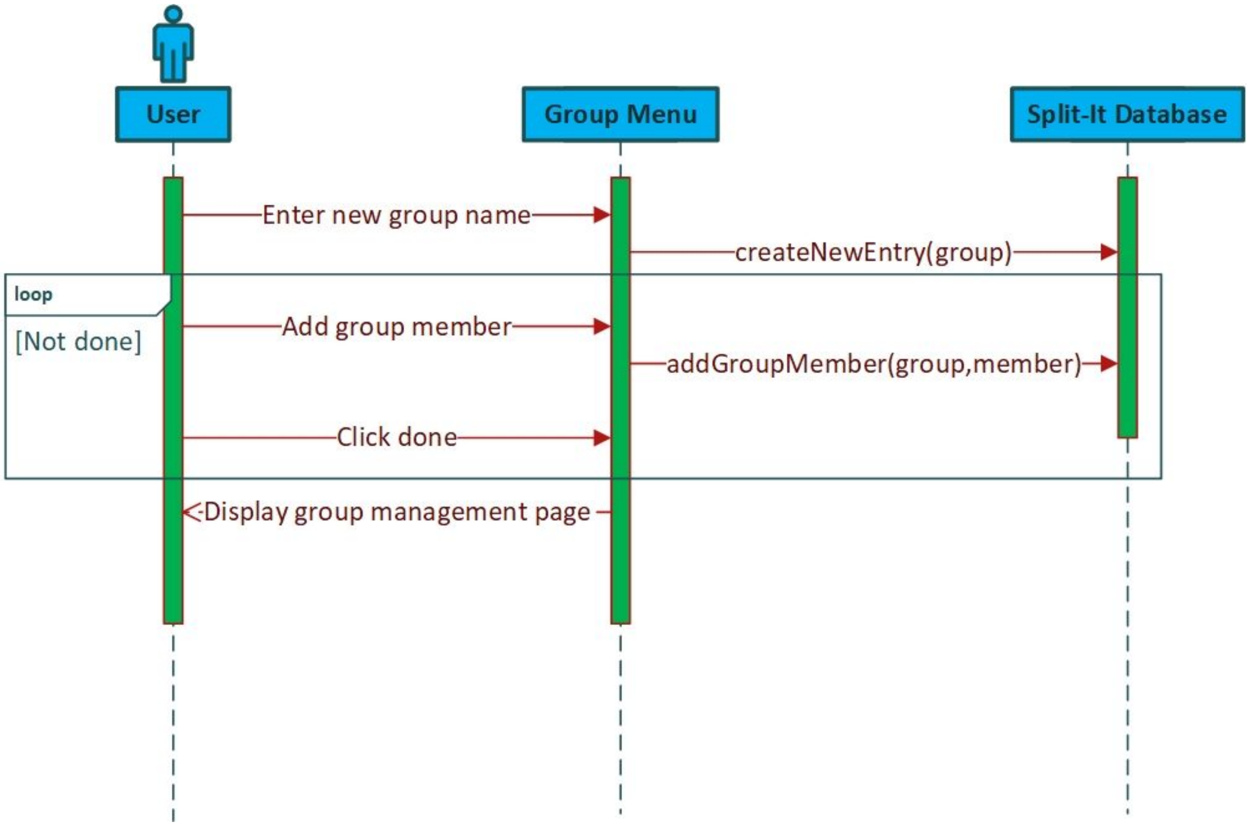
2. User Login

User login will call to validateUser() after information is entered by the user. Valid information will result in the user being directed to the homepage and invalid information will display an error.



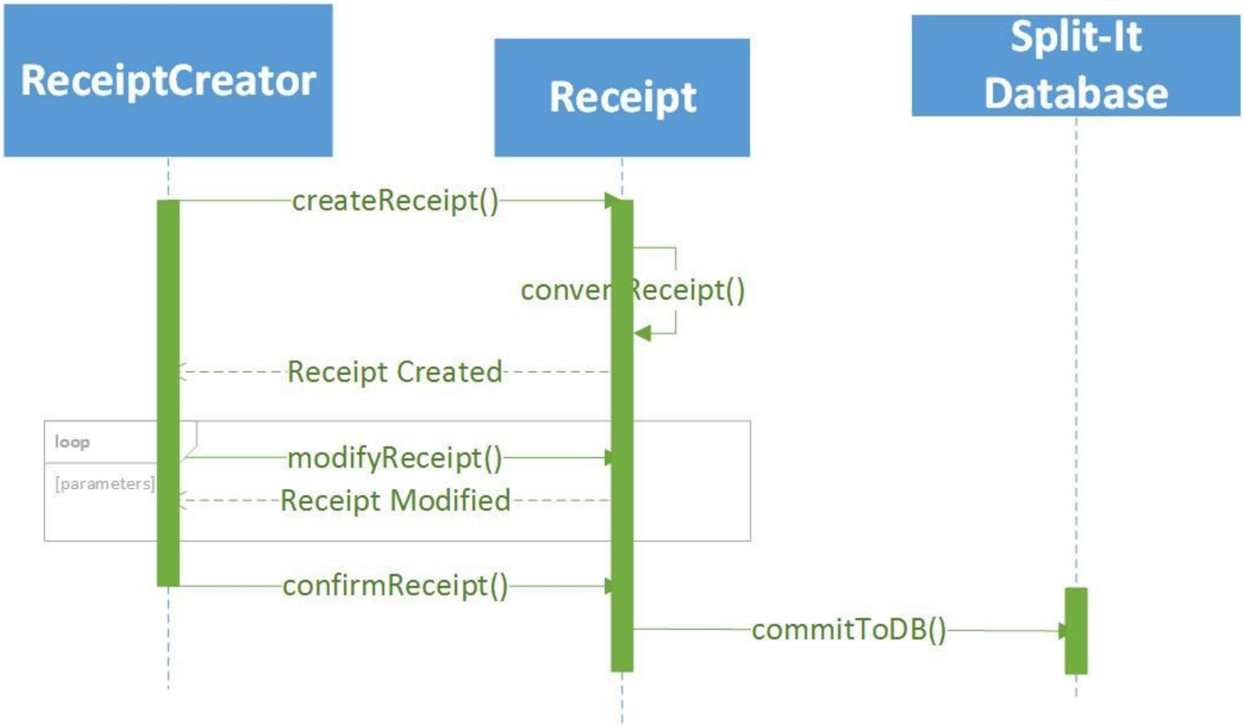
3. Group Creation

Group creation will call createNewEntry() in database and store the group name. New users can be added to the group once it is created. Adding new group members will call addGroupMember() which will link group members to a group name in the database. After group creation is complete, the app will display the group management page.



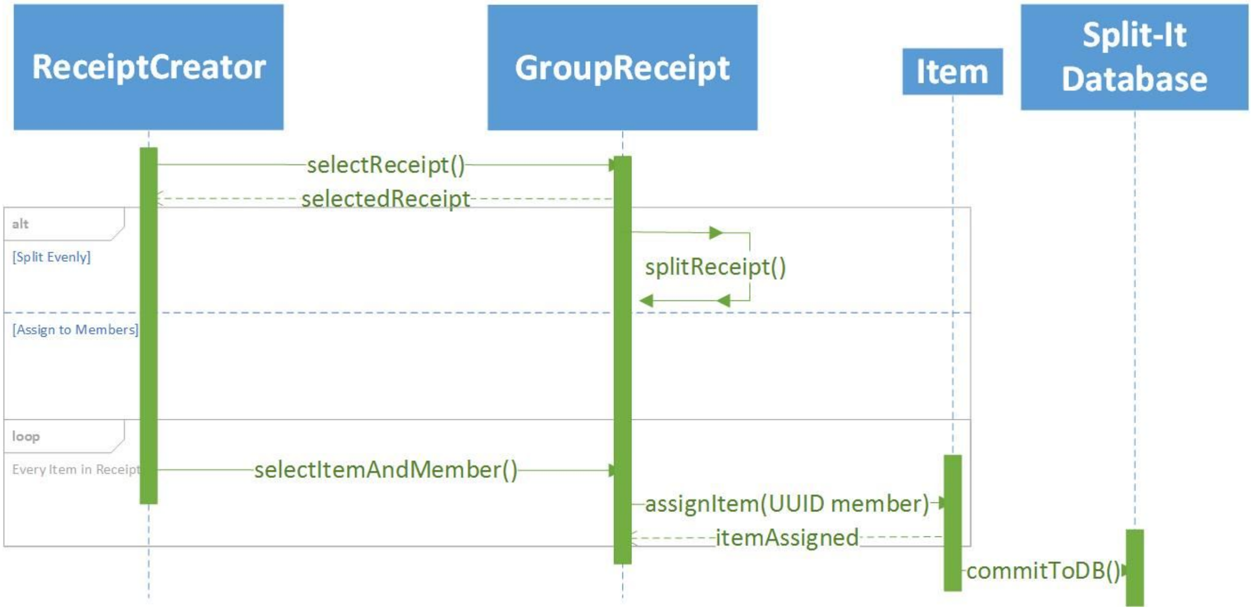
4. Add Receipt

The Receipt Creator may use the createReceipt() function to scan a document/receipt and it will be converted into a digital receipt using the BlinkReceipt API. After the receipt is created, the creator may continuously modify the receipt as needed, and then may confirm the receipt, at which point the receipt is added to the database.



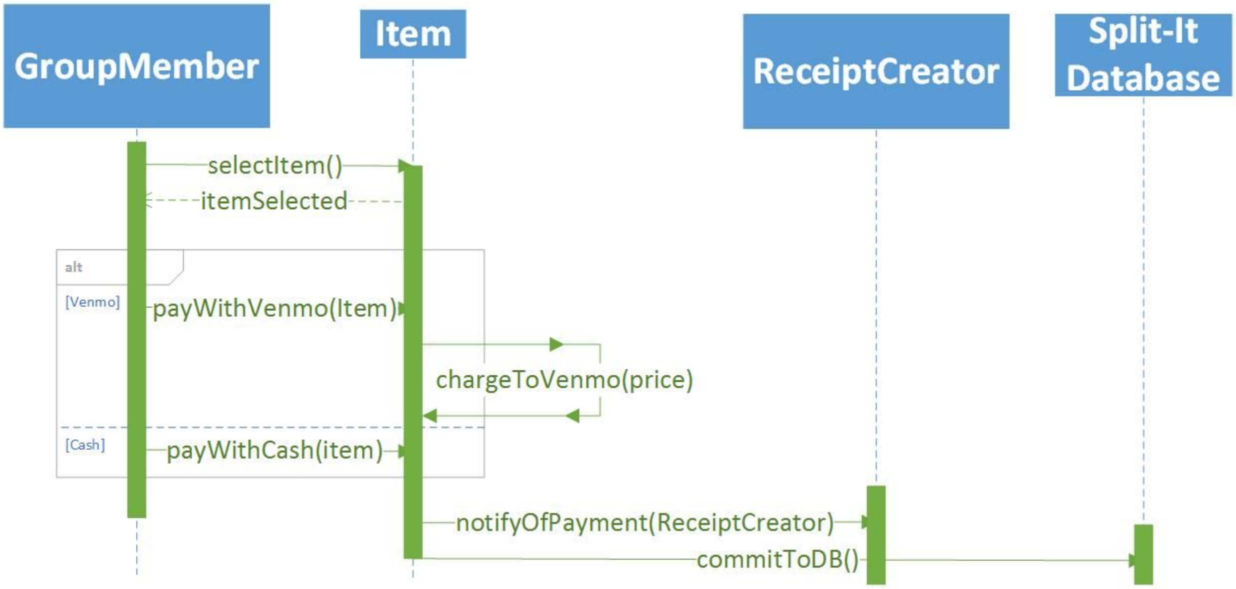
5. Assign Items

The Receipt Creator may select a receipt he/she has created and then can choose to either split the receipt evenly or assign items to specific group members. The changes are all recorded and committed to the database.



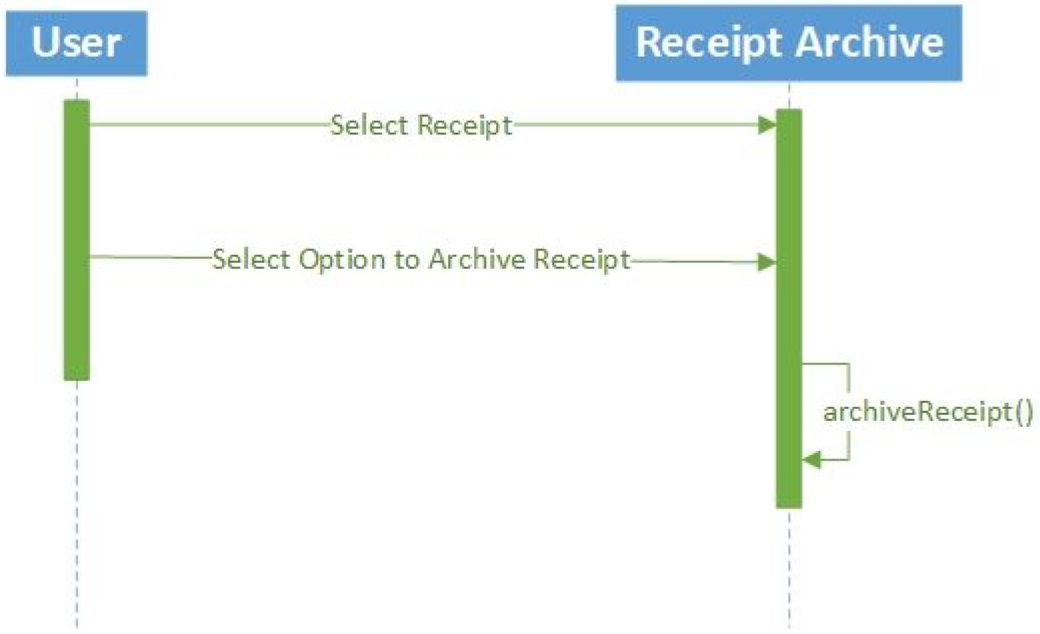
6. Payment

Each group member may select items they have not yet paid. They can then choose to either pay with Venmo, in which case Venmo is opened, or they may pay with cash, which happens outside of the app. The receipt creator is then notified of payment. Each payment is recorded in the database as the last step.



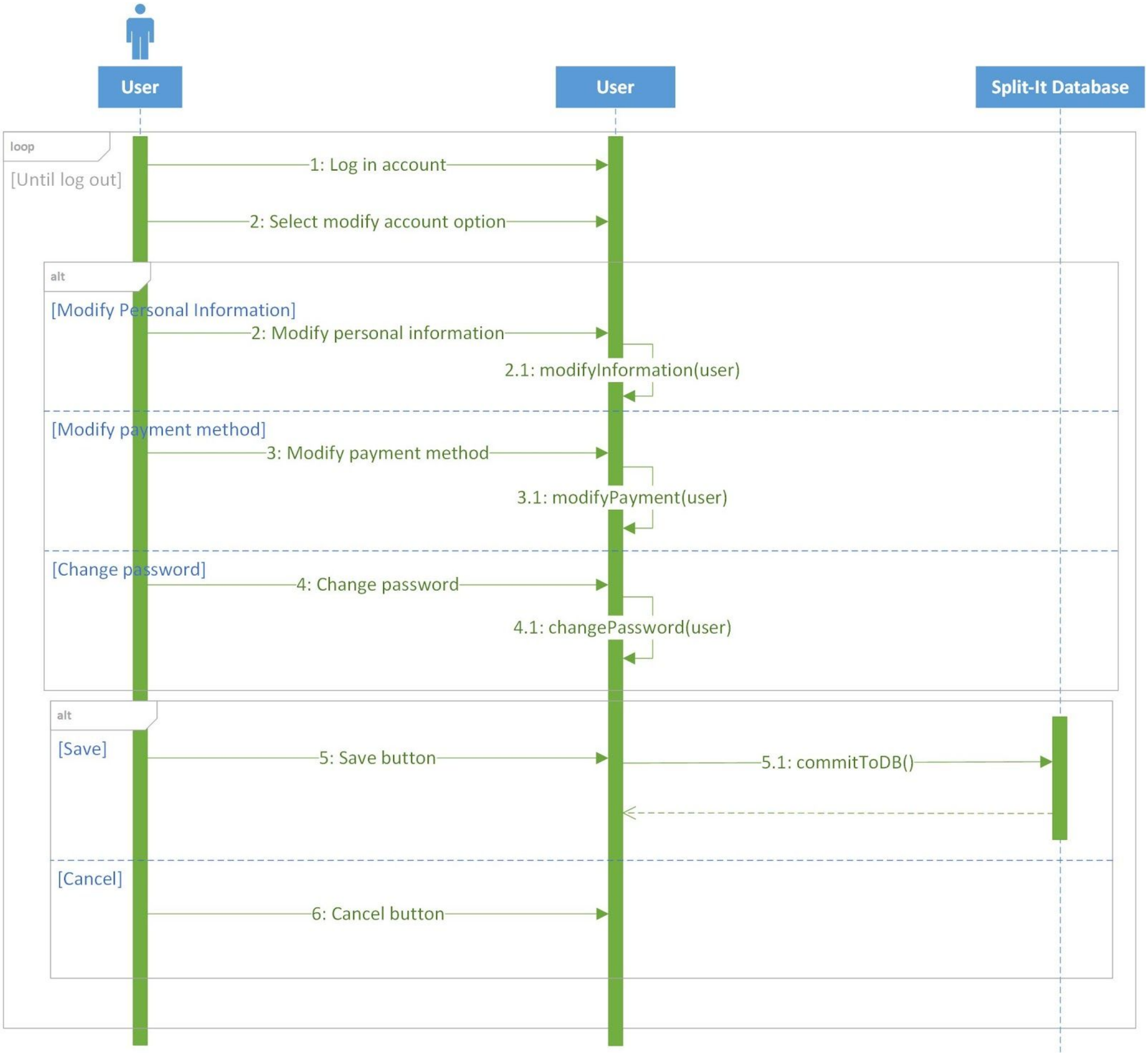
7. Receipt Archive

The creator of a receipt has the option to archive a receipt, which allows them to clear out old receipts that do not need to be modified anymore. In order to archive a receipt the receipt creator will select the receipt, and then select archive receipt from the options presented. Once the user does this the receipt will be archived, and they will not be able to edit it in the future.



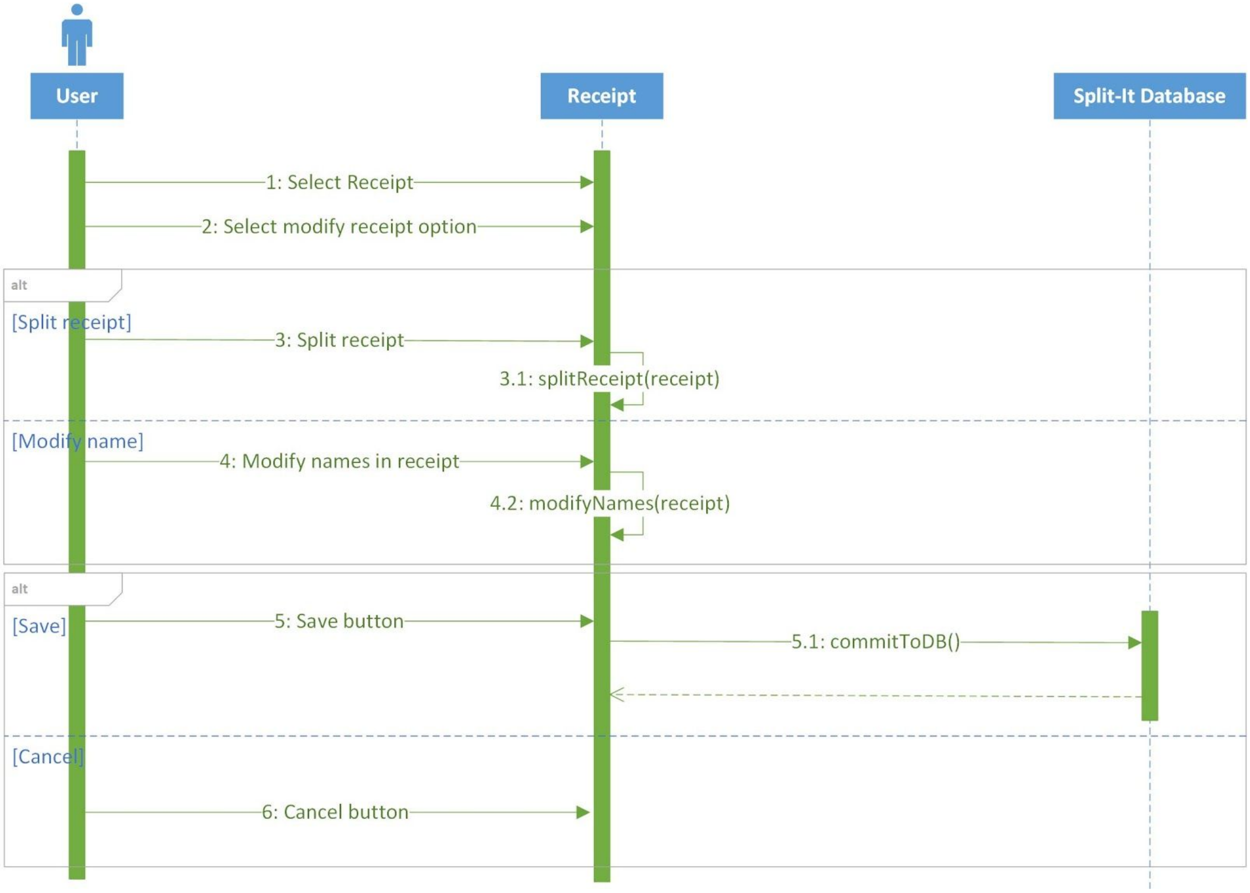
8. Modify an account

To modify an account's information, the user needs to log into their account, which opens a User class. This displays three different options that are: modify personal information, modify payment method, and change password. When the user is ready to save the changes, he or she presses the save button, which updates the account and saves updates to the Split-It Database. Alternately, the user may press cancel to ignore changes to the account's information.



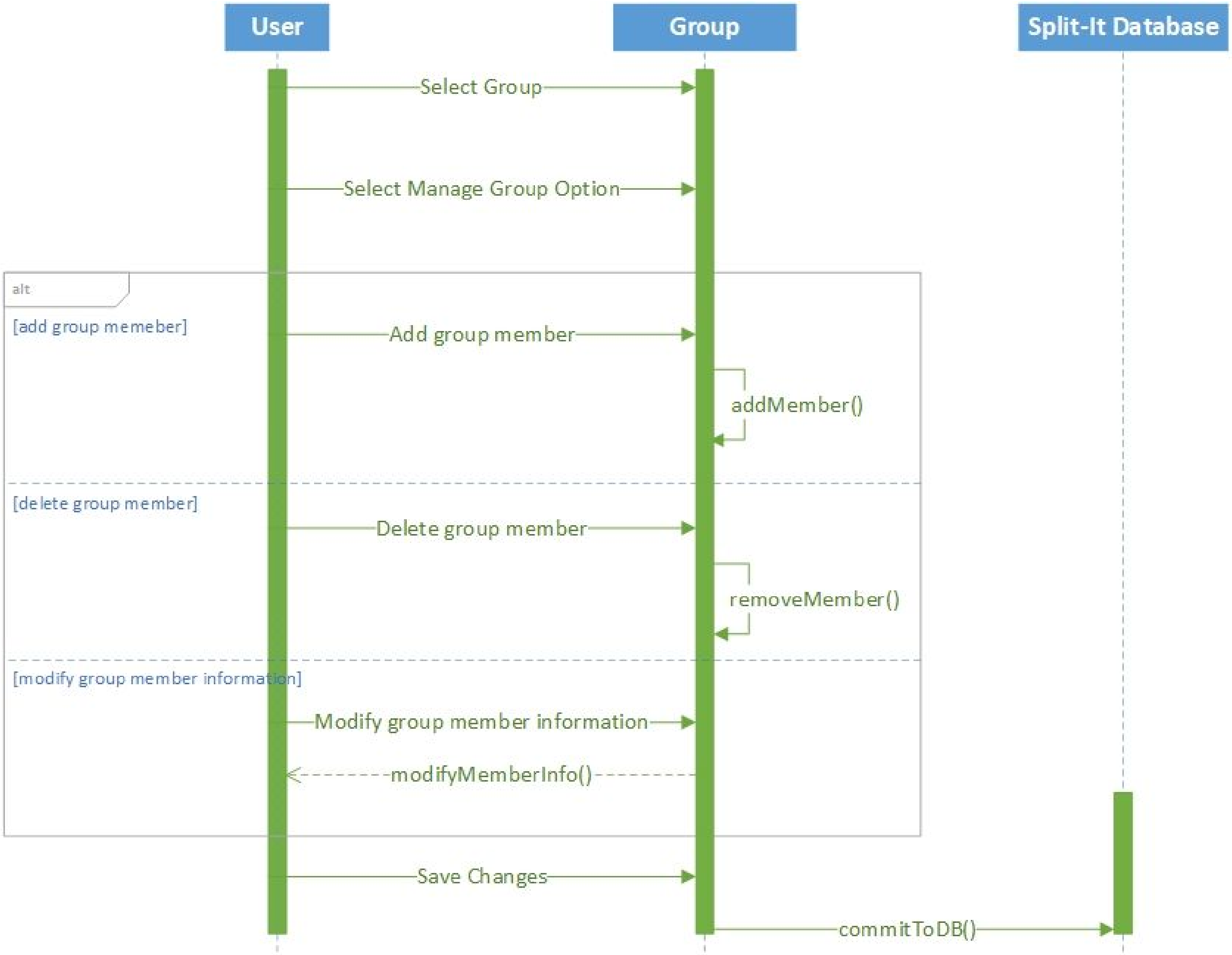
9. Modify a receipt

To modify a receipt, the user taps the receipt, which opens a Receipt class. This displays two different options: split receipt and modify names in receipt. When the user is ready to save the changes, he or she presses the save button, which updates the receipt and saves updates to the Split-It Database. Alternately, the user may press cancel to ignore changes to the receipt.



10. Manage Group

To manage a group the group owner must first select the group and then choose the manage group option from a list of options presented to them. They will then have the option to add a group member, delete a group member, or modify a group members information. All of the changes will be saved to the database, and all everyone else will be able to see these changes the next time they open the app.


User
Group
Split-It Database
Select Group
Select Manage Group Option
alt
[add group memeber]
Add group member
addMember()
[delete group member]
Delete group member
removeMember()
[modify group member information]
Modify group member information
modifyMemberInfo()
Save Changes
commitToDB()