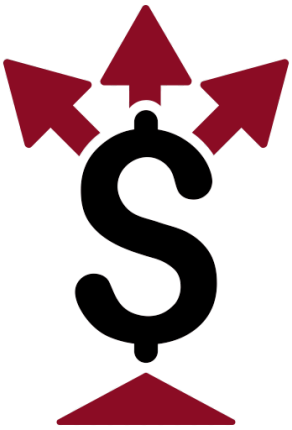


Split-It

Software Requirements and Design Specifications

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

March 1, 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

Table of Contents

Version History **2**

Table of Contents **3**

I. INTRODUCTION **4**

Purpose 4

Scope 4

Goals 4

Glossary 5

II. PROJECT DESCRIPTION **5**

User Account Creation 5

User Account Management 5

Group Creation 6

Group Management 6

Receipt Creation 6

Receipt Management 6

Price Assignment 6

User Payment 6

Receipt Archive 7

III. FUNCTIONAL REQUIREMENTS **7**

IV. NONFUNCTIONAL REQUIREMENTS **8**

V. FUNCTIONALITY COMPARISON **9**

VI. Use Case Diagram **11**

VII. Class Diagram **12**

VIII. Activity Diagrams **13**

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

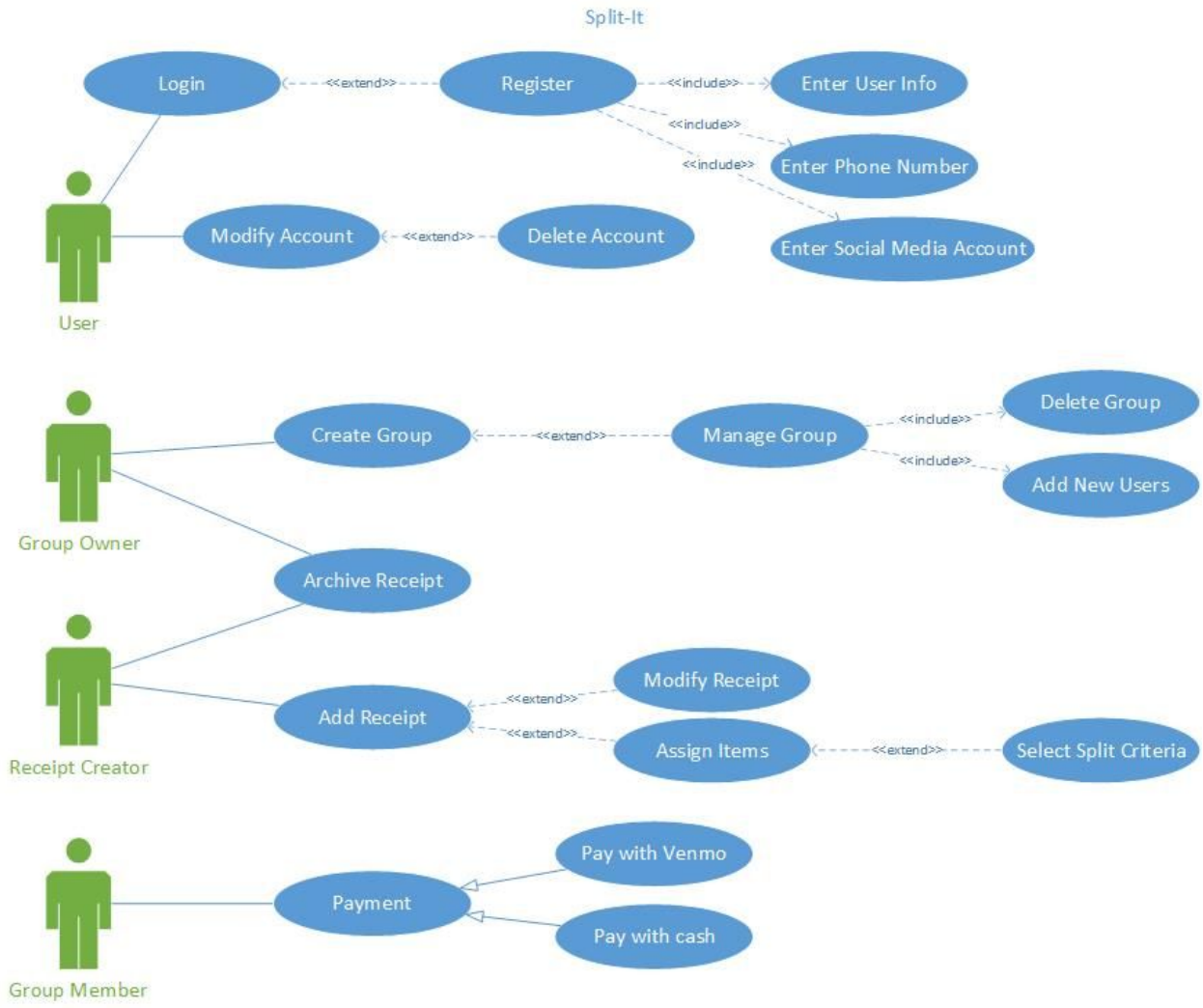
V. FUNCTIONALITY COMPARISON

	Scan Receipt Using OCR	**Create Digital Receipt**	**Create/Manage Consumer Groups**	**Split Costs**	**Initiate Payment**
Splitwise	No	Yes, not	Yes	Yes	No

		itemized.			
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

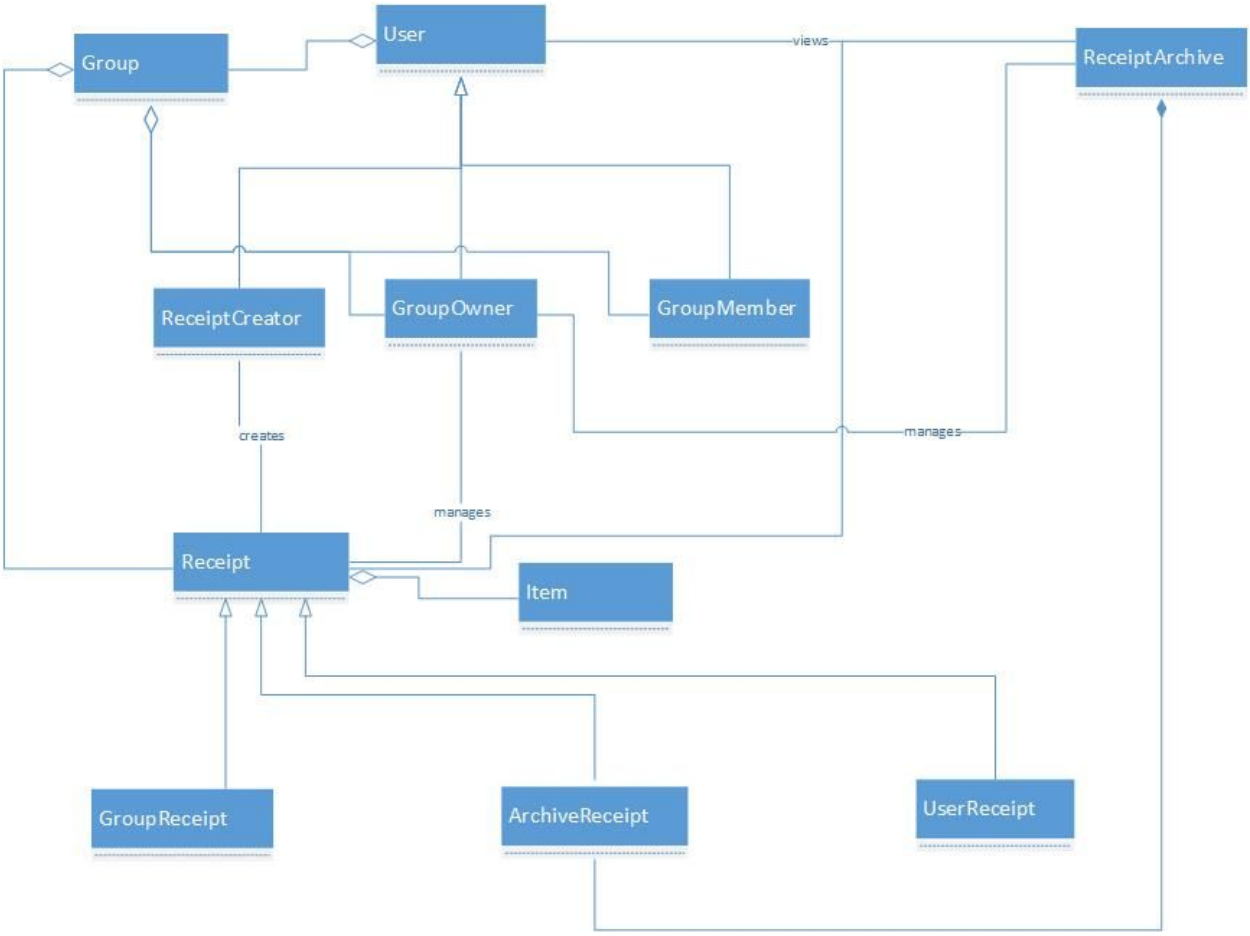
Of all the apps above Splitwise is most comparable to what we are trying to implement. While it does not provide OCR capability, it allows users to manually enter prices for items, and choose from an array of options regarding how they want to split the cost between a group of people. Each person can 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



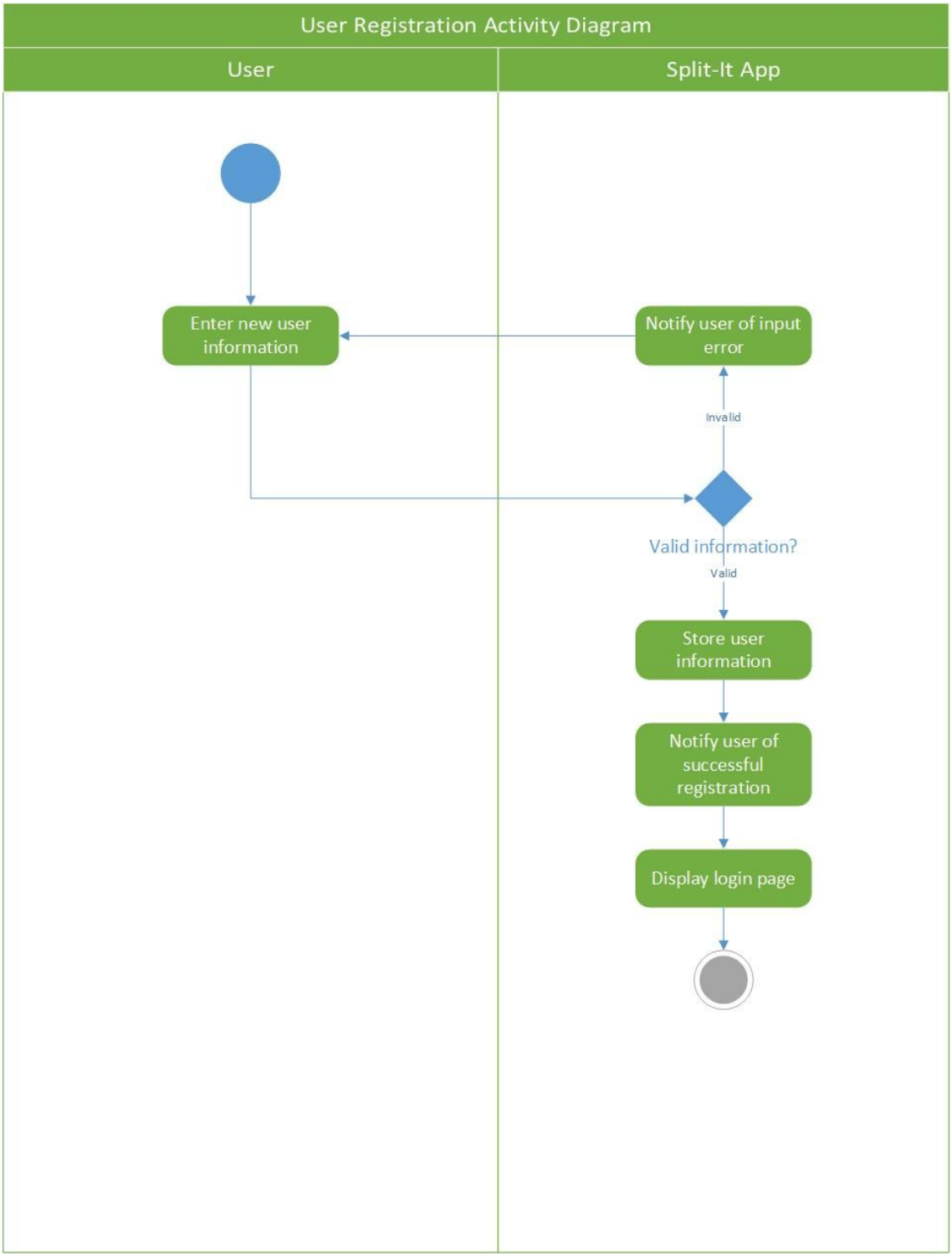
This use case diagram shows the typical interactions for each of the different user types, or states.

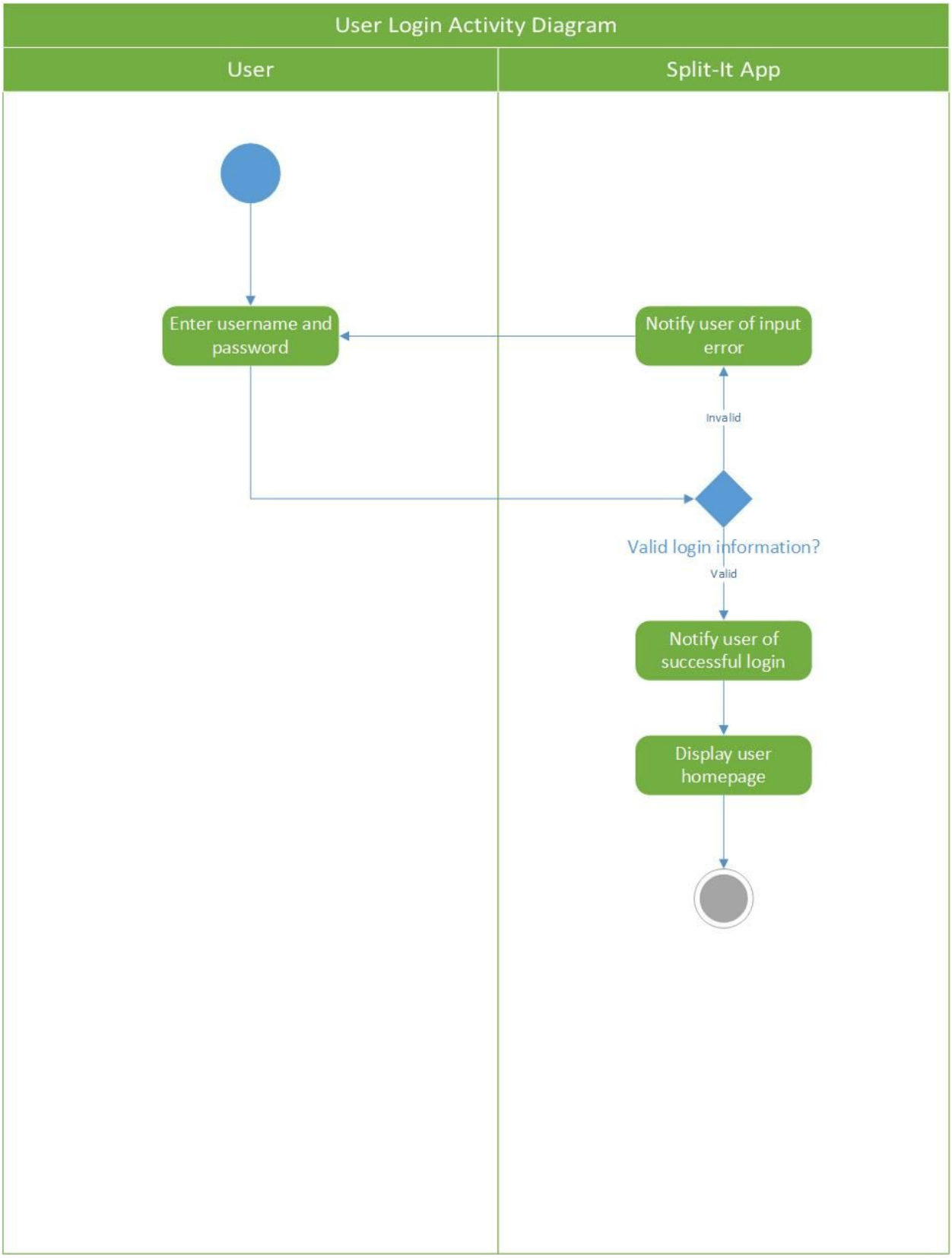
VII. Class Diagram

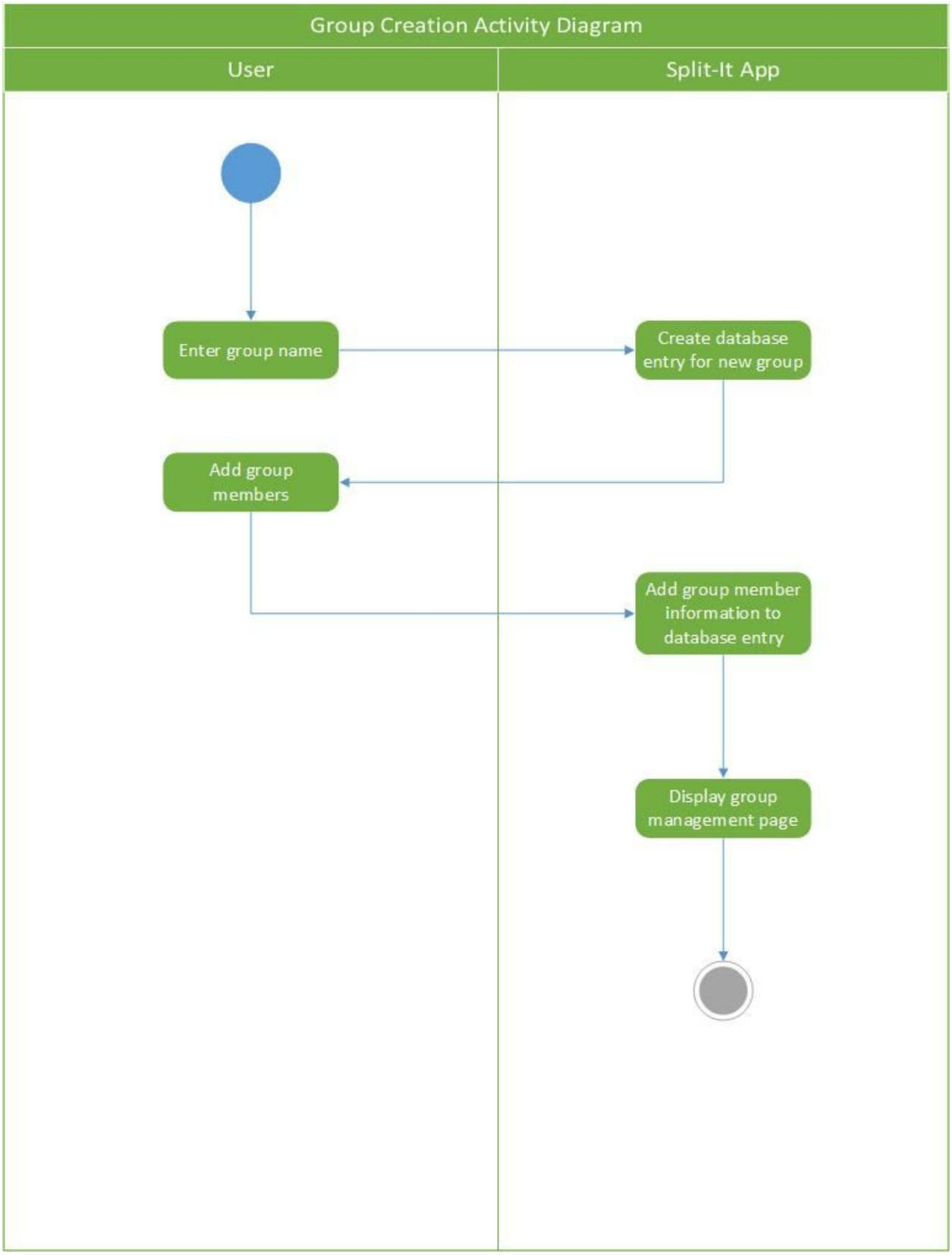


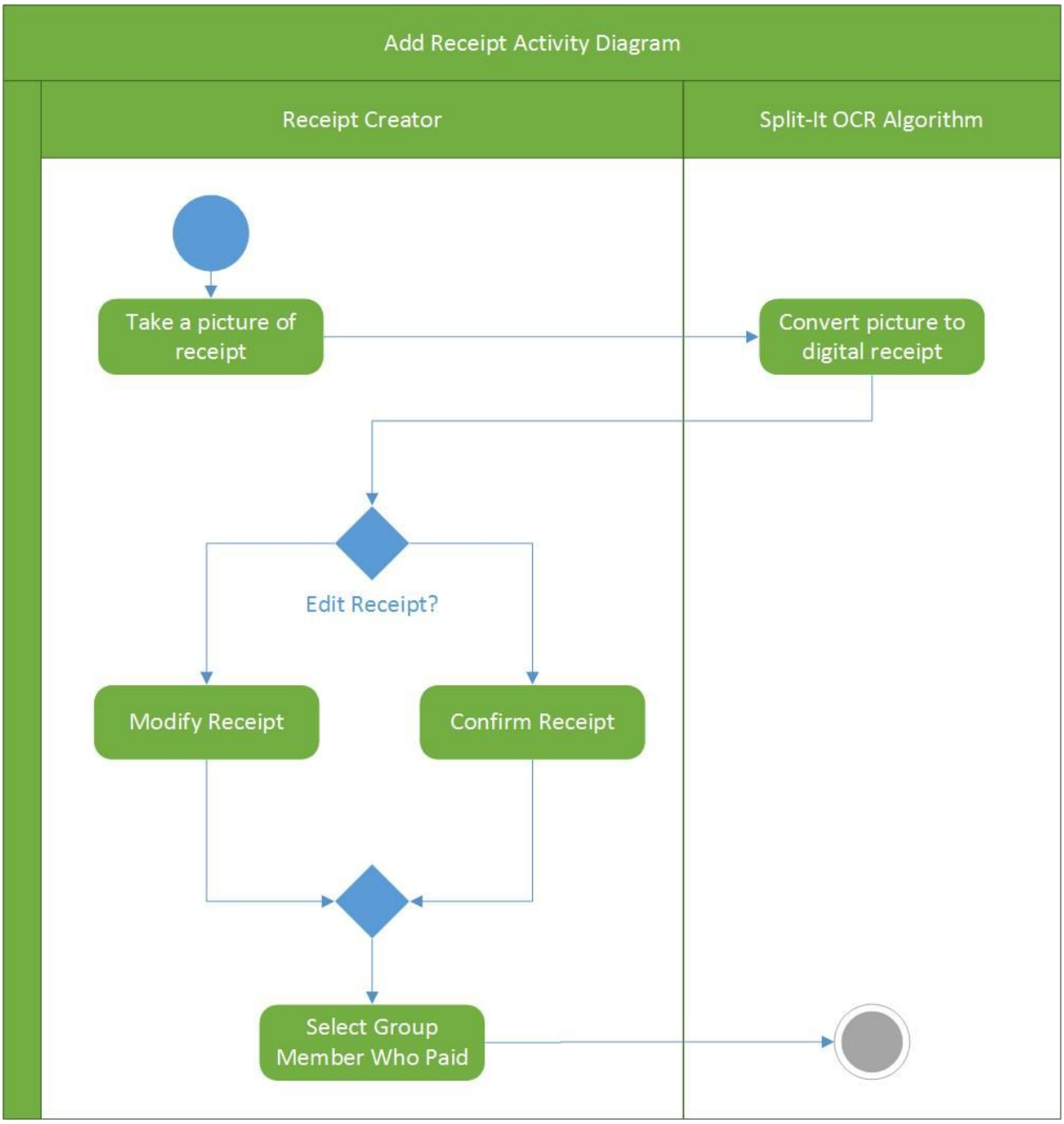
There are three primary objects that compose the application. The most important object is User. This class represents the individual operating the app, storing personal information such as name and Venmo ID as well as their login details. ReceiptCreator, GroupOwner, and GroupMember extend the User class. 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.

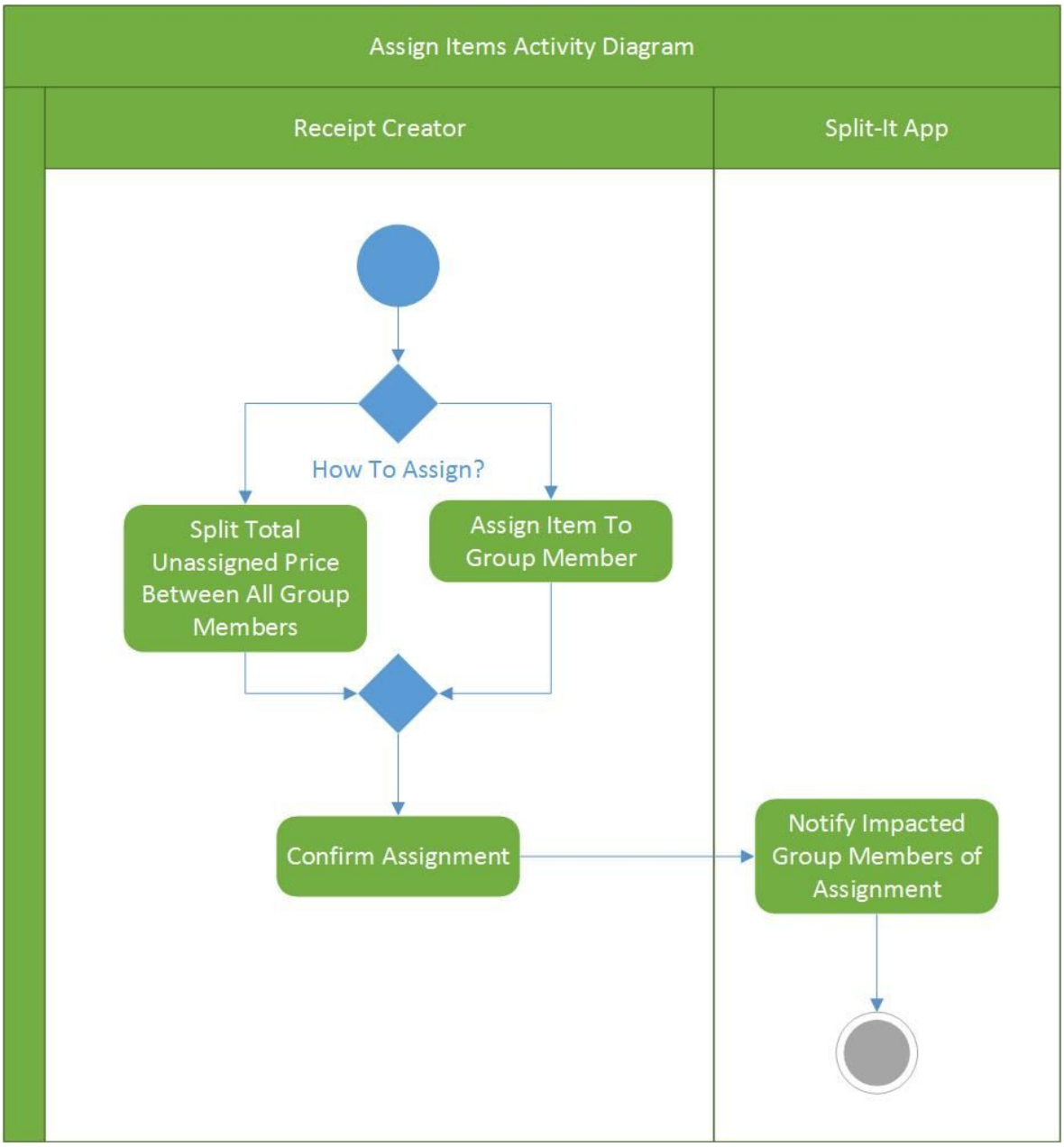
VIII. Activity Diagrams

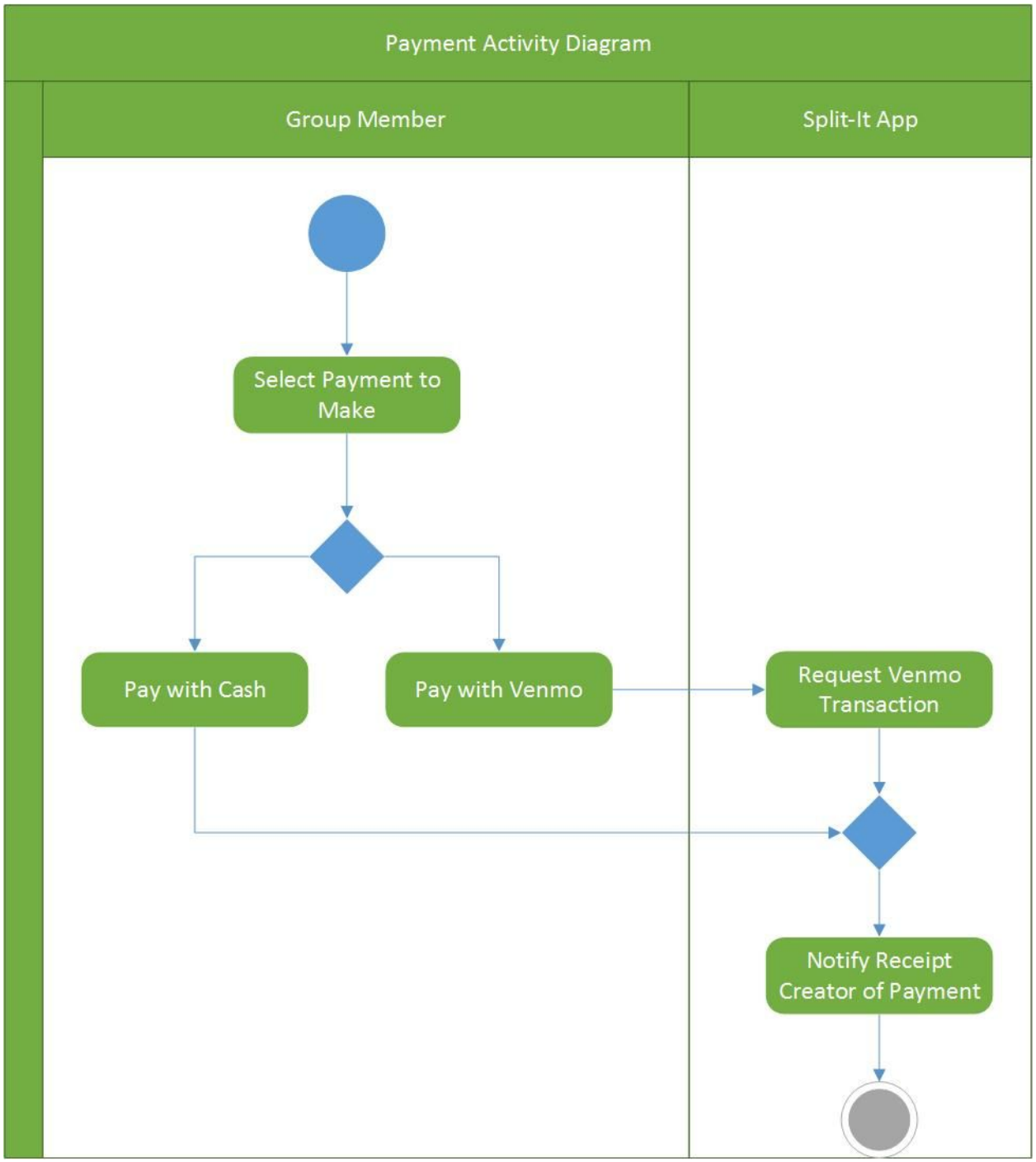


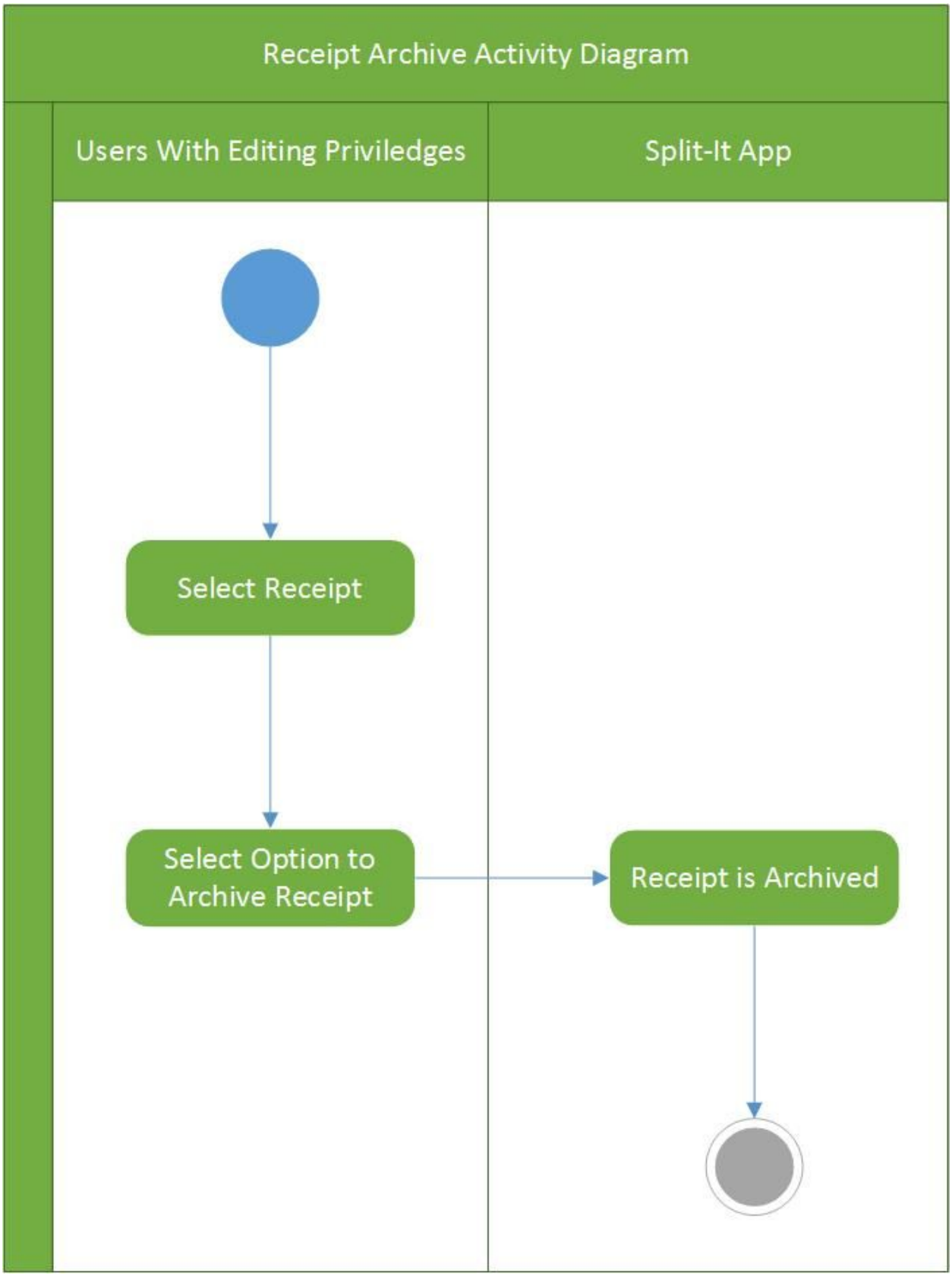

User Login Activity Diagram
User
Split-It App
Enter username and password
Notify user of input error
Invalid
Valid login information?
Valid
Notify user of successful login
Display user homepage

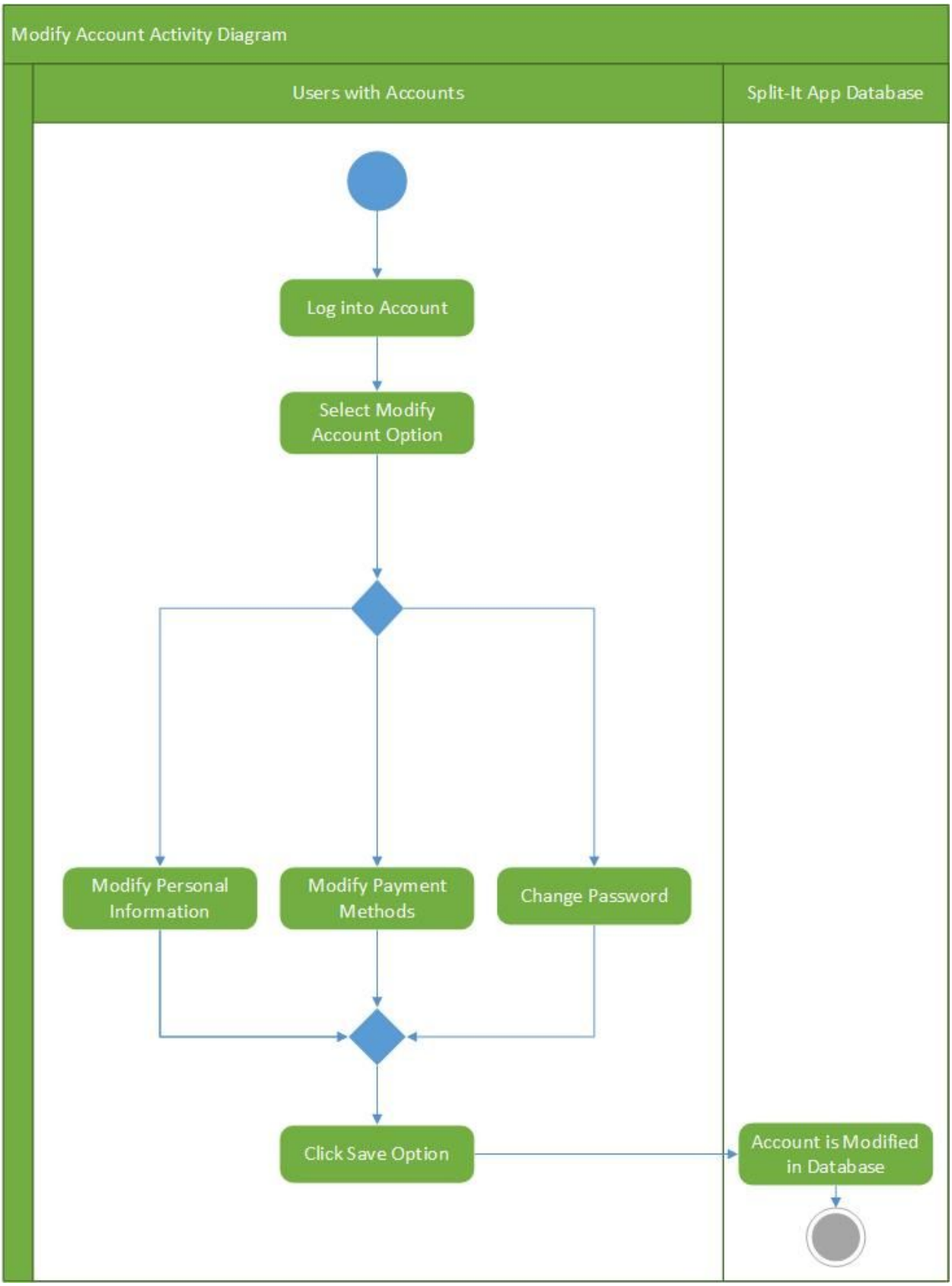

Group Creation Activity Diagram
User
Split-It App
Enter group name
Create database entry for new group
Add group members
Add group member information to database entry
Display group management page

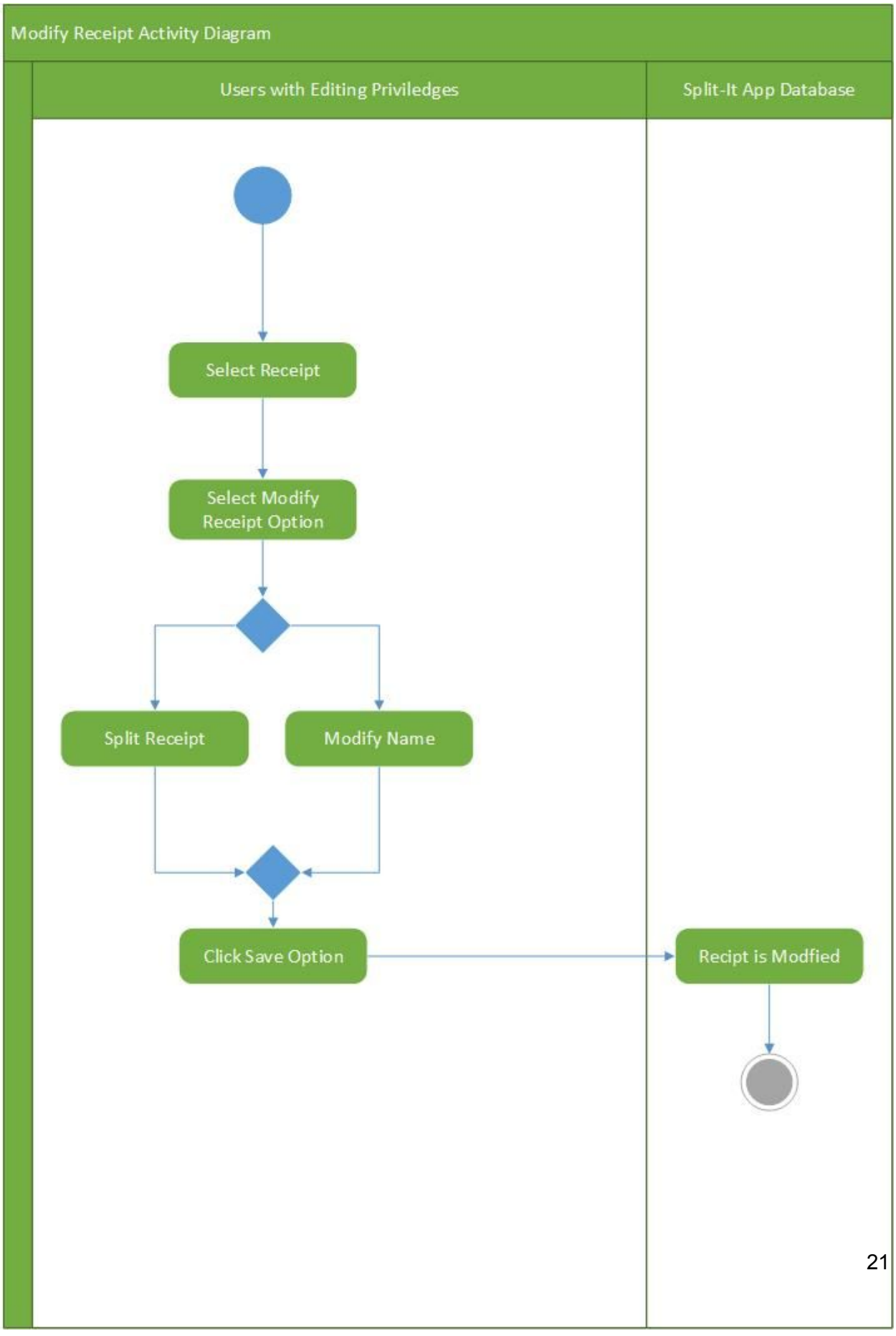

Add Receipt Activity Diagram
Receipt Creator
Split-It OCR Algorithm
Take a picture of receipt
Convert picture to digital receipt
Edit Receipt?
Modify Receipt
Confirm Receipt
Select Group Member Who Paid

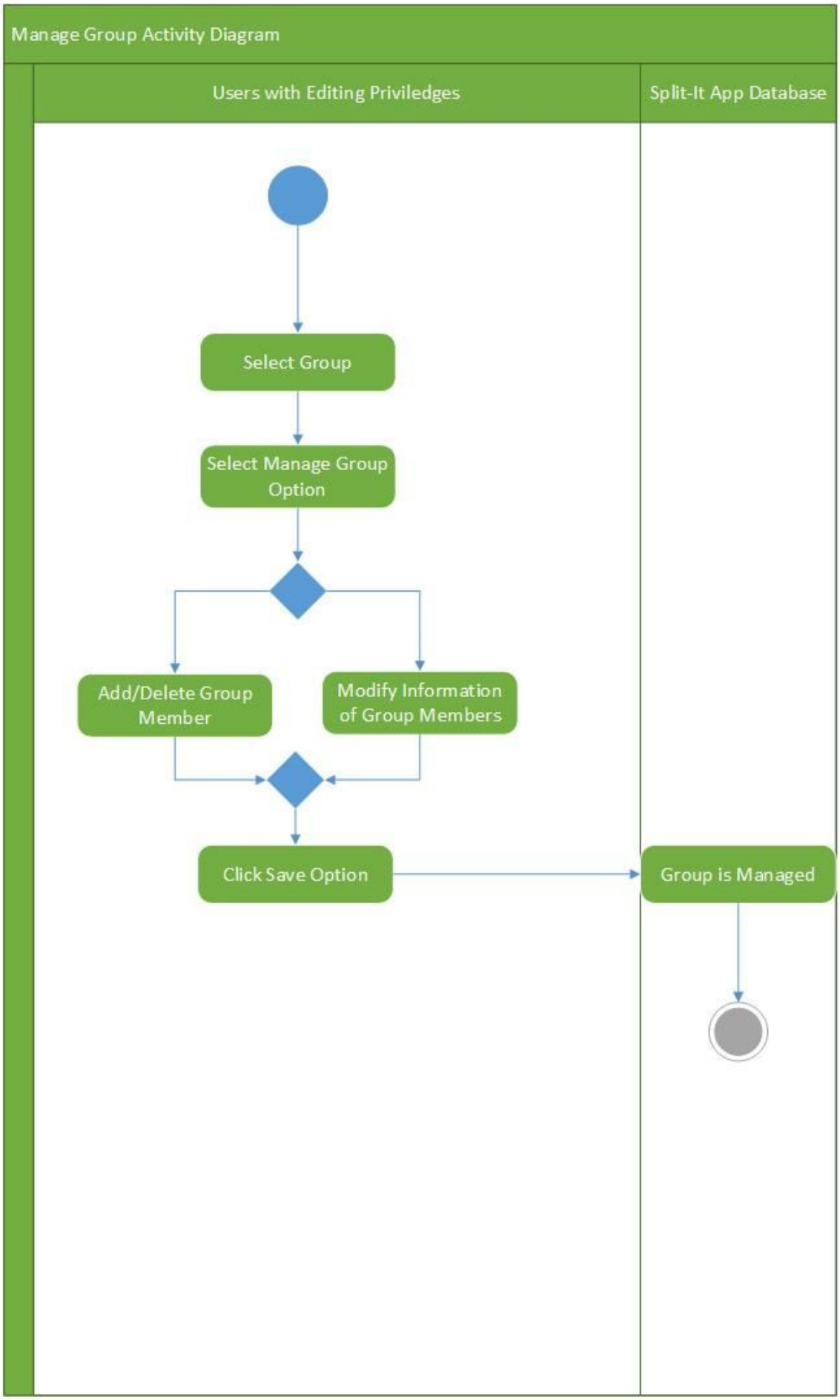

Assign Items Activity Diagram
Receipt Creator
Split-It App
How To Assign?
Split Total Unassigned Price Between All Group Members
Assign Item To Group Member
Confirm Assignment
Notify Impacted Group Members of Assignment


Payment Activity Diagram
Group Member
Split-It App
Select Payment to Make
Pay with Cash
Pay with Venmo
Request Venmo Transaction
Notify Receipt Creator of Payment


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


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


Modify Receipt Activity Diagram
Users with Editing Priviledges
Split-It App Database
Select Receipt
Select Modify Receipt Option
Split Receipt
Modify Name
Click Save Option
Recipt is Modfied


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