# Use Case

## Brief Use Case

1. **Sign Up Account**: create an account with name, email address, password.
2. **Sign In Account**: sign in with email address and password or Facebook or Google Account.
3. **Edit Account**: modify user information of name, date of birth, gender, email address, or password
4. **View list of receipts**: view the list of receipts after logged in
5. **View the details of receipt**: view the details of receipts such as store, total, items, description, comment, tag, receipt, and picture.
6. **Create a receipt**: take a picture of receipt and record a new receipt with store, total, type, items, etc.
7. **Edit a receipt**: select a receipt and modify data and then store at database
8. **Delete a receipt**: select a receipt and delete data from database
9. **Search for receipts**: filter data and show the list of results
10. **Organize data structure**: predefine categories for expenses such as store, type, tag, etc.
11. **Create Report**: create and export report defined by user such as store or period and show or export into a file format with different styles of charts

## Fully Dressed Use Case

|  |  |
| --- | --- |
| **Name** | Process Receipt |
| **Scope** | Receipt Keeper Android App |
| **Level** | User goal |
| **Primary Actor** | Any user |
| **Stakeholders and Interests** | Personal User – wants to be able to gather most crucial pieces of data from a receipt as quickly and easily as possible |
| **Preconditions** | A receipt is available to take an image of. |
| **Success Guarantee** | Receipt record is saved with at least correct amount and store name |
| **Main Success Scenario** | 1. User initiates photographing receipt 2. Receipt Keeper uses continuous preview to assist the user, provide simple feedback about photo quality 3. App notifies user about confidence level in a simple way so they can easily judge when to take photo 4. O.C.R. automatically attempts to extract amount and store name from bottom and top of receipt respectively 5. App presents extracted data to user, allows user to correct data before saving 6. User saves data, receipt record is stored locally |
| **Extensions** | 2a. Continuous preview is slow or not set up properly   1. User can override process and enter data manually   3a. Quality of receipt printing is difficult for O.C.R. to read   1. App allows user to complete process and take photo anyway 2. App still allows user to update data, not dependent on confidence level of O.C.R. 3. App can display reminder tips to check lighting, angle of phone relative to receipt, etc.   3b. User takes photo when confidence level is low   1. App allows user to confirm whether they want to save the data or not 2. If they don’t save, they can try taking another photo   5a. User is dissatisfied with results and does not wish to save data   1. App will allow user to attempt retaking photo   6a. Issues with local storage, SQLite problems   1. If local saving fails, warn user of technical issue that needs resolving |
| **Special Requirements** | * For Continuous Preview to work with O.C.R., the continuous preview and language settings must be selected * Receipt must be printed clearly enough to be easily readable for best results |
| **Technology and Data Variations List** | * Some receipts may print several amounts, it is assumed that an amount at the bottom of the block of text from the O.C.R. is the most likely amount to save * It is assumed store name is on the top one or two lines of the text retrieved |

## Fully Dressed Use Case

|  |  |
| --- | --- |
| **Name** | Sign Up Account(Mobile) |
| **Scope** | Receipt Keeper Android App |
| **Level** | User goal |
| **Primary Actor** | Any user |
| **Stakeholders and Interests** | Personal User – wants to be able to start using the mobile app |
| **Preconditions** | User has a mobile device currently on Wi-Fi or other Internet connection and has installed Receipt Keeper and allowed the needed permissions. |
| **Success Guarantee** | User is able to complete account creation and Sign In. |
| **Main Success Scenario** | 1. User Enters an email 2. User enters a password 3. User enters a name 4. User taps Submit 5. User selects Auto Login option |
| **Extensions** | 1a. . User enters data that is not of a proper email format   1. User gets an appropriate error message after tapping Submit   4a. User taps Submit before all fields have been filled in   1. Appropriate error messages are displayed   4b. User taps Submit when the back end is not running   1. User gets an error message suggesting they tap the link to activate the back end.   5a. When running receipt keeper, the app will go to the main ListView activity without requiring sign in when this option is on. |
| **Special Requirements** | * For a successful result, the back end should be up and running but this can’t be guaranteed at all times |
| **Technology and Data Variations List** |  |

## 

## Fully Dressed Use Case

|  |  |
| --- | --- |
| **Name** | Sign Up Account(Web) |
| **Scope** | Receipt Keeper Android App |
| **Level** | User goal |
| **Primary Actor** | Any user |
| **Stakeholders and Interests** | Personal User – wants to be able to start using the web app |
| **Preconditions** | User has an Internet connection and a common browser running(Chrome, IE), and an e-mail address and password they would like to use. |
| **Success Guarantee** | User is able to complete account creation and Sign In. |
| **Main Success Scenario** | 1. User Enters an email 2. User enters a password 3. User clicks SignUp 4. User enters the web app and is able to view the available features. |
| **Extensions** | 1a. . User enters data that is not of a proper email format   1. User gets an appropriate error message after tapping Submit   3a. User taps SignUp after entering an email that is already registered   1. Appropriate error messages are displayed   3b. User tries to tap SignUp without entering a valid e-mail   1. SignUp button is not available in this scenario   3c. User taps SignUp without entering a password   1. An appropriate error message is displayed |
| **Special Requirements** | * For a successful result, a reliable Internet connection is needed. |
| **Technology and Data Variations List** |  |

## Fully Dressed Use Case

|  |  |
| --- | --- |
| **Name** | View the details of receipt |
| **Scope** | Receipt Keeper Android App |
| **Level** | User goal |
| **Primary Actor** | Any user |
| **Stakeholders and Interests** | Personal User |
| **Preconditions** | User has a mobile device currently on Wi-Fi or other Internet connection and has installed Receipt Keeper |
| **Success Guarantee** | User is able to view list of receipts and select one to view details |
| **Main Success Scenario** | 1. User selects and touches a receipt from the ListView 2. The app goes to the View Receipt Activity 3. The user scrolls the screen to see all details: view the details of receipts such as store, total, description, comment, tag, receipt, and picture. 4. User taps the image to see a larger version |
| **Extensions** | 1a. The user has not yet saved any receipts   1. The system notifies the user of this and handles it cleanly   2a. Some data for the receipt was not filled in when the receipt was added.   1. The app displays what data is available and the absence of data does not cause errors   3b. The User reorients the screen   1. The app reorients the view 2. Scrolling still allows the user to see all data and image   5a. User taps the background view after viewing larger image   1. App returns to original view. |
| **Special Requirements** | * Most android devices should have a screen size and resolution that will work well for this, there are no exceptional requirements for that. |
| **Technology and Data Variations List** | * Different receipts will have different information filled in, all may not be complete. * Viewing in both landscape and portrait mode should be tested on different devices with different screen sizes. |