# Use Case

## Brief Use Case

1. **Sign Up Account**: create an account with name, email address, password, date of birth, and gender or Facebook or Google Account.
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 |