# *Budget Management*

### Requirements and Specification Document

### Authors: Anna Kim, Ali Zaidi, Na Li, Jack Long, Joon Lim, Andy Boho

Rev. 2.1<2020-03-30>: Iteration 1\_Final version

## Project Abstract

Many people write a budget planner or an account book to know how much money they have spent and where they spend their money. However, most people regret excessive spending while checking their spending report at the end of the month, since they could not realize it while spending their money. Also, some people struggle to control spending habits, especially the small amount of money they spend almost every day, such as buying a cup of coffee. The money management application does not only allow users to keep track of their income, expense, and remaining balance, but also alert them if they spend excessive amounts of money on a certain category. The goal of the application is to help people who have less experience on managing budgets or struggle to control their spending habits. The application provides features in account books such as tracking income, expense, and balance. Additionally, the application can remind a user when he or she tries to spend excessively by sending a notification.

## Document Revision History

Rev. 1.0 <2020-02-22>: Initial version

Rev. 1.1 <2020-02-25>: Added system architecture and design details

Rev. 1.2 *<2020*-*02*-*27>*: Added implementation plan and testing plan

Rev 1.3 *<2020*-*02*-*28>*: Added UML diagrams, Finalized Document

Rev 2.0 <2020-03-24>: Iteration 1\_Initial version

Rev. 2.1 <2020-03-30>: Iteration 1\_Final version

## Customer

The app is specially designed for users seeking help for budget management but with limited budget for paid apps. This app also works best for people having security concerns of connecting apps with bank accounts and hope to manage the budget more safely.

The client of the project is University of Wisconsin Credit Union. There are increasing needs to manage the budget for customers in UW Credit Union. The Budget Management app is specially designed for UW Credit Union to meet this need, so that UW Credit Union can provide a better service for its customers with the option of offering a budget management strategy. We plan to add the Budget Management app to the official website of UW Credit Union, so that only people having a UW Credit Union account can access the app. We are still waiting for the official corporate approval from UW Credit Union.

A large percent of UW Credit Union users are students attending University of Wisconsin campus. The budget management application provides a free and easy way to manage budget for the university students who have less budget managing experience.

## Competitive Landscape

There are some budget management apps in the market designed for budgeting beginners, such as You Need a Budget, Mint, Mvelopes, Quicken, CountAbout etc. The budgeting apps can take the tedium and challenge out of the budgeting process. They can help users to stick to the spending plan and many of the apps come with handy extra features that depend on the users’ needs. However, some of the apps are designed to be hard for users to follow, in which users may need to sign up for class with a live instructor if help is needed. Budget apps like Quicken, are a bit old fashioned in some way, though it sets the standards and does all the basic things for budgeting apps. Most apps that are popular in the market require users to pay some money on a monthly or yearly basis, which greatly increases the cost of budget management. For example, Mvelopes users pay $6 - $59/month for different versions of the app. While Quicken charges $35 - $90 for users with different budgeting needs and expertise. The high cost of using these budget apps may scare away people with limited budget for extra spending on apps. For free apps, such as Mint, users have to deal with a lot of ads, which makes the using experience annoying and less efficient. Most importantly, Mint allows users to connect with banks and lenders, though it makes tracking of spending and balance easier, it also causes security concerns when users click on the ads posted in the apps.

For our project, we aim to build an app that is free to use and allows self management of the budget. Users set the monthly income and budget to use and log each spending through the app. This might be especially useful for customers with basic needs to track spendings. It will also be a good option for users having security concerns for using apps with a bank account connected.

## User Requirements

The main idea of the budget management mobile application is to allow users to monitor and manage their spending. The features of the application are listed below.

1. When user opens the budget management application, login page is shown **(1st Priority)**
   1. Create account, lead to create account page
      1. Enter user name
      2. Enter password
      3. Select two security questions among 10 security questions
      4. Answer to these selected two questions
      5. Create account (button)
   2. Login
      1. Type user name
      2. Type password
      3. login (button)
   3. Forgot password: settings-Change/forgot password
      1. Enter username
      2. Answer security questions
2. After the user logs in with their Username and password number, the user can see a navigation bar on every screen of the application and change a page by clicking a desired page’s icon **(1st Priority)**
   1. Home
   2. Report
   3. Goal
   4. Category
   5. Notification
   6. Calculator
   7. Setting
3. Home page **(1st Priority)**
   1. Current balance of current month
   2. Current spending of current month
   3. Target spending of current month
   4. Add transaction
      1. a category
      2. type of transaction (Expense or Income)
      3. spending amount
      4. Note
      5. If the result of adding transaction exceed user specified percent of goal amount
         1. Warning pop up
   5. Daily expense trend of current month
4. Report page **(1st Priority)**
   1. A range of month selector
      1. Select start month, year
      2. Select end month, year
   2. Pie graph of expense
      1. Show ratio of categories
      2. Show spending amount for each category
   3. Total spending of specified range of months
   4. Total income of specified range of months
   5. Transaction history
      1. Type of transaction
      2. Amount
      3. Category
      4. Date
      5. Edit or remove past transactions
      6. filter transactions using a category
   6. A bar graph showing monthly expense trend
5. Goal page **(2nd Priority)**
   1. Show current month as a default
      1. select a past month to view past month’s goal
   2. Show a current progress of a goal with a bar graph
      1. Income and expense as a default
         1. goal amount
         2. Current amount
      2. Categories with user specified goal
         1. goal amount
         2. Current amount
   3. target balance
      1. Edit amount of target balance
   4. Target income
      1. Edit amount of target income
   5. Target expense
      1. Edit amount of target expense
6. User customizable categories with sub-categories **(1st Priority)**
   1. Show categories
      1. Categories under Income section
      2. Categories under Expense section
         1. goal amount (2nd Priority)
      3. Sub-categories under a category
         1. goal amount (2nd Priority)
   2. Select category
      1. Show expense history of selected category
   3. Create a new category or sub category (select a parent category)
      1. Type (income or expense)
      2. Name of a category
      3. Goal amount of a category (or empty as ‘None’)
   4. Edit or remove category
      1. Select a category to edit or delete
         1. Edit
            1. Rename a category
            2. Type of category (income or expense)
            3. Goal amount of a category (or empty as ‘None’)
         2. Remove
7. Calculator **(3rd Priority: additional feature)**
   1. Tax calculation
      1. Cost
      2. State
      3. Displayed amount of tax based on entered cost
   2. Tip calculation
      1. tip percentage
      2. Cost
      3. Displayed amount of tip based on entered cost
8. Settings **(2nd Priority)**
   1. Change/Forgot password (1st Priority among settings)
      1. Answer to verification questions to change password
      2. If the user forgets their password, they can click forgot password and then will be prompted with the two security questions they selected in the beginning.
         1. If the user answers correctly, the user can reset the password.
   2. Download transaction history (2nd Priority among settings)
      1. Select term
         1. Create list and download
   3. Notification (3rd Priority among settings: additional feature)
      1. Set a condition of sending notification in percentage
      2. Turn on or off a notification
         1. Expense
         2. Income
         3. Categories with a goal
   4. User guide: application tutorial (3rd Priority among settings)
   5. Application information (3rd Priority among settings)
      1. System requirements
      2. Terms of service
      3. Version of the application

## Use Cases

|  |  |
| --- | --- |
| Name | Set password |
| Actors | First time user of budget management application |
| Triggers | When the app is first downloaded and opened. |
| Events | A user is asked to select three security questions and answers to them. Then the user can type in four digits of password two times. If the second password is matched with the first password , the user will be led to the home screen. If those are different, the user needs to set a password again. |
| Exit Condition | The first and second password are matched. |
| Post-Conditions | The user is brought to the home screen. |
| Acceptance Test | A correct password moves the user to the home screen, and the Username, password and answers to the three questions are stored in the database; while an incorrect password keeps the user at the setting password page. |

|  |  |
| --- | --- |
| Name | Log In |
| Actors | Whenever a user wants to get into the app. |
| Triggers | When the user clicks on the app from their home screen |
| Events | The user is prompted to enter their username and password from the log in window. |
| Exit Condition | The username and password are found and match the same that are in the database. |
| Post-Conditions | The user is brought to the main screen of the app where they can then carry out all the functions of the app. The data that’s stored for their account will be displayed after they log in and it is fetched from the database. |
| Acceptance Test | A matching username and password with one that already exists in the database within the server will verify that the user entered valid information and is able to log in to the app. |

|  |  |
| --- | --- |
| Name | Open the application |
| Actors | A user of budget management application |
| Triggers | When the app is first opened. |
| Events | A user can type in his/her Username and four digits password. The correct Username and password will lead to the home screen, the wrong Username or password will return the user to the login page. |
| Exit Condition | Correct Username and password are entered. |
| Post-Conditions | The user is brought to the home screen. |
| Acceptance Test | Correct Username and password move the user to the home screen, while an incorrect password asks the user to change password by answering security questions, and an unrecognized Username will lead the user to the register screen. |

|  |  |
| --- | --- |
| Name | Add Recent Transaction |
| Actors | Any user of budget management application |
| Triggers | The application is on the home page |
| Events | * Select transaction type among expense and income * Select a category * the transaction information (price, category, note) is entered   User click OK button to save a transaction |
| Exit Condition | The user hits the OK button. |
| Post-Conditions | The user then must be able to return to the home screen. |
| Acceptance Test | The transaction is added to the user’s monthly report, and the added transaction information such as date, amount etc. are stored in the database; Account balance is also calculated after the transaction is made and updated in the database. |

|  |  |
| --- | --- |
| Name | Setting budget goal |
| Actors | Any user of budget management application |
| Triggers | User clicks on a button on the setting tab through the Home Screen (optional) |
| Events | A user can choose to have a budget goal at the beginning of each month or the first time a user uses the app.   * The default choice for the budget button is closed. * If a user chooses to set a budget goal, he/she needs to type in the amount of the budget. |
| Exit Condition | The user clicks on the “save” button to save the budget option. |
| Post-Conditions | The user can click on the “edit” button to modify the value of the budget or choose to close the budget button. |
| Acceptance Test | The user will have the budget set up and show up on the home page if a value is typed in. Date and amount of budget set for a given month are stored in the database. |

|  |  |
| --- | --- |
| Name | Changing balance for the month |
| Actors | Any user of budget management application |
| Triggers | The user will click on the “Goals” tab on the navigation bar and then navigate to the bottom and click “Edit Target Balance”. |
| Events | The user will be prompted with a pop-up that allows them to edit the current amount of money that will be left over in their account with the current budget applied. If the number doesn't fit in with the users budget, they will be told that their balance isn’t valid. |
| Exit Condition | The user will click “Apply” |
| Post-Conditions | The user must be taken back to the “Goals” screen |
| Acceptance Test | The new balance must be reflected along with the current budget the user has already set. |

|  |  |
| --- | --- |
| Name | Setting notifications |
| Actors | Any user of budget management application |
| Triggers | User clicks on the setting tab through the Home Screen (optional) |
| Events | The user will be able to choose whether they want notifications:   * If they go over budget for the month * If they get close to the budget for a certain category for that month (ex: 75% of the spending) * If they go over budget for a specific category in that month |
| Exit Condition | The user will either check the box saying that they would like to receive notifications for that specific category, or leave the box unchecked and the user will not receive notifications for the specified category |
| Post-Conditions | The user can click on on another section of the app through the the button navigation bar |
| Acceptance Test | The user will get a notification when he/she has spent over what they’ve put in as their budget |

|  |  |
| --- | --- |
| Name | Set Budgets for Individual Categories |
| Actors | Any user of budget management application |
| Triggers | The user clicks on the categories tab at the bottom navigation bar |
| Events | Once the user is redirected to the categories page, he/she is able to add categories (optional). When they add a category, they will also be prompted to add a budget for that category (optional). |
| Exit Condition | The user must be able to click “add” and the category with the corresponding budget will be saved. |
| Post-Conditions | The category will be added to the user’s total budget, and appear in both the goal tab as well as remains in the category tab |
| Acceptance Test | The category that is selected or inputted by the user will now appear in the goals tab so that the budget can be edited. |

|  |  |
| --- | --- |
| Name | Forgot password |
| Actors | Any user of budget management application |
| Triggers | The user clicks on the link under the login page with the message “Forget your password?” |
| Events | The user will be prompted with the three security questions that they chose when they started the app. If they answer one of them correctly, they will be presented with their password number. |
| Exit Condition | The user will be prompted with their password number |
| Post-Conditions | The password number will appear as a pop-up to the user and then the user will have to log in with the password they received. |
| Acceptance Test | The user can now log in with the password that they have been presented with. |

|  |  |
| --- | --- |
| Name | Adjust monthly budget |
| Actors | Any user of budget management application |
| Triggers | The user will click the “Goals” tab at the bottom navigation bar. |
| Events | The user will enter the “Goals” tab being able to see the current categories they’ve inputted through the categories tab. Towards the bottom of the screen, the user must be able to edit the target balance, the target income as well as see the expenses that they’ve already spent throughout the course of the month. |
| Exit Condition | The user clicks “apply” after changing the target balance |
| Post-Conditions | The balance must be updated. |
| Acceptance Test | The user will be able to see the new budget on both the goal screen as well as the home screen if updated correctly. The database will be updated with a new monthly budget. |

|  |  |
| --- | --- |
| Name | See monthly report |
| Actors | Any user of budget management application |
| Triggers | The user will click on the monthly report tab in the navigation bar at the bottom of the screen. |
| Events | The user will select the last month as both the start and end date and a graph will pop up notifying the user of how much they spent last month as opposed to how much they hoped to spend last month. |
| Exit Condition | When the user picks the month they would like to see as the end month. |
| Post-Conditions | The graph pops up on the screen with the spending from start to end |
| Acceptance Test | The user must be able to see how much they have spent from the time period that they inputted. |

|  |  |
| --- | --- |
| Name | Download past data |
| Actors | Any user of budget management application |
| Triggers | The user clicks on the Download Transaction button to download the data. |
| Events | The user will be able to download the data they select from the menu. |
| Exit Condition | When download is complete |
| Post-Conditions | The downloaded data shows up in a given folder in the user’s cell phone. |
| Acceptance Test | The user must be able to see the downloaded data with graphs and summaries. |

|  |  |
| --- | --- |
| Name | See Spending Over a Period of Time |
| Actors | Any user of budget management application |
| Triggers | Visual representation of spending should be available at the home screen. Triggers for updates to the visuals will be the entering of new transactions or the beginning of a new monthly period. In addition, If the user wishes to see spending over a specified time period, they can click the “Report” tab on the navigation bar at the bottom of the screen. |
| Events | New month resets the visuals. New transactions update the total spending. The user is also able to pick a range of months in the Report tab through the navigation screen. |
| Exit Condition | When the home screen is navigated away from. |
| Post-Conditions | Must be able to return to an up to date version of the spending visuals when the home screen is navigated back to. |
| Acceptance Test | Spending visuals must properly correspond to the transactions that have been entered into the app by the user. |

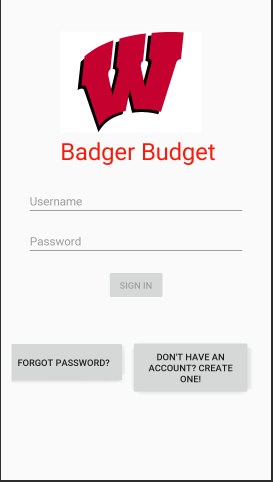
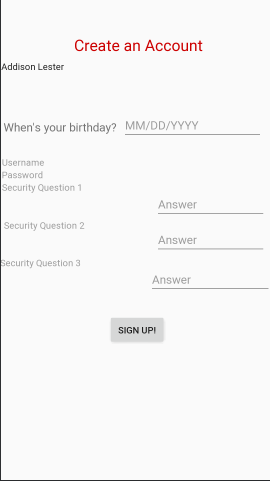
|  |  |
| --- | --- |
| Name | View spending by categories |
| Actors | Any user of budget management application |
| Triggers | The user can click on Category at the bottom of the app and check the spending by categories |
| Events | The user is able to see all spendings by category with their corresponding goal listed. |
| Exit Condition | The user can exit by close the app or switch to other section of the app |
| Post-Conditions | Must be able to return to an up to date version of the spending when the home screen is navigated back to. |
| Acceptance Test | Spendings correspond to the transactions that have been entered into the app by the user. |

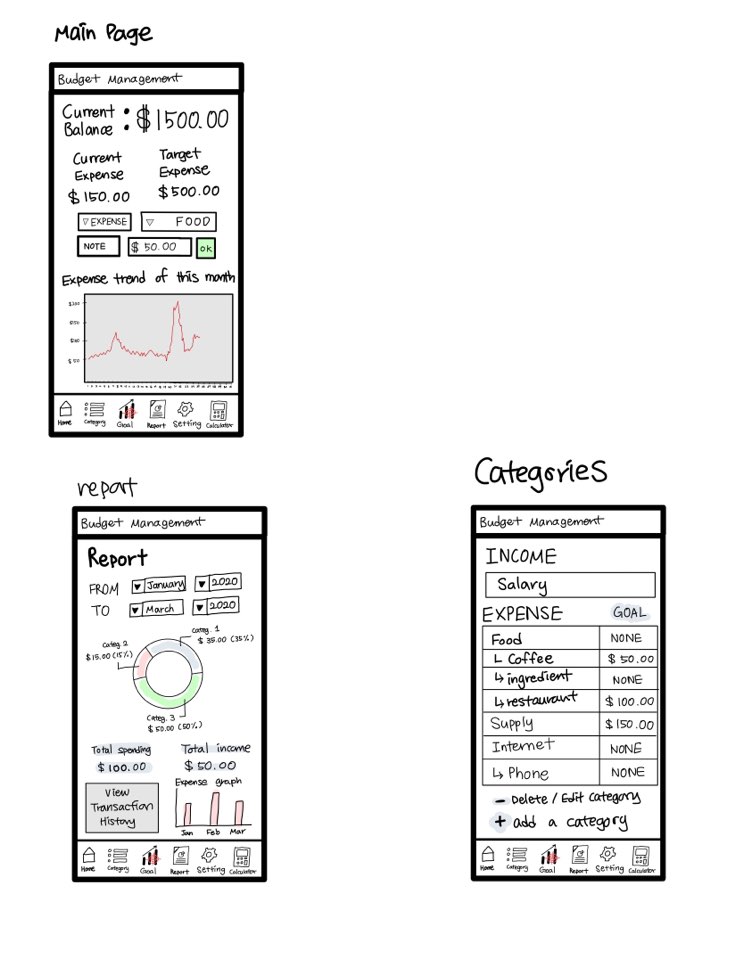
|  |  |
| --- | --- |
| Name | Calculate Tip |
| Actors | Any user of budget management application |
| Triggers | Opening the calculator from the navigation bar |
| Events | Entering the tip percentage, the price, and hitting Enter |
| Exit Condition | When the user has calculated the tip and navigates away |
| Post-Conditions | There should be no Post-Conditions. |
| Acceptance Test | The calculator successfully calculates a percentage. |

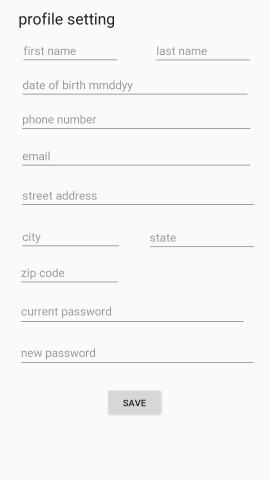
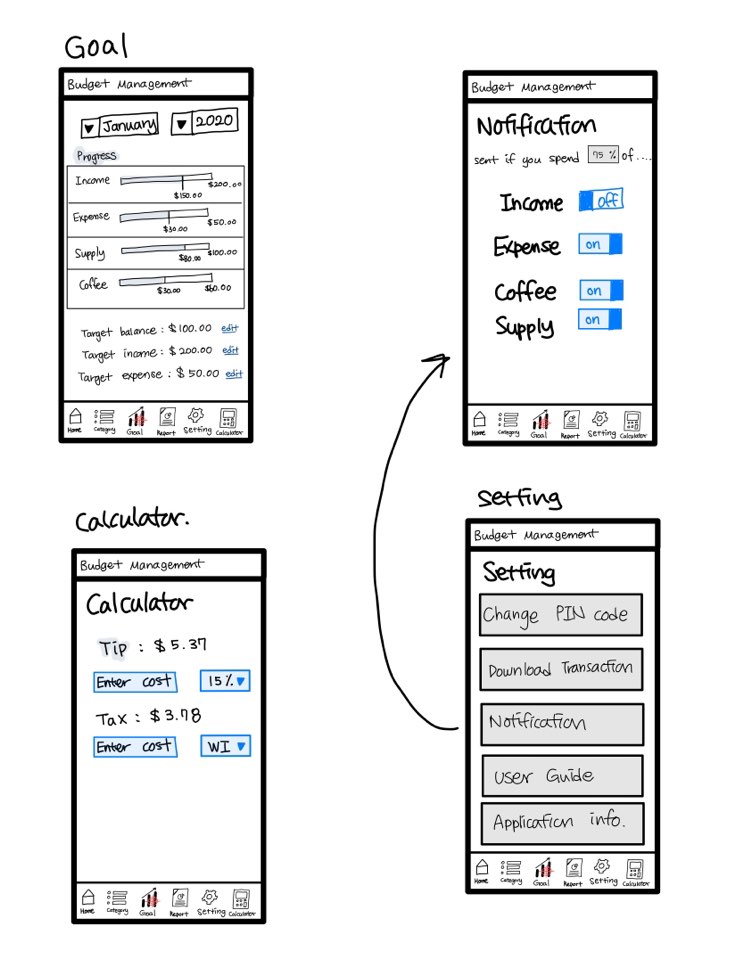
|  |  |
| --- | --- |
| Name | Calculate Tax |
| Actors | Any user of budget management application |
| Triggers | Opening the calculator from the navigation bar |
| Events | Selecting a state and entering the price, and hitting Enter |
| Exit Condition | When the user has calculated the tax and navigates away |
| Post-Conditions | There should be no Post-Conditions. |
| Acceptance Test | The calculator successfully calculates a percentage. |

|  |  |
| --- | --- |
| Name | Create a category for expense |
| Actors | Any user of budget management application |
| Triggers | Clicking ‘Add a category’ button on ‘Categories’ page |
| Events | * User clicks ‘Add a category’ button on ‘Categories’ page * The page is changed to ‘Create Category’ page * User enters in category information  1. User selects ‘Expense’ section by clicking a button    * + User types category name      + User types goal amount on a category (optional) 2. User selects ‘Income’ section by clicking a button  * User types category name * User clicks ‘Save’ button to create a category OR User clicks ‘Cancel’ button to not save it * The page is changed back to ‘Categories’ page |
| Exit Condition | The application stores a category that the user has created. |
| Post-Conditions | Added category is displayed on category list on the ‘Categories’ page   * User specified goal amount of the added category is also displayed if the user specified it. |
| Acceptance Test | The user can view a created category on the ‘Categories’ page under a section, ‘Expense’ or ‘Income’, that user specified. |

## User Interface Requirements





As the diagrams above show, there are several pages within our application, from where the user can do several different things. On the home page, which is the first page the user sees after “logging in”, the user will first see their current balance. Below this, the user can see their current expenses and their target expenses for the month. The user can also quickly view each category’s spending amount using a drop down menu, and a line graph depicting their spending over the last month.

At the bottom of the screen, there is a touch menu that will remain no matter what page the user is on, as it is meant to be a navigation menu. The user can get to the home, category, goal, report, setting and calculator pages from this menu.

The report page is where the user can view detailed reports of their budgets and spending over a period of time that they can define via a drop down menu. There are several graphs meant to visually show their spending, such as pie and bar graphs. They can also view their specific transaction history by touching a button on this page.

On the Categories page, the user can view their currently defined categories of income and expenses. There can also be subcategories within larger categories that the user can view on this page as well. They can also add, edit, or delete categories from this page.

The Goal page is where the user can set their target balance, income, and expenses. They can also view their current progress towards certain spending goals.

The calculator page allows the user to easily calculate tip and tax in situations where this is necessary, such as at a restaurant.

Finally, the settings page will allow the user to access various settings, such as changing their password , downloading transaction history, a user guide, and the application info. In addition, they can adjust the amount of notifications they receive from the application.

## Security Requirements

We decided to use a server and database rather than data being stored locally in the device. Since we are using a server, each user has their own account. This ensures access control since users that are logged in only have access to their own data. It also allows multi-users to access a single device and users to switch between devices. The app will allow users to create passwords to protect data from other people. In the case users forget the password, instead of the PIN code, there will be security questions when creating the account. Other people won’t be allowed to access this information. However, data regarding money transactions is private and sensitive and others might access the user's device.

Later if the app becomes popular, we will need to directly connect to bank accounts for convenience. Also, we will have to make user accounts and store data in the database. In this case, data regarding banks and users must be secured. So, all the data will be encrypted. Moreover, instead of security questions, users will be using their email or phone to reset their password if our app gets sold and the number of users increase a lot.

## System Requirements

Android 6.0 or higher

100 MB of Storage

Our app will need Android 6.0 or higher. We believe that most people with Android devices will have version 6.0 or higher. In other words, our app will be able to run on nearly any Android device. In fact, most phones with Android 6.0 and higher will have enough system memory and processor speed to run our app. Moreover, all smartphones these days have touch screens so that users can freely control their actions on the app.

We’ve decided that 100 MB of internal storage will be enough. Mint is one of the well known budget management apps that has been downloaded over 10 million times and it’s size is about 120 MB when installed and about 50 MB on Google Play store. Since it’s likely that Mint will have more features and data than our app, we believe that 100 MB is more than enough for our app.

## Specification

