

Budget Bunny

Project Specifications Document

Revision History

Revision	Date	Author(s)	Description
0.1	04/10/2016	Kiefer Yap	Created the document
0.1.1	05/11/2016	Kiefer Yap	Added Add New Account Screens
0.1.2	05/29/2016	Kiefer Yap	Added View Account Screens
0.1.3	06/08/2016	Kiefer Yap	Added Edit Account Screens
0.1.4	06/10/2016	Kiefer Yap	Added Delete Account Screens
0.1.5	06/15/2016	Kiefer Yap	Added Core Data Section
0.1.6	06/17/2016	Kiefer Yap	Added Use Case Diagram Section
0.1.7	06/21/2016	Kiefer Yap	Added Design Specifications Section

Contents

1	Support Specifications	4
1.1	Hardware	4
1.2	Software	4
1.3	Development Environment	4
2	Use Cases	5
2.1	Use Case Diagram	5
2.2	Use Case Description	5
3	Database Schema	8
3.1	Core Data Diagram	8
3.2	Model Attributes	8
3.2.1	Account	8
3.3	Model Relationships	8
3.3.1	Account	8
4	Application Screens	9
4.1	Adding a New Account	9
4.1.1	Application Screenshots	9
4.1.2	Error Scenarios	12
4.2	Currency Selection	13
4.2.1	Application Screenshots	14
4.3	Account List Display	15
4.3.1	Application Screenshots	16
4.3.2	Error Scenarios	17
4.4	Editing an Existing Account	18
4.4.1	Application Screenshots	18
4.5	Deleting an Existing Account	20
4.5.1	Application Screenshots	20
4.6	Adding a New Budget	21
4.6.1	Application Screenshots	23
4.6.2	Error Scenarios	25
5	Design Specifications	26
5.1	Add and Edit Account Screen	26
5.2	Currency Selection Screen	26
5.3	Account List Display Screen	26

1 Support Specifications

This document is empty at first, but will grow as project features are implemented. Screenshots and specific use cases will be specified during the implementation phase. The localization word list is located in the project's database dump, in the `/BudgetBunny/Scripts/output/budget_bunny.sql` file.

1.1 Hardware

iOS Devices: iPhone and iPad

Device Capabilities: None

1.2 Software

iOS Version: iOS 8.1 to 9.3.2

Device Orientations: Landscape, Portrait, Landscape (Inverted)

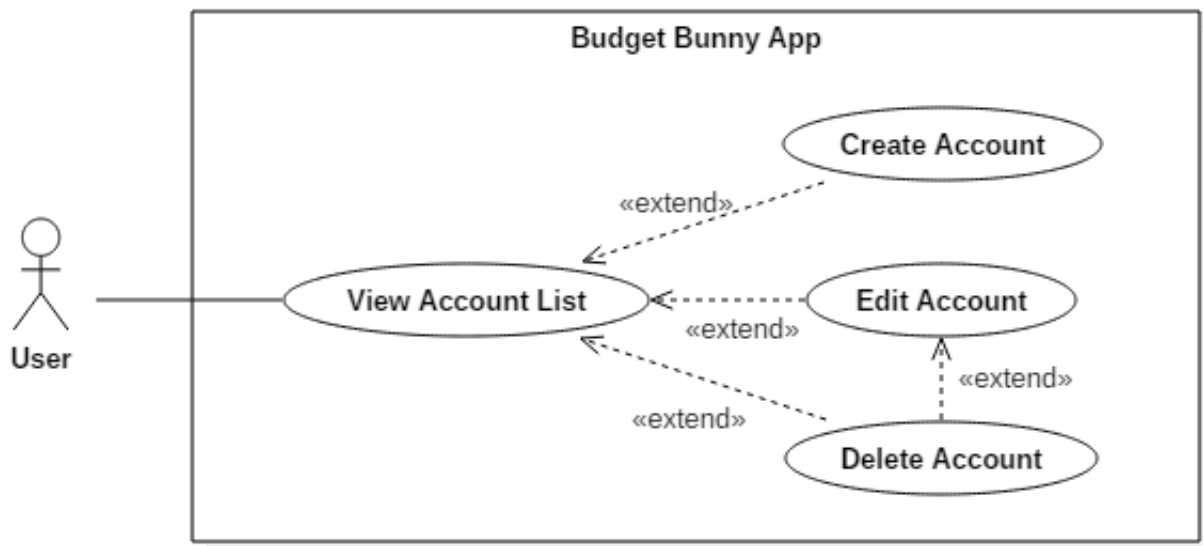
Language Support: English, Japanese, Chinese

1.3 Development Environment

XCode Version: XCode 7.3.1

2 Use Cases

2.1 Use Case Diagram



2.2 Use Case Description

Unless otherwise stated, the primary actor for all use cases is the user of the application, which shall simply be called, "User". Note: the Usage Frequency portion is inaccurate for now. Once the usage data has been gathered and collated, the Usage Frequencies of each use case will be updated.

Use Case ID: ACC-0001

- **Title:** Create Account
- **Description:** The user creates a new account, supplying its name, currency, and initial amount. All fields are required. The user may also set the newly created account as default.
- **Precondition:** The number of accounts in the core data is not more than 20, which is the maximum number of accounts.
- **Postcondition:** The account is added into the core data.
- **Main Success Scenario:** The user enters the name in alphanumeric characters, and the initial amount in numeric characters, both within the limits of the text field's length. The user then chooses the account's currency, and taps the Done button, saving the newly created account into the core data.
- **Extensions:**
 - All fields are required. When at least one field is not filled, an error alert will appear.
 - Both the account name and the initial amount text fields have a limit in their length. When the user hits the limit and keys in a new character, the text field will not respond.
 - On a related note, when the user pastes text into the text field with a length longer than the given limit, it will not respond as well.
 - No two account names must be the same. An alert will be shown if the name already exists in the core data.

- **Usage Frequency:** N/A

Use Case ID: ACC-0002

- **Title:** View Account List
- **Description:** The user views a list of all their accounts.
- **Precondition:** There is at least one account in the core data available for viewing.
- **Postcondition:** The state and number of accounts before and after viewing stays the same.
- **Main Success Scenario:** The user is able to view all their accounts.
- **Extensions:**
 - When the user swipes a non-default account cell to the left, two buttons appear, enabling the user to either mark the account as default, or delete the account.
 - Default accounts cannot be deleted, nor can they be set as default again. When the user swipes a default account cell to the left, one button appears, enabling the user to edit the account. Ultimately, this means that there is always at least one account available for viewing.
 - All table cells can be tapped. Once tapped, the user proceeds to the Edit Account Screen.
 - If there are no accounts available for viewing, then the table will not be visible, containing a block of placeholder text in its place. This scenario is technically not possible, because upon installation of the app, placeholder accounts will be automatically created.
- **Usage Frequency:** N/A

Use Case ID: ACC-0003-1

- **Title:** Edit Account
- **Description:** The user can edit the account's name, current amount, or the currency.
- **Precondition:** There is at least one account in the core data available for editing.
- **Postcondition:** The user has updated the account that they wanted to edit.
- **Main Success Scenario:** The user edits all the fields in the account – the name, current amount, and the currency and taps save, updating the core data with the changes.
- **Extensions:**
 - If the account is not a default account, the user can set the account as default, or they can delete the account.
 - All conditions in ACC-0001 apply here. Specifically, all fields are required, all fields have a character limit, and no two account names must be the same.
- **Usage Frequency:** N/A

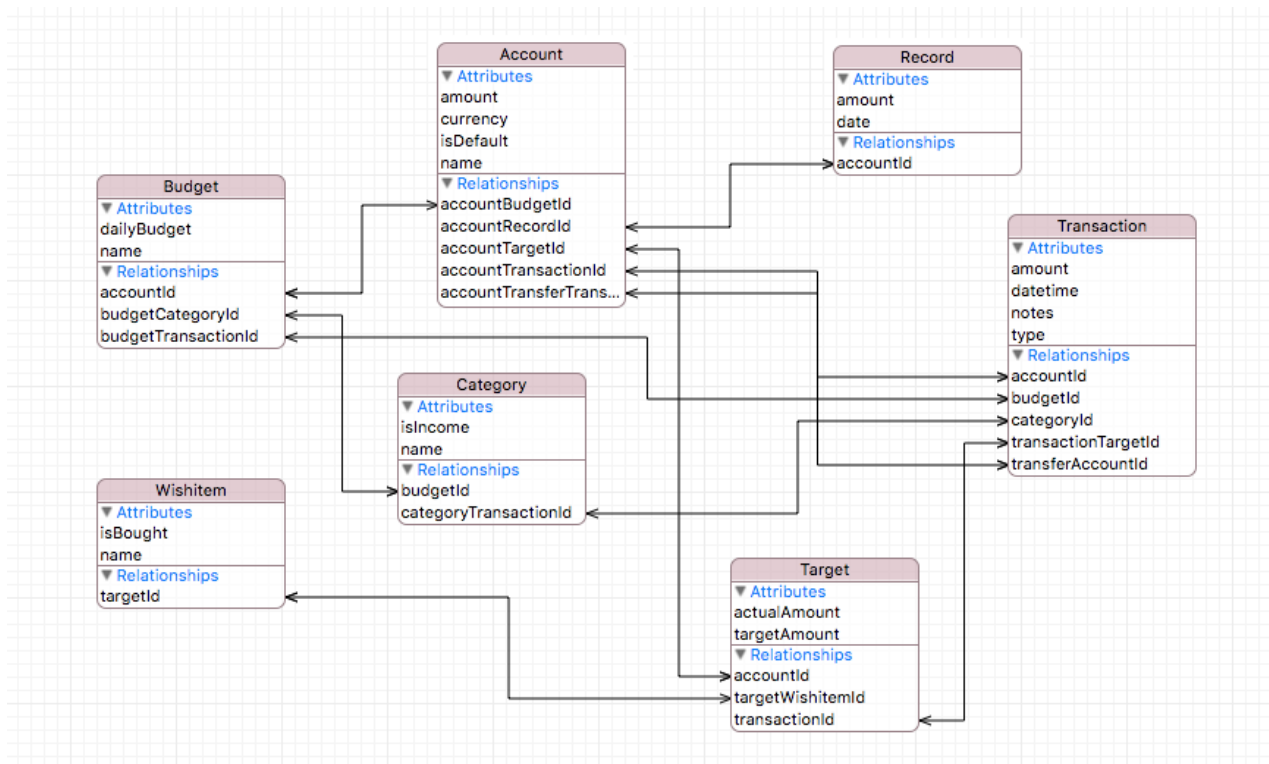
Use Case ID: ACC-0003-2

- **Title:** Delete Account
- **Description:** The user will delete the account, removing it from the core data.

- **Precondition:** There is at least one non-default account which is available for deletion.
- **Postcondition:** The non-default account is deleted, leaving one less account in the core data.
- **Main Success Scenario:** The delete button can be located in either the Edit Account or the View Account List. Once tapped, an alert will be displayed, warning them that the action is destructive and cannot be undone. The user then taps the delete button, confirming the deletion.
- **Extensions:** Once the confirmation alert is displayed, the user may tap the darkened area of the screen, or the cancel button to cancel the deletion.
- **Usage Frequency:** N/A

3 Database Schema

3.1 Core Data Diagram



3.2 Model Attributes

3.2.1 Account

- **amount** (*double*): Contains the current amount present in the account.
- **currency** (*string*): Contains the country identifier for the currency used in the account.
- **isDefault** (*boolean*): If this value is true, then the account is the default account to use for everyday transactions.
- **name** (*string*): The name of the account.

3.3 Model Relationships

3.3.1 Account

- **accountBudgetId** (*links with the Budget table*): Description to follow once implemented.
- **accountRecordId** (*links with the Record table*): Description to follow once implemented.
- **accountTargetId** (*links with the Target table*): Description to follow once implemented.
- **accountTransactionId** (*links with the Transaction table*): Description to follow once implemented.
- **accountTransferTransactionId** (*links with the Transaction table*): Description to follow once implemented.

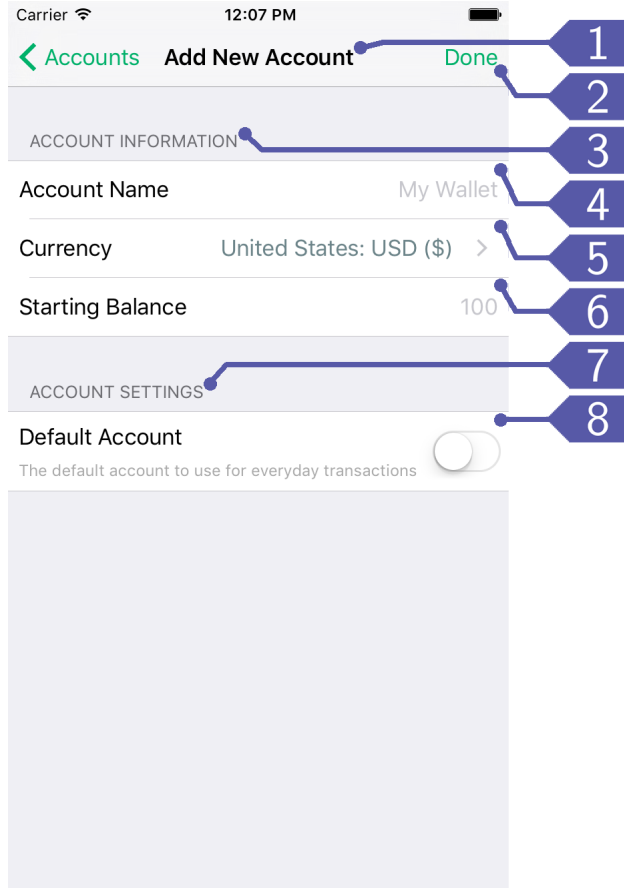
4 Application Screens

4.1 Adding a New Account

The Add New Account screen enables users to add a new account by specifying the account's name, currency used and starting balance. The user can also specify whether or not to use this account as the default account to use for everyday transactions.

4.1.1 Application Screenshots

1AddAccount-1



Screen Component	Type	Description
1. Screen Title	Label Title	Localization Key: MENULABEL_ADD_ACCOUNT
2. Done Button	Button	<p>When tapped, the app performs form validation and saves all the account information – the account name, currency, starting balance, and whether or not to use the account as default – into the database.</p> <p>For more details on the form validation, please see the Error Scenarios section.</p> <p>Localization Key: BUTTON_DONE</p>
3. Account Name Cell	Table Cell	<p>This table cell consists of two UI elements: A label and a textfield.</p> <p>Tap anywhere in the cell and the textfield will be placed in focus, bringing up a QWERTY keyboard. Tap anywhere outside the cell to dismiss the keyboard.</p> <p>The textfield has a character limit of 25 characters.</p> <p>Localization Keys: LABEL_NAME, TEXTFIELD_NAME_PLACEHOLDER</p>
4. Currency Cell	Table Cell	<p>This table cell consists of two UI Label elements.</p> <p>When this table cell is tapped, the cell will be briefly filled with a light green background, then fade out as the screen transitions to the Currency Selection screen. The default currency depends on the user's device locale.</p> <p>Localization Key: LABEL_CURRENCY</p>

Screen Component	Type	Description
5. Starting Balance Cell	Table Cell	<p>This table cell consists of two UI elements: A label and a textfield.</p> <p>Tap anywhere in the cell and the textfield will be placed in focus, bringing up a keyboard that is numeric with a decimal point. Tap anywhere outside the cell to dismiss the keyboard.</p> <p>The textfield has a character limit of 15 characters.</p> <p>Localization Keys: LABEL_STARTING_BALANCE, TEXTFIELD_STARTING_BALANCE_PLACEHOLDER</p>
6. Default Account Cell	Table Cell	<p>This table cell consists of three UI elements: two labels and a switch. The switch is turned off by default.</p> <p>Tap anywhere in the cell and the switch will be toggled. There can only be one default account at any time.</p> <p>Localization Keys: LABEL_IS_DEFAULT_ACCOUNT, LABEL_IS_DEFAULT_ACCOUNT_DESCRIPTION</p>
7. Error Alert	Alert Controller	<p>This alert controller has two labels.</p> <p>The alert is displayed when either the account name or the initial amount has not been filled, and the user taps "Done".</p> <p>Localization Keys: ERRORLABEL_ERROR_TITLE, ERRORLABEL_NAME_CURRENCY_NOT_EMPTY</p>

4.1.2 Error Scenarios

Title	Description
Duplicate Account Name	<p>This error is triggered when the Done button is pressed and a similar account name already exists in the Core Data.</p> <p>Localization Key: ERRORLABEL_DUPLICATE_ACCOUNT_NAME</p>
Required Fields Unfilled	<p>This error is triggered when the Done button is pressed and the account name or the amount has not been filled.</p> <p>Localization Key: ERRORLABEL_NAME_CURRENCY_NOT_EMPTY</p>
Multiple Decimals	<p>This check confirms that decimals can only be inputted once. It only exists in textfields that have numeric input.</p> <p>If a decimal is in the input, and the user keys in another decimal point, the textfield will not respond. There will be no error alert displayed.</p> <p>If the user pastes text that has two decimal points, then the textfield will not respond as well. Once again, there will be no error alert displayed.</p>
Maximum Length	<p>This check confirms that the length of the input text is less than or equal to a given number. It is present in all textfields.</p> <p>Once the maximum length is reached, the user will no longer be able to input any new character. No error alert is displayed.</p> <p>If the user pastes text that is longer than the maximum length, it will fail to paste and no error alert is displayed.</p>

4.2 Currency Selection

The currency selection screen enables users to search and select a currency to use for their new account. The user can search for the country's name, the currency name, or the currency symbol through the search bar. Once the user is satisfied with their selection, they can tap the Back button to return to the Add New Account screen, where they will see their newly selected currency in the Currency Cell (Item #4 of the Application Screenshots section).

4.2.1 Application Screenshots

Screen Component	Type	Description
1. Screen Title	Label Title	Localization Key: MENULABEL_CURRENCY_PICKER
2. Search Field	Search Bar	<p>When tapped, a QWERTY keyboard is displayed, and the search textfield is given focus.</p> <p>At the same time, the width of the textfield shortens, giving way to the Cancel button which, when tapped, dismisses the keyboard and returns it to the state shown in the screenshot.</p> <p>Note that because the search bar is placed at the header of the table, it will move together as the user scrolls and it is not fixed on the top of the screen.</p>
3. Loading Screen	Label	<p>This label is displayed while the currency data is being loaded.</p> <p>While loading, the user cannot scroll.</p> <p>Localization Key: GUIDELABEL_LOADING</p>
4. Currency Selection Cell	Table Cell	<p>This table cell consists of three UI labels showing the currency's country, code, and symbol.</p> <p>Tap anywhere in the cell and a checkmark will appear on its right side, indicating that a type of currency has been selected.</p>

4.3 Account List Display

The Account List screen displays the name and the current amount of the accounts that the user has added. It also displays a flag indicating whether or not an account has been marked as default.

It allows the user to delete an account, or set it as default by swiping the list item left. When an account item is tapped, the user is taken to a screen where the account details can be viewed and edited. Note that an account marked as default cannot be deleted, so when swiping left, the only available option is "Edit", which has the same functionality as tapping the account list item.

4.3.1 Application Screenshots

Screen Component	Type	Description
1. Screen Title	Label Title	Localization Key: MENULABEL_ACCOUNT
2. Add Button	Button	<p>When tapped, the screen transitions to the Add Account Screen (See section: Adding a New Account) if the number of accounts is less than 20.</p> <p>However, if the number of accounts has reached 20, it will display an error. (See section: Error Scenarios)</p>
3. Default Icon	View Element	<p>An indicator showing whether or not an account is marked as default. There can only be one default account at any one time.</p> <p>Localization Key: LABEL_DEFAULT</p>
4. Edit Button	Table Cell Button	<p>This button is displayed when the swiped cell is not a default account. When tapped, the screen transitions to the Edit Account Screen (See section: Editing an Existing Account)</p> <p>Localization Key: BUTTON_EDIT</p>
5. Account Name Label	Label	A label that displays the name of the account.
6. Amount Label	Label	A label that displays the account's currency and the amount.
7. Chevron Icon	Table cell icon	Indicates that when the entire cell can be tapped. When tapped, the screen transitions to the Edit Account Screen (See section: Editing an Existing Account)

Screen Component	Type	Description
8. Set Default Button	Table Cell Button	<p>This button is displayed when the swiped cell is as default account.</p> <p>When tapped, the cell will slide back and will be marked as default. The previously default cell will slide too, indicating that it was updated and unmarked.</p> <p>Localization Key: BUTTON_SET_DEFAULT</p>
9. Delete Button	Table Cell Button	<p>When tapped, the account is deleted and the cell slides away to disappear.</p> <p>Localization Key: BUTTON_DELETE</p>
10. No Accounts Guide	Label	<p>This label is displayed when there are no accounts present.</p> <p>This scenario theoretically should not happen under normal circumstances, as there is no way for the user to delete an account marked as default, and when the app is freshly installed, there will be one account available, and it is marked as default.</p> <p>Localization Key: GUIDELABEL_NO_ACCOUNTS</p>

4.3.2 Error Scenarios

Title	Description
Maximum Number of Accounts Reached	<p>This error is triggered when the Add a New Account button is pressed the number of accounts present in the core data has reached 20.</p> <p>Localization Key: ERRORLABEL_TOO_MANY_ACCOUNTS</p>

4.4 Editing an Existing Account

The Edit Account screen enables users to edit existing accounts that were previously added. Its layout is and functionality is similar to the Add Account screen, but users can now set an existing account as default, or delete an account through this screen. Note that accounts that have been marked as default cannot be deleted or set as default once again.

4.4.1 Application Screenshots

Screen Component	Type	Description
1. Screen Title	Label Title	Localization Key: MENULABEL_EDIT_ACCOUNT
2. Save Button	Button	This button has the same functionality as the Done Button (See section: Application Screenshots, Item #2), but it updates the core data instead of adding a new entry. Localization Key: BUTTON_SAVE
3. Current Balance Cell	Table Cell	This cell functions is similar to the Initial Balance cell in the Application Screenshots section, but it has its localization key changed. Localization Keys: LABEL_CURRENT_BALANCE
4. Disabled Default Account Button	Button	This button cannot be tapped, and it informs the user that the current account is marked as default. Localization Key: BUTTON_SET_AS_ACCOUNT_DISABLED
5. Disabled Delete Account Button	Button	This button cannot be tapped, and it informs the user that it is not possible to delete default accounts. Localization Key: BUTTON_DELETE_ACCOUNT_DISABLED

Screen Component	Type	Description
6. Set as Default Button	Button	<p>Once this button is tapped, it will mark the account as default, and will be disabled, turning into a state seen in the fourth item of this screen.</p> <p>Localization Key: BUTTON_SET_AS_DEFAULT</p>
7. Delete Button	Button	<p>Once this button is tapped, it will display a warning, telling the user that the action cannot be undone, as seen in the Deleting an Existing Account section.</p> <p>Localization Key: BUTTON_DELETE_ACCOUNT</p>

4.5 Deleting an Existing Account

This screen consists of a confirmation alert dialog, activated whenever an account is about to be deleted. Deleting an account will delete all transactions related to the account, and it is an action that cannot be undone. Hence, this alert dialog exists to confirm the action.

4.5.1 Application Screenshots

Screen Component	Type	Description
1. Cancellation Area	Darkened Area	Tap anywhere within the darkened area, and the alert controller will be dismissed.
2. Deletion Message	Alert Controller	<p>This alert controller is in the style of an action sheet. It displays the title and the message of the alert.</p> <p>Localization Keys: LABEL_WARNING_DELETE_ACCOUNT_TITLE, LABEL_WARNING_DELETE_ACCOUNT_MESSAGE</p>
3. Delete Alert Action	Alert Action	<p>Once tapped, this button will delete the selected account and its associated transactions from the core data.</p> <p>Localization Key: BUTTON_DELETE_ACCOUNT</p>
4. Cancel Alert Action	Alert Action	<p>Once tapped, the alert controller will be dismissed.</p> <p>Localization Key: BUTTON_CANCEL</p>

4.6 Adding a New Budget

The Add New Budget screen enables users to add a new budget by specifying the budget's name and its allotted amount within a specified frequency (that is, Monthly, Weekly, or Daily). The frequency can be selected from the Budget Screen (Section TBA). The user can also add categories within the Budget.

4.6.1 Application Screenshots

Screen Component	Type	Description
1. Screen Title	Label Title	Localization Key: MENULABEL_ADD_BUDGET
2. Done Button	Button	<p>When tapped, the app performs form validation and saves all the budget information – the budget name, and the budget amount – into the core data.</p> <p>For more details on the form validation, please see the Error Scenarios section.</p> <p>Localization Key: BUTTON_DONE</p>
3. Budget Name Cell	Table Cell	<p>This table cell consists of two UI elements: A label and a textfield.</p> <p>Tap anywhere in the cell and the textfield will be placed in focus, bringing up a QWERTY keyboard. Tap anywhere outside the cell to dismiss the keyboard.</p> <p>The textfield has a character limit of 25 characters.</p> <p>Localization Keys: LABEL_BUDGET_NAME, TEXTFIELD_BUDGET_PLACEHOLDER</p>
4. Budget Amount Cell	Table Cell	<p>This table cell consists of two UI elements: A label and a textfield.</p> <p>Tap anywhere in the cell and the textfield will be placed in focus, bringing up a keyboard that is numeric with a decimal point. Tap anywhere outside the cell to dismiss the keyboard.</p> <p>The label changes in response to what is selected in the Budgets Screen. If the user has chosen "Monthly" in the Budgets Screen, the label will show up as "Monthly Budget", "Weekly Budget" if "Weekly" was chosen, and "Daily Budget" if "Daily" was chosen. (Section X.Y)</p> <p>The textfield has a character limit of 15 characters.</p>

Screen Component	Type	Description
5. Category Cell	Table Cell	<p>This table cell consists of a label.</p> <p>Once the cell is tapped, an alert is brought up, allowing the user to edit the category's name, or delete it.</p> <p>The contents of these cells depend on user input in the "Add New Category"</p>
6. Add New Category Cell	Table Cell	<p>This table cell consists of a textfield.</p> <p>Tap anywhere in the cell and the textfield will be placed in focus, bringing up a QWERTY keyboard. Tap anywhere outside the cell to dismiss the keyboard.</p> <p>The textfield has a limit of 25 characters.</p> <p>Localization Keys: LABEL_IS_DEFAULT_ACCOUNT, LABEL_IS_DEFAULT_ACCOUNT_DESCRIPTION</p>
7. Error Alert	Alert Controller	<p>This alert controller has two labels.</p> <p>The alert is displayed when either the account name or the initial amount has not been filled, and the user taps "Done".</p> <p>Localization Keys: ERRORLABEL_ERROR_TITLE, ERRORLABEL_NAME_CURRENCY_NOT_EMPTY</p>

4.6.2 Error Scenarios

Title	Description
Duplicate Account Name	<p>This error is triggered when the Done button is pressed and a similar account name already exists in the Core Data.</p> <p>Localization Key: ERRORLABEL_DUPPLICATE_ACCOUNT_NAME</p>
Required Fields Unfilled	<p>This error is triggered when the Done button is pressed and the account name or the amount has not been filled.</p> <p>Localization Key: ERRORLABEL_NAME_CURRENCY_NOT_EMPTY</p>
Multiple Decimals	<p>This check confirms that decimals can only be inputted once. It only exists in textfields that have numeric input.</p> <p>If a decimal is in the input, and the user keys in another decimal point, the textfield will not respond. There will be no error alert displayed.</p> <p>If the user pastes text that has two decimal points, then the textfield will not respond as well. Once again, there will be no error alert displayed.</p>
Maximum Length	<p>This check confirms that the length of the input text is less than or equal to a given number. It is present in all textfields.</p> <p>Once the maximum length is reached, the user will no longer be able to input any new character. No error alert is displayed.</p> <p>If the user pastes text that is longer than the maximum length, it will fail to paste and no error alert is displayed.</p>

5 Design Specifications

This section contains information regarding visual properties of certain UI elements. Examples of visual properties include height, width, and relative distance to another UI element. The UI elements highlighted in this section may range from text fields, to labels, or to switches and progress bars.

Note that a light maroon line crossing right through the center of a parent UI element (e.g.: The Account Name table cell) indicates that it is divided into two equal portions. A darkened maroon line through a sub-element (e.g.: "Account Name" UI Label, "My Wallet" text field) indicates that it contains the property of being situated in the middle of the parent element.

Note that a vertical or horizontal line leaning against the inner wall of a parent element indicates the distance between the sub-element to the parent element's walls. For example, in the figure above, it can be seen that the "Account Name" UI Label leans against the wall of its table cell at a distance of zero points.

If an element does not have any measurement displayed, this indicates that the element's properties was determined by the operating system. Examples of these include the check mark seen in the currency selection screen, or the title and buttons in the navigation bar.

5.1 Add and Edit Account Screen

5.2 Currency Selection Screen

5.3 Account List Display Screen