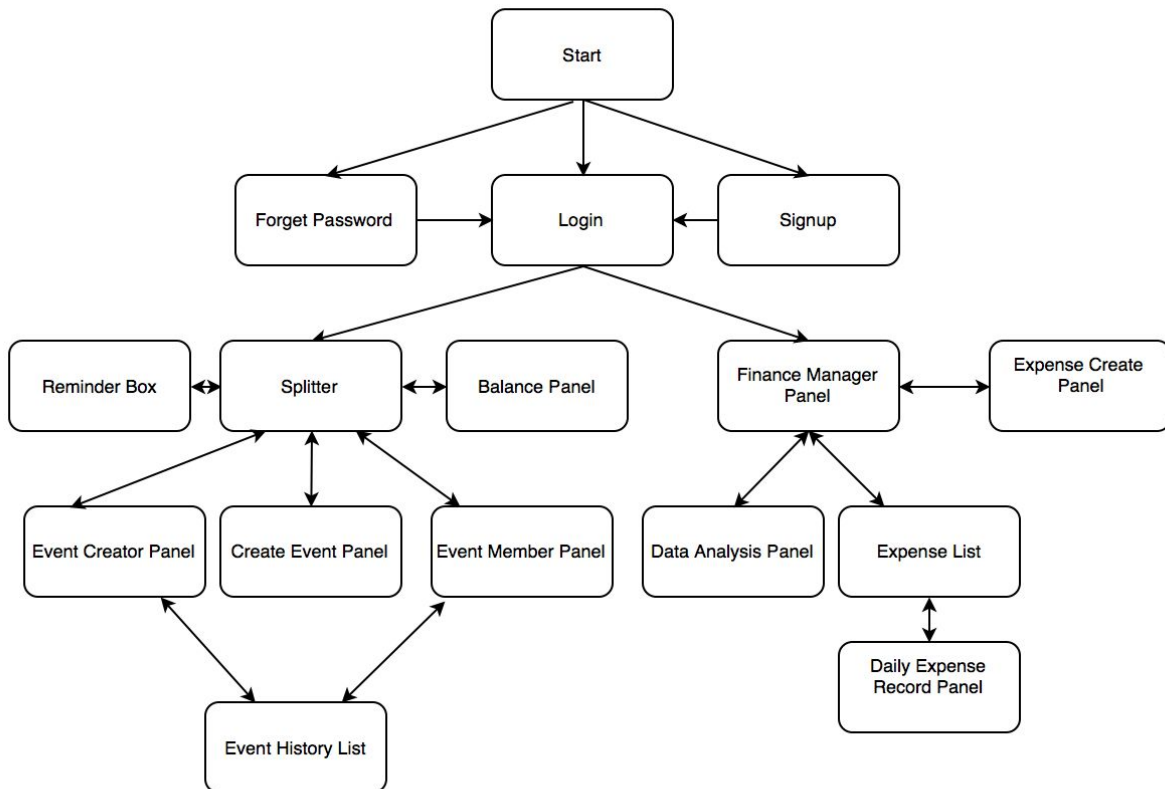


# Incremental and Regression Testing

## 1. Classification of Component

### 1.1 Component Structure



#### Start

- Input: None
- Output: None
- Parent Dependency: None
- Child Dependency: Forget Password, Login, Signup

#### Login:

- Input: email address, password
- Output: None
- Parent Dependency: Start, Forget Password, Signup
- Child Dependency: Splitter, Finance Manager Panel

#### Signup:

- Input: email address, password, username

- Output: None
- Parent Dependency: Start
- Child Dependency: Login

Forget Password:

- Input: email address, temporary password, new password
- Output: temporary password (sent to email entered)
- Parent Dependency: Start
- Child Dependency: Login

Splitter:

- Input: None
- Output: None
- Parent Dependency: Login, Reminder Box, Balance Panel, Event Creator Panel, Event Member Panel, Create Event Panel
- Child Dependency: Reminder Box, Balance Panel, Event Creator Panel, Event Member Panel, Create Event Panel

Balance Panel:

- Input: None
- Output: The amount of money who owe you and the amount of money you owe others
- Parent Dependency: Splitter
- Child Dependency: Splitter

Reminder Box:

- Input: email, money amount
- Output: email reminder
- Parent Dependency: Splitter
- Child Dependency: Splitter

Event Creator Panel:

- Input: Total amount
- Output: None
- Parent Dependency: Splitter
- Child Dependency: Splitter

Event Member Panel:

- Input: The amount you owe in this event
- Output: None
- Parent Dependency: Splitter
- Child Dependency: Splitter

Event History Panel:

- Input: None
- Output: None
- Parent Dependency: Event Creator Panel, Event Member Panel
- Child Dependency: Event Creator Panel, Event Member Panel

Create Event Panel:

- Input: Event name, split type, category, members
- Output: None

- Parent Dependency: Splitter
- Child Dependency: Splitter

Finance Management Panel:

- Input: None
- Output: None
- Parent Dependency: Login
- Child Dependency: Login, Data Analysis Panel, Expense List, Expense Create Panel

Data Analysis Panel:

- Input: None
- Output: Total expense, average expense and corresponding comparison; Top 3 categories; Data visualization.
- Parent Dependency: Finance Management Panel
- Child Dependency: Finance Management Panel

Expense List:

- Input: None
- Output: List of expense
- Parent Dependency: Finance Management Panel
- Child Dependency: Finance Management Panel

Expense Create Panel:

- Input: Category, Amount, Date, Note
- Output: None
- Parent Dependency: Finance Management Panel
- Child Dependency: Finance Management Panel

Daily Expense Record:

- Input: Category, Amount, Note
- Output: None
- Parent Dependency: Expense List
- Child Dependency: Expense List

## **1.2 Top-down incremental testing**

In this project, we followed top-down incremental testing. We tested modules from the top module where users signup or login the system to bottom modules as our application's sub functionalities. This makes it easier for us to discover the defects by following a proper control flow. This testing strategy ensured our major components work properly when we try to add new component.

## **2. Incremental Testing and Regression Testing**

Module	Create Event Panel		
Incremental Testing			
Defect #	Description	Severity	How to correct
1	When create an event. The event owner cannot be successfully stored in database	2	Add an attribute of OwnerID in the event_info schema
Regression Testing			
Defect #	Description	Severity	How to correct
1	After added the ownerID in the event_info schema. Only the event owner could be able to edit the event.	2	Check if a user is the event owner. If he/she is the owner than he/she will have the event creator page, if not, he/she will not have edit option.

Module	Event Creator Panel		
Incremental Testing			
Defect #	Description	Severity	How to correct
1	Only the event creator should be able to edit the event, but not all members	2	Check the ownerID attribute in event_info schema, If he/she is the owner than he/she will have the event creator page, if not, he/she will just have a normal user page.
Regression Testing			
Defect #	Description	Severity	How to correct
1	The “verify” button should not be able until all the member input their individual expense.	2	Create a showVerify flag, flag is true if the number of input equal to the number of members.

2	The “Split” button should not be able until the system verify that all the inputs are valid.	2	Create a showSplit flag, flag is true if the total amount of the member inputs matches with the owner input.
---	--	---	--

Module	Splitter Dashboard		
Incremental Testing			
Defect #	Description	Severity	How to correct
1	Made the component with proper size (length and height), but it does not seem to integrate well with other components on the page such as event history list, event creator/member panel, sidebar and header.	2	Designed the overall layout for the page by using bootstrap grid system. So we could put each component into its corresponding cell in the grid.
Regression Testing			
Defect #	Description	Severity	How to correct
1	Though every component looks properly by its size alone. But when putting into the responsive grid system, some components with absolute length defined tend to over crowd the cell.	2	Change all absolute length “px” to relative length using “rem” or “%”

Module	Balance Panel		
Incremental Testing			
Defect #	Description	Severity	How to correct
1	Able to load the component by inserting the component into its parent component(hard code), which should be responsive to the user event such as when user clicks on the balance button in	2	Add state called mainActivity into redux store. And display right component in the middle of the page based on the mainActivity state in MidContainer component.

	splitter dashboard		
Regression Testing			
Defect #	Description	Severity	How to correct
1	Adding mainActivity prop to balance component but forgetting to add this prop to other parent components that need use MidContainer such as event history list panel makes such logic only partially implemented	2	Add mainActivity prop to event history list panel as well to uniformly manage this state in the redux store

Module	Event history list module		
Incremental Testing			
Defect #	Description	Severity	How to correct
1	When there are too many events in the history, the scroller when scroll down, does not reach the bottom of the list. Although the last event can still be seen, this better be fixed for user experience	3	Adjust the parameter and the position of the scroller to fix the bug
Regression Testing			
Defect #	Description	Severity	How to correct
1	When an event is completed for a member, the event still stays on the member panel and will not be appearing in the event history list	1	Check code logic for event completion, possibly in the database the operation is not done.
2	In the member event panel, clicking ‘view my history’ button will not redirect the page to member’s history event list	1	Review the logic of the button ‘view my history’. See if the button is correctly linked to the event history list page.

Module	Event member panel		
Incremental Testing			
Defect #	Description	Severity	How to correct
1	When the event member panel is added to the application, part of its display overlapped with the balance panel	3	Fix the stylesheet to ensure the event member panel component appears on the position where it is supposed to appear
2	When a member is in so many events that the panel cannot hold all the events, the panel scroller on the side when scroll down will not reach bottom of the event list	2	Adjust the UI code to fix the scroller component.
Regression Testing			
Defect #	Description	Severity	How to correct
1	When a member pays off his amount in an event, even the event status is showing as “completed”, the event still stays on the event list of the member panel	1	Check the code logic for the event completion and also check if the member is removed from the event member list in the database.