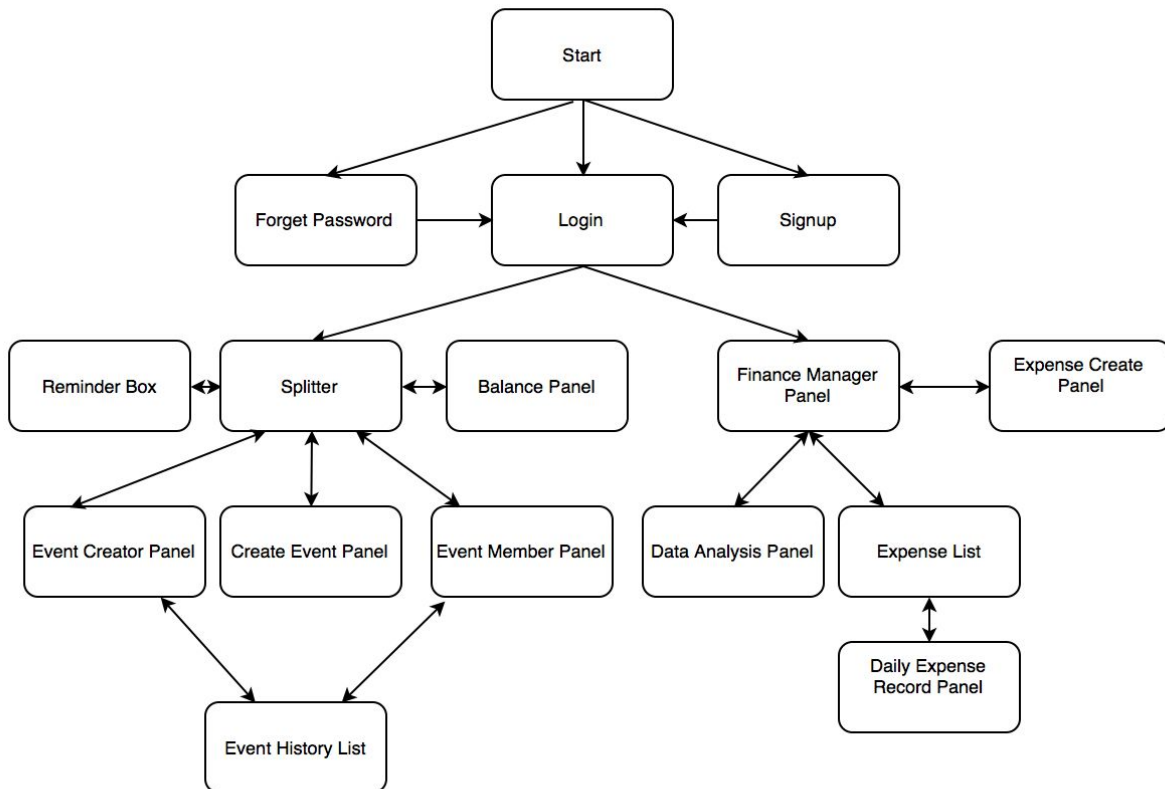


Incremental and Regression Testing

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	Event History List		
Incremental Testing			
Defect #	Description	Severity	How to correct
1	Not able to display events by its status as ongoing or finished	2	Add status property in event schema in database. Set status of all new created events as ongoing by default
Regression Testing			
Defect #	Description	Severity	How to correct
1	After an event’s status changes from ongoing to finished, the event should not show up under ongoing event list and should show up in finished event list	2	Update the status of the event to finished when its status changes

Module	Event Creator Panel		
Incremental Testing			
Defect #	Description	Severity	How to correct
1	Only event creator can have access to “Edit” button to edit the setup info for an event	2	Check the entitlement of the user before rendering the buttons available at the top of the panel
2	Only after each member inputs their individual expense would there be a “Verify” button on the top of the panel	2	Set showVerify flag to true in database when number of inputs match number of members in the event
3	Should reset every member’s individual expense input, including the event creator’s total amount if system finds the	2	Reset all inputs of each member for this event receipt as 0 in the database as well as number of confirmed member as 0, which

	event receipt is invalid after the event creator hit “Verify” button		marks current number of users who has input their individual expense.
Regression Testing			
Defect #	Description	Severity	How to correct
1	If the input of the event receipt is valid, “Split” button should show up and “Verify” button should disappear	2	Render all available buttons on the top of the panel based on the current status of the event

Module	Event Member Panel		
Incremental Testing			
Defect #	Description	Severity	How to correct
1	Event member should also have ability to see the total amount expense in case the event creator input it wrong	2	Remove the condition check on user entitlement for an event when showing the bill sum information
Regression Testing			
Defect #	Description	Severity	How to correct
1	Event member should not be able to see the available buttons only event creator could view	2	Make a separate condition check on user entitlement for buttons that show on the top panel.

Module	Sign up module		
Incremental Testing			
Defect #	Description	Severity	How to correct
1	Before a new user object can be created, the server needs to perform email address validation to check if the format of the input email address is correct.	2	Add email address validation function to check if email address has correct format. Possible techniques could be regular expression.
Regression Testing			
Defect #	Description	Severity	How to correct
1	After changing the field ‘username’ to ‘email’ in the user model schema, backend database should be able to store user information with respect to email address and hashed password rather than username and password. However, now the database can only store one user object. Error ‘duplicate key’ is given when attempt to store more user objects.	1	Manually drop the index for ‘username’ field for the database to properly work. Used empty function to simulate unimplemented functions such as isEmailValid()

Module	Login module		
Incremental Testing			
Defect #	Description	Severity	How to correct
1	When an email address used to log in does not exist in the database, the server should return null json object and display error message. However, currently, the server will fall through and perform password matching check and then crash.	1	Add return statement under condition where user email address does not exist in the database to prevent from falling through.

Regression Testing			
Defect #	Description	Severity	How to correct
1	When server performs password check, it should compare the hashed version of input password to the stored password since unencoded password is not stored in the database	1	Apply hash() function to the input password when performing password check.
2	Sending GET request to /all_useremail return empty list even when there are users in the database.	2	Currently not yet fixed. Will look into the route and identify problems

Module	Forget password module(reset password module)		
Incremental Testing			
Defect #	Description	Severity	How to correct
1	After the user reset his/her password, the original password should be modified to new password instead of creating a new user object with the new password	2	After user input new password, server access the database directly modifies the old password.
Regression Testing			
Defect #	Description	Severity	How to correct
1	Currently there is no validation of email to which the reset password link will be sent.	1	Used the same function used in sign up module to handle email address validation

Module	Reminder Box		
Incremental Testing			
Defect #	Description	Severity	How to correct
1	After sending the reminder message, the text in textfield still remain.	2	Added a function to empty text area every time when the send button is hit.
Regression Testing			
Defect #	Description	Severity	How to correct
1	Member cannot retrieve previous messages sent by others	2	Save all previous message to database.

Module	Start module		
Incremental Testing			
Defect #	Description	Severity	How to correct
1	Wrong reference path for bundle.js file in index.html	1	Change its relative path “../public/bundle.js” to “bundle.js” since the compiled frontend files are put in same output directory called public(now called dist)
Regression Testing			
Defect #	Description	Severity	How to correct
1	Imported “Component” from React DOM in an improper way	2	Fix it by first “ import ReactDOM from 'react-dom' “, then ReactDOM.Component to reference Component