Test plan for AdventureAudit

Note that you can refine your testing plan as the project development goes. Keep the change log as follow:

ChangeLog

Version	Change Date	By	Description
1.0.0	Feb 17, 2023	Taeho, Suhjin, Leo, Luke	Initial Version (Sprint 2)
1.0.1	Mar 14, 2023	Taeho, Suhjin, Leo, Luke	Added acceptance testing and feature testing for Sprint 3
1.0.2	Mar 31, 2023	Taeho, Suhjin, Leo, Luke	Added loading testing and fixed bugs for Sprint 4
1.0.3	Apr 16, 2023	Taeho, Suhjin, Leo, Luke	Update test plan

1 Introduction

1.1 Scope

Tested Controller in the backend (Server)

- UserController
- TripController
- TransactionController
- CategoryController
- ReportController

- CountryController
- PayerController
- GroupController
- InviteController

By Sprint 2, the following features are going to be tested.

1. **Expense/payment tracking**
    a. Users should be able to create a new trip.
    b. Users should be able to see a list of trips they have created.
    c. Users should be able to delete a selected trip.
    d. When creating a new trip, users should be able to select a destination from a list of countries.
    e. Users should be able to create a new transaction for a selected trip.
    f. When creating a new transaction, users should be able to categorize their expenses from seven default categories.
2. **Trip Report**
    a. Users should be able to see their total expenses for a selected trip.

By Sprint 3, the following features are going to be tested.

1. **Expense/payment tracking**
    a. Users should be able to add custom categories.
    b. Users should be able to change their home country.
    c. Users should be able to see the budget in two currencies (Home and destination).
2. **Trip Report**
    a. Users should be able to see their transactions in a pie chart by category.
    b. Users should be able to see their top 5 most expensive transactions.
    c. Users should be able to see how much they owe to others and who owes them.
3. **Trip Sharing**
    a. Users should be able to invite other users to the trip.
    b. Users should be able to select the other users that users paid for.
    c. Users should be able to see how much they owe to others and who owes them.

1.2 Roles and Responsibilities

Name	Net ID	GitHub username	Role
Taeho Choi	Choit1	Taehoya	Developer
Suhjin Kang	Kangs2	Skang9810	Developer
Leonardo Warsito	Warsitol	Leonw00	Developer, QA Analyst
Luke Lepa	lepal	Luke-Lepa	Developer

2 Test Methodology

2.1 Test Levels

Test Levels define the Types of Testing to be executed on the Application Under Test (AUT). In this course, unit testing, integration testing, acceptance testing, regression testing, and load testing are mandatory.

Core Feature 1: Expense Tracking
API:
GET /trip/:trip_id
GET /trip/user/:user_id
POST /trip/:trip_id
DELETE /trip/:trip_id

GET /transaction/trip/:trip_id
POST /transaction
DELETE /transaction/trip_id

GET /country

GET /category/trip/:trip_id
POST /category
DELETE /category/:category_id

Unit Test

1. **Model**
    - Ensure that the user can get the list of trips from the database.
    - Ensure that the user can get detailed trip information from the database.
    - Ensure that when the user creates a new trip, a row is added to the trip table.
    - Ensure that the user can delete a trip row from a database.
    - Ensure that the user can get the list of transactions of a specific trip from the database.
    - Ensure that the user can delete a transaction from the database.
    - Ensure that when the user creates a new transaction, a row is added to the transaction table.
    - Ensure that the user can get detailed transaction information from the database.
    - Ensure that the users can get the list of categories of trips from the database.
    - Ensure that when the user creates a new category, a row is added to the category table.
    - Ensure that the user can delete a category row from a database.
2. **Controller**
    - Ensure that the user can create a new trip by successfully getting the HTTP response 200 from the request.

- Ensure that the user can get the list of their trips by successfully getting the HTTP response 200 from the request.
- Ensure that the user can delete a trip by successfully getting the HTTP response 200 from the request.
- Ensure that the user can get detailed information about their trip by successfully getting the HTTP response 200 from the request.
- Ensure that the user can add a new transaction to the selected trip by successfully getting the HTTP response 200 from the request.
- Ensure that the user can create a new category by successfully getting the HTTP response 200 from the request.
- Ensure that the user can delete a category by successfully getting the HTTP response 200 from the request.
- Ensure that the user can see a list of categories by successfully getting the HTTP response 200 from the request.
- Ensure that all controllers return the HTTP response 400 when the user did not specify the user_id and email and provided an invalid type.

Integration Test

- Ensure that the user can create a new trip when they are signed in by successfully getting the HTTP response 200 from the request.
- Ensure that the user can get the list of their trips when they are signed in by successfully getting the HTTP response 200 from the request.
- Ensure that the user can delete a trip by successfully getting the HTTP response 200 from the request.
- Ensure that the user can get detailed information about their trip by successfully getting the HTTP response 200 from the request.
- Ensure that the user can add a new transaction to the selected trip by successfully getting the HTTP response 200 from the request.
- Ensure that the user can create a new category by successfully getting the HTTP response 200 from the request.
- Ensure that the user can delete a category by successfully getting the HTTP response 200 from the request.
- Ensure that the user can see a list of categories by successfully getting the HTTP response 200 from the request.
- Ensure that all controllers return the HTTP response 400 when an error or issue occurs within the controller and/or model.

Core Feature 2: Report
API:
GET /report/sum
GET /report/category
GET /report/debt
GET /report/transaction

Unit Test

1. **Model**
    - Ensure that the user can get the total expense from the database.
    - Ensure that the user can get how much they spent for each category from the database.
    - Ensure that the user can get the top 5 expensive transactions from the database.
    - Ensure that the user can get how much the user owes to others and who owes the user money from the database.
2. **Controller**
    - Ensure that the user can get the total expense by successfully getting the HTTP response 200 from the request.
    - Ensure that the user can get how much they spent for each category by successfully getting the HTTP response 200 from the request.
    - Ensure that the user can get the top 5 expensive transactions by successfully getting the HTTP response 200 from the request.
    - Ensure that the user can get how much the user owes to others and who owes the user money by successfully getting the HTTP response 200 from the request.
    - Ensure that the user gets HTTP response 400 from the request when the user did not specify the trip_id and user_id and provided an invalid type.

Integration Test

- Ensure that the user can get the total expense by successfully getting the HTTP response 200 from the request.
- Ensure that the user can get how much they spent for each category by successfully getting the HTTP response 200 from the request.
- Ensure that the user can get the top 5 expensive transactions by successfully getting the HTTP response 200 from the request.
- Ensure that the user can get how much the user owes to others and who owes the user money by successfully getting the HTTP response 200 from the request.
- Ensure that the user gets HTTP response 400 from the request when the user did not specify the trip_id and user_id and provided an invalid type.

Core Feature 3: Trip Sharing
API:
POST /group
DELETE /group/trip/:trip_id/user/:user_id

POST /payer
DELETE /payer/transaction/:transaction_id/user/:user_id

POST /invite
DELETE /invite/trip/:trip_id/user/:user_id

GET /user/group/:trip_id

GET /user/payer/:transaction_id

Unit Test

1. **Model**
   - Ensure that the user can add a new row to the group table.
   - Ensure that the user can delete a row from the group table.
   - Ensure that the user can add a new row to the payer table.
   - Ensure that the user can delete a row from the payer table.
   - Ensure that the user can add a new row to the invite table.
   - Ensure that the user can delete a row from the invite table.
   - Ensure that the user can get the users that part of a trip.
   - Ensure that the user can get the users that paid for a transaction.
2. **Controller**
   - Ensure that the user can add a new group with HTTP response 200 from the request.
   - Ensure that the user can delete a row from the group table with HTTP response 200 from the request.
   - Ensure that the user can add a related payer for a transaction with HTTP response 200 from the request.
   - Ensure that the user can invite the users to trips with HTTP response 200 from the request.
   - Ensure that the user can decline the invitation for a trip with HTTP response 200 from the request.
   - Ensure that the user can get the users that part of a trip with HTTP response 200 from the request.
   - Ensure that the user can get the users that paid for a transaction with HTTP response 200 from the request.
   - Ensure that the user gets HTTP response 400 from the request when the user did not specify the trip_id and user_id and provided an invalid type.

Integration Test

- Ensure that the user can add a new group with HTTP response 200 from the request.
- Ensure that the user can delete a row from the group table with HTTP response 200 from the request.
- Ensure that the user can add a related payer for a transaction with HTTP response 200 from the request.
- Ensure that the user can invite the users to trips with HTTP response 200 from the request.
- Ensure that the user can decline the invitation for a trip with HTTP response 200 from the request.
- Ensure that the user can get the users that part of a trip with HTTP response 200 from the request.
- Ensure that the user can get the users that paid for a transaction with HTTP response 200 from the request.

- Ensure that the user gets HTTP response 400 from the request when the user did not specify the trip_id and user_id and provided an invalid type.
- **Regression Test**
  - Execute all above unit tests + integration tests you have for each commit pushed to the main branch.
  - https://github.com/Taehoya/Adventure-Audit/actions/runs/4200420925/jobs/7286413113

Acceptance Test
Sprint 2:
https://drive.google.com/file/d/19A16KW-m05hzYsUIa2cRmu5w-7GlXQcg/view?usp=sharing
Sprint 3:
https://drive.google.com/drive/folders/1_-N49WdWNOmuWQwROAuYIr0R2nzChToe?usp=sharing
Sprint 4:
https://drive.google.com/drive/folders/1D7mtNAO0H6wunMGHCVjzHI_EoeZY6irq?usp=sharing

User story 1: As a traveller, I want to be able to create a trip so that I can record my transactions.
Test:
1. Click on “Sign in with Google” on the centre of the screen
2. Sign in using the Google account
3. Click on “+ New Trip” button on the bottom of the sidebar (left bottom corner of the screen)
4. Fill in the new trip form:
   a. On the very top, type in the trip title: “Hawaii with friends”
   b. Select start date: “2023-06-01”
   c. Select end date: “2023-06-22”
   d. Select destination/country from drop down: “United States”
   e. Type in the budget field: “1000”
5. Click on the “Save” button on the bottom right corner
6. Click on the “Hawaii with friends” button on the left sidebar under “UPCOMING”

Expected Result:
1. All the fields in the new trip form should be field before being able to save the trip
2. If any field is not filled, it should give an error
3. The end date should not allow the user to fill in dates that are prior to the start date
4. Clicking on the “Cancel” button should not save the trip information (should not show up on the Sidebar)
5. Clicking on the “Save” button should save the trip information as filled and show up on the sidebar as a button
6. Clicking on the “Hawaii with friends” button should display the trip details beside the sidebar

User story 2: As a traveller, I want to be able to view a list of all my trips so that I can access a specific trip.

Expected condition:
1. At least one trip added under each “CURRENT”, “UPCOMING” and “PAST”

Test:
1. Click on “Sign in with Google” on the centre of the screen
2. Sign in using the Google account
3. Click on the “Hawaii with friends” button on the left sidebar under “UPCOMING”
4. Click on the “Trip to Paris” button on the left sidebar under “CURRENT”
5. Click on the “Edmonton” button on the left sidebar under “PAST”

Expected Result:
1. Clicking on each of the trip buttons should display the correct trip details beside the sidebar accordingly

User story 3: As a traveller, I want to be able to enter my budget for a trip so that I know how much I can spend and not overspend.

Expected condition:
1. Logged in successfully

Test:
1. Click on “+ New Trip” button on the bottom of the sidebar (left bottom corner of the screen)
2. Fill in the new trip form:
    a. On the very top, type in the trip title: “Winnipeg”
    b. Select start date: “2023-01-01”
    c. Select end date: “2023-01-22”
    d. Select destination/country from drop down: “Canada”
    e. Type in the budget field: “500”
3. Click on the “Save” button on the bottom right corner
4. Click on the “Winnipeg” button on the left sidebar under “PAST”

Expected Result:
1. Clicking on the “Save” button should save the trip information as filled and show up on the sidebar as a button
2. Clicking on the “Winnipeg” button should display the budget under “Budget” as “$ 500.00”

User story 4: As a traveller, I want to be able to add transactions to a trip so that I can record what I have spent during the trip.

Expected condition:
1. Logged in successfully
2. Trip added with the following information:
    a. Trip title: “Hawaii with friends”
    b. Start date: “2023-06-01”
    c. End date: “2023-06-22”
    d. Destination: “United States”
    e. Budget: “1000”

Test:
1. Click on the “Hawaii with friends” button on the left sidebar under “UPCOMING”
2. Click on the “+” button on the top right corner of the screen

3. Fill in the form from top to bottom in order:
    a. Item details: “Breakfast”
    b. Amount: “10”
    c. Date: “2023-06-01 7:40 AM”
    d. Category: Click on the first icon button (top left corner)
    e. Notes: at McDonalds
4. Click on the “Save” button on the bottom left corner of the form

Expected result:

1. Under “Transactions” tab in the middle of the screen under the trip information, there should be the transaction showing up

User story 5: As a traveller, I want to be able to categorize my transactions so that I can get a better understanding of how I’m spending my money.

Expected condition:

1. Logged in successfully
2. Trip added with the following information:
    a. Trip title: “Hawaii with friends”
    b. Start date: “2023-06-01”
    c. End date: “2023-06-22”
    d. Destination: “United States”
    e. Budget: “1000”

Test:

1. Click on the “Hawaii with friends” button on the left sidebar under “UPCOMING”
2. Click on the “+” button in the top right corner of the screen
3. Fill in the form from top to bottom in order:
    a. Item details: “Hotel”
    b. Amount: “250”
    c. Date: “2023-06-01 10:50 AM”
    d. Category: 
    e. Leave the Notes empty
4. Click on the “Save” button in the bottom left corner of the form

Expected result:

1. Under “Transactions” tab in the middle of the screen under the trip information, the added transaction “Hotel”
2. The “Hotel” transaction should display the category icon with the pink background

User story 6: As a traveller, I want to be able to view all my transactions of a trip, so that I can view a record of my spending.

Expected condition:

1. Logged in successfully
2. Trip added with the following information:
    a. Trip title: “Hawaii with friends”
    b. Start date: “2023-06-01”
    c. End date: “2023-06-22”
    d. Destination: “United States”
    e. Budget: “1000”

3. One transaction added in the “Hawaii with friends” trip with the following information:
    a. Item details: “Breakfast”
    b. Amount: “10”
    c. Date: “2023-06-01 7:40 AM”
    d. Category: first icon button (top left corner)
    e. Notes: at McDonalds
4. Second transaction added in the “Hawaii with friends” trip with the following information:
    a. Item details: “Hotel”
    b. Amount: “250”
    c. Date: “2023-06-01 10:50 AM”
    d. Category: 

Test:
1. Click on the “Hawaii with friends” button on the left sidebar under “UPCOMING”
2. Click on each of the transactions to see the transaction details

Expected result:
1. Under “Transactions” tab in the middle of the screen under the trip information, there should be the 2 transactions showing up accordingly
2. Clicking on each of the transactions would show the details on the 3rd column

User story 7: As a traveller, I want to be able to see how much the entire trip cost so that I can learn about how much it took to travel.

Expected condition:
1. Logged in successfully
2. Trip added with the following information:
    a. Trip title: “Hawaii with friends”
    b. Start date: “2023-06-01”
    c. End date: “2023-06-22”
    d. Destination: “United States”
    e. Budget: “1000”
3. One transaction added in the “Hawaii with friends” trip with the following information:
    a. Item details: “Breakfast”
    b. Amount: “10”
    c. Date: “2023-06-01 7:40 AM”
    d. Category: first icon button (top left corner)
    e. Notes: at McDonalds
4. Second transaction added in the “Hawaii with friends” trip with the following information:
    a. Item details: “Hotel”
    b. Amount: “250”
    c. Date: “2023-06-01 10:50 AM”
    d. Category: 

Test:
1. Click on the “Hawaii with friends” button on the left sidebar under “UPCOMING”

Expected result:

1. Under the “Total Expense” in the middle of the screen under the trip title and trip date, the total expense should display “260”

User story 8: As a traveller, I want to delete the trip information so that I can remove the trip event that I no longer want to keep a record of.
Expected condition:
1. Logged in successfully
2. Trip added with the following information:
    a. Trip title: “Edmonton”
    b. Start date: “2022-04-28”
    c. End date: “2022-04-30”
    d. Destination: “Canada”
    e. Budget: “500”

Test:
1. Click on the “Edmonton” button on the left sidebar under “PAST”
2. Click on the  button on the right side of the trip title
3. Click on “Cancel” button in the bottom right side of the modal
4. Click on the  button on the right side of the trip title
5. Click on the “Delete” button in the bottom left side of the modal
6. Click on the “Delete” button for the confirmation message

Expected result:
1. Clicking on the “Cancel” button should not do anything and still display the trip information details on the screen
2. Clicking on the “Delete” button on the alert window should refresh the page and the sidebar on the left should not have the trip button with the name “Edmonton” under “PAST”

User story 9: As a traveller, I want to be able to record transactions in different currencies to make documenting foreign purchases more accurate
Expected condition:
1. Logged in successfully
2. Trip added with the following information:
    a. Trip title: “Hawaii with friends”
    b. Start date: “2023-06-01”
    c. End date: “2023-06-22”
    d. Destination: “United States”
    e. Budget: “1000”

Test:
1. Click on the “Hawaii with friends” button on the left sidebar under “UPCOMING”
2. Click on the edit button in the top right corner of the 2nd column of the page next to the title of the trip.
3. Edit the trip information with the following information:
    a. Trip title: “Japan with friends”
    b. Destination: “Japan”
4. Click the “Save” button

5. Click on the “+” button in the top right corner of the screen
6. Fill in the form from top to bottom in order:
    a. Item details: “Hotel”
    b. Amount: “250”
    c. Date: “2023-06-01 10:50 AM”
    d. Category: Click on 
    e. Leave the Notes empty
7. Click on the “Save” button in the bottom left corner of the form

Expected result:

1. The trip title should change to “Japan with friends” instead of “Hawaii with friends”
2. The “Destination” and “Currency” section should be updated to “Japan” and “¥” accordingly
3. All “$” should be replaced by “¥”
4. The entry for putting in the amount of a transaction should be in “¥”
5. New transaction that was added should show the amount in “¥”

User story 10: As a traveller, I want to be able to add custom categories so that I can personalize my transactions to have a more detailed record

Expected condition:

1. Logged in successfully
2. Trip added with the following information:
    a. Trip title: “Japan with friends”
    b. Start date: “2023-06-01”
    c. End date: “2023-06-22”
    d. Destination: “Japan”
    e. Budget: “100000”

Test:

1. Click on the “Japan with friends” button on the left sidebar under “UPCOMING”
2. Click on the  button in the top right corner of the screen
3. Click on the  button in the top right corner of the screen
4. Click “Add” button in the bottom right corner
5. Type “Coffee” for the category name.
6. Click “Add” button in the bottom right corner
7. Choose the icon : 
8. Select colour:
    a. Drag the slider to any colour
    b. Click on the very first block of colour beneath the slider
9. Click “Add” button in the bottom right corner right
10. Click on the  button in the top right corner of the screen
11. Fill in the form from top to bottom in order:
    a. Item details: “Coffee”
    b. Amount: “2500”

c. Date: “2023-06-03 7:00 PM”
d. Category: Click on the newly added category
e. Leave the Notes empty
12. Click on the “Save” button in the bottom left corner of the form

Expected result:
1. New categories with the selected icon and colour should show up in the list of categories when successfully added.
2. Trying to click “Add” should give an error about the empty entries except for the color
3. The color is optional to select. If not selected, the color of the category will stay white.
4. Newly added transaction with the custom category should appear in the transaction list

User story 11: As a traveller, I want to be able to invite other users to trips so that I can document transactions with others

Expected condition:
1. Logged in successfully
2. Trip added with the following information:
    a. Trip title: “Japan with friends”
    b. Start date: “2023-06-01”
    c. End date: “2023-06-22”
    d. Destination: “Japan”
    e. Budget: “100000”

Test:
1. Click on the “Japan with friends” button on the left sidebar under “UPCOMING”
2. Click on the  button in the top right corner of the screen
3. Click the “Invite” button underneath the input field for “Invite”
4. Type in “kangsuhjingmail.com” in the input field under “Invite”
5. Click the “Invite” button underneath the input field
6. Type in “kang@gmail.com” in the input field under “Invite”
7. Click the “Invite” button underneath the input field
8. Type in “kangsuhjin@gmail.com” in the input field under “Invite”
9. Type in “suhjin9810@gmail.com” in the input field under “Invite”
10. Logout by clicking the Profile avatar and clicking on the “Logout” button in the top right corner
11. Login with “kangsuhjin@gmail.com” account
12. Click on the profile button
13. Click on the “Accept” button with the invite from “Suhjin Kang” with trip name “Japan with friends”
14. Click “x” button on the top right corner
15. Click on the “Japan with friends” button on the left sidebar under “UPCOMING”
16. Click on the  button in the top right corner of the screen

Expected result:
1. The first time clicking the “Invite” button should give an error saying “Please fill out this field”

2. The second input should give an error saying “Please include an ‘@’ in the email address…”
3. The third input should give an error saying “There is no user with that email”
4. The fourth input should successfully show an invite on the Profile window.
5. The fifth input should give an error saying “That user is already in this group or has a pending invite”
6. Logging into “kangsuhjin@gmail.com” account and accepting the invite should successfully show the “Japan with friends” trip and show both users’ names (Suhjin Kang & Lydia) in the “Collaboration” tab

User story 12: As a transaction payer, when I add a transaction, I want to be able to select the other users that I paid for, so that we can divide the cost later

Expected condition:

1. Logged in successfully
2. Trip added with the following information:
    a. Trip title: “Japan with friends”
    b. Start date: “2023-06-01”
    c. End date: “2023-06-22”
    d. Destination: “Japan”
    e. Budget: “100000”
    f. There are the following members part of the trip “Japan with friends”
        i. “Suhjin Kang”
        ii. “Luke Lepa”
        iii. “Taeho Choi”
        iv. “Leonardo Warsito”

Test:

1. Click on the “Japan with friends” button on the left sidebar under “UPCOMING”
2. Click on the  button in the top right corner of the screen
3. Click on the “+” button in the top right corner of the screen
4. Fill in the form from top to bottom in order:
    a. Item details: “breakfast”
    b. Amount: “90”
    c. Date: “2023-06-01 8:30 AM”
    d. Category: first icon button (top left corner)
    e. Leave the Notes empty
    f. Related members: “Suhjin Kang”
    g. Who paid?: “Suhjin Kang”
    h. Related members: “Luke Lepa”, “Taeho Choi”
5. Click on the “Save” button in the bottom left corner of the form
6. Click on the transaction “breakfast”

Expected result:

1. Step 4-g should uncheck “Suhjin Kang” from the “Related Members” and disabled from checking the checkbox.
2. A new transaction should show up on the transactions section with the name “breakfast” which the column of “Payers” with 3 profile avatars representing the 3 users

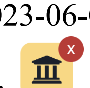
3. Clicking on the transaction should show the details of the transaction on the right side of the screen, which should also show the 3 users under “Payers:”

User story 13: As a group traveller, I want to see how much money I owe to others and who owes me money, so that everyone’s debts are documented

Expected condition:

1. Logged in successfully
2. Trip added with the following information:
    a. Trip title: “Japan with friends”
    b. Start date: “2023-06-01”
    c. End date: “2023-06-22”
    d. Destination: “Japan”
    e. Budget: “100000”
    f. There are the following members part of the trip “Japan with friends”
        i. “Suhjin Kang”
        ii. “Luke Lepa”
        iii. “Taeho Choi”
        iv. “Leonardo Warsito”
3. Transaction added with the following information:
    a. Item details: “breakfast”
    b. Amount: “90”
    c. Date: “2023-06-01 8:30 AM”
    d. Category: first icon button (top left corner)
    e. Leave the Notes empty
    f. Who paid?: “Suhjin Kang”
    g. Related members: “Luke Lepa”, “Taeho Choi”

Test:

1. Click on the “Japan with friends” button on the left sidebar under “UPCOMING”
2. Click on the “+” button in the top right corner of the screen
3. Fill in the form from top to bottom in order:
    a. Item details: “Ticket to museum”
    b. Amount: “100”
    c. Date: “2023-06-05 10:00 AM”
    d. Category: 

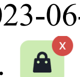
    e. Leave the Notes empty
    f. Who paid?: “Leonardo Warsito”
    g. Related members: “Suhjin Kang”
4. Click on the “Save” button in the bottom left corner of the form
5. Click on the  button top of the 3rd column.

Expected result:

1. Under “Payment Summary” section, it should display three rows (no significance in order):
    a. You ← Luke Lepa ¥30
    b. You ← Taeho Choi ¥30
    c. You → Leonardo Warsito ¥50

User story 14: As a traveller, I want to be able to see my top 5 most expense transactions so I can see what activities/services cost the most during a trip

Expected condition:

1. Logged in successfully
2. Trip added with the following information:
    a. Trip title: “Japan with friends”
    b. Start date: “2023-06-01”
    c. End date: “2023-06-22”
    d. Destination: “Japan”
    e. Budget: “100000”
    f. There are the following members part of the trip “Japan with friends”
        i. “Suhjin Kang”
        ii. “Luke Lepa”
        iii. “Taeho Choi”
        iv. “Leonardo Warsito”
3. Transaction added with the following information:
    a. Item details: “breakfast”
    b. Amount: “1500”
    c. Date: “2023-06-01 8:30 AM”
    d. Category: 
    e. Leave the Notes empty
    f. Who paid?: “Suhjin Kang”
    g. Related members: “Luke Lepa”, “Taeho Choi”
4. Transaction added with the following information:
    a. Item details: “Ticket to museum”
    b. Amount: “3900”
    c. Date: “2023-06-05 10:00 AM”
    d. Category: 
    e. Leave the Notes empty
    f. Who paid?: “Leonardo Warsito”
    g. Related members: “Suhjin Kang”
5. Transaction added with the following information:
    a. Item details: “gifts for family”
    b. Amount: “10004”
    c. Date: “2023-06-04 12:00 PM”
    d. Category: 
    e. Leave the Notes empty
    f. Who paid?: “Suhjin Kang”
    g. Related members: empty
6. Transaction added with the following information:
    a. Item details: “hotel”
    b. Amount: “50000”
    c. Date: “2023-06-01 12:00 PM”

d. Category: 
e. Leave the Notes empty
f. Who paid?: “Suhjin Kang”
g. Related members: empty

7. Transaction added with the following information:
    a. Item details: “bus fares”
    b. Amount: “4000”
    c. Date: “2023-06-04 12:00 PM”
    d. Category: 
    e. Leave the Notes empty
    f. Who paid?: “Suhjin Kang”
    g. Related members: empty

Test:
1. Click on the “Japan with friends” button on the left sidebar under “UPCOMING”
2. Click on the  button top of the 3rd column

Expected result:
1. Under “Top 5 Transactions” section, it should display the following in order:
    a.  “hotel ¥50000”
    b.  “gifts for family ¥10004”
    c.  “bus fares ¥4000”
    d.  “Ticket to museum ¥3900”
    e.  “breakfast ¥1500”

User story 15: As a traveller, I want to be able to view my transactions in a pie chart by category so that I can see where I spent the most money

Expected condition:
1. Logged in successfully
2. Trip added with the following information:
    a. Trip title: “Japan with friends”
    b. Start date: “2023-06-01”
    c. End date: “2023-06-22”
    d. Destination: “Japan”
    e. Budget: “100000”
    f. There are the following members part of the trip “Japan with friends”
        i. “Suhjin Kang”
        ii. “Luke Lepa”
        iii. “Taeho Choi”
        iv. “Leonardo Warsito”
3. Transaction added with the following information:
    a. Item details: “breakfast”

b. Amount: “1500”
c. Date: “2023-06-01 8:30 AM”
d. Category: 
e. Leave the Notes empty
f. Who paid?: “Suhjin Kang”
g. Related members: “Luke Lepa”, “Taeho Choi”

4. Transaction added with the following information:
    a. Item details: “Ticket to museum”
    b. Amount: “3900”
    c. Date: “2023-06-05 10:00 AM”
    d. Category: 
    e. Leave the Notes empty
    f. Who paid?: “Leonardo Warsito”
    g. Related members: “Suhjin Kang”
5. Transaction added with the following information:
    a. Item details: “gifts for family”
    b. Amount: “10004”
    c. Date: “2023-06-04 12:00 PM”
    d. Category: 
    e. Leave the Notes empty
    f. Who paid?: “Suhjin Kang”
    g. Related members: empty
6. Transaction added with the following information:
    a. Item details: “hotel”
    b. Amount: “50000”
    c. Date: “2023-06-01 12:00 PM”
    d. Category: 
    e. Leave the Notes empty
    f. Who paid?: “Suhjin Kang”
    g. Related members: empty
7. Transaction added with the following information:
    a. Item details: “bus fares”
    b. Amount: “4000”
    c. Date: “2023-06-04 12:00 PM”
    d. Category: 
    e. Leave the Notes empty
    f. Who paid?: “Suhjin Kang”
    g. Related members: empty

Test:

1. Click on the “Japan with friends” button on the left sidebar under “UPCOMING”
2. Click on the  button on top of the 3rd column

Expected result:

1. On the top of the “Report” section, it should display a pie chart with all the categories
2. Clicking on each of the categories should display the category icon and the amount spent on the category in the centre of the pie chart

Load Testing:

For the load testing, we will be testing the condition of being able to run the program at 1000 requests for 100 users every minute. We will do so by running the test for 1 minute while adding 5 new users per minute. There are a couple tests in total where each user would be doing one of the three testing scenarios available.

Test:

1. Main page simulation
    a. Create a new user
    b. View all the user’s trip
    c. View the user profile
2. Trip Creation, Edit and Deletion
    a. Create a new user
    b. Create a new trip
    c. Edit the newly created trip
    d. Delete the trip
3. Report Viewing
    a. Create a new user
    b. Go to a trip
    c. Go to the report section
    d. View the top 5 transactions
    e. View the total cost of the trip

Run:

1. Go to /server folder
2. Run `npm run load-test` in the console

Test Completeness:

Criteria for test completeness:

- 100% code coverage (Funcs), all the source code should be covered by test cases.
- All Manual & Automated Test cases executed.
- All open bugs are fixed or will be fixed in the next release.

3 Resource & Environment Needs

3.1 Testing Tools

- Requirements Tracking Tool: Jest, SuperTest

- Bug Tracking Tool: Chrome Developer Tools
- Load Testing Tool: Artillery

3.2 Test Environment

- Local Environment: Node.js [16.x] (NPM (8.15.0), Yarn (1.22.19))
- GitHub Actions: Node.js [16.x]

4 Terms/Acronyms

Make a mention of any terms or acronyms used in the project

TERM/ACRONYM	DEFINITION
API	Application Program Interface