Test Document

for

Brokeo

Version <1.0>

Prepared by

Group Number:17 **Group Name: Runtime3rror**

Anjali Patra	**230148**	anjalip23@iitk.ac.in
Aujasvit Datta	**220254**	aujasvitd22@iitk.ac.in
Bhavnoor Singh	**230293**	bhavnoor23@iitk.ac.in
Darshan Sethia	**220326**	darshands22@iitk.ac.in
Dhriti Barnwal	**230364**	dhritib23@iitk.ac.in
Marisha Thorat	**230637**	marishajt23@iitk.ac.in
Rudransh Verma	**230881**	rudranshv23@iitk.ac.in
Sanjna S	**230918**	sanjnas23@iitk.ac.in
Solanki Shrey Jigneshbhai	**231017**	shreyjs23@iitk.ac.in
Suryansh Verma	**231061**	suryanshv23@iitk.ac.in

Revisions

Version	Primary Author(s)	Description of Version	Date Completed
1.0	Full Group	Completed first version of the document	06/04/25

1 Introduction

Test Strategy

Brokeo is a financial management app designed to streamline expense tracking, budgeting, transaction categorization, and split payments. The app ensures seamless financial management by integrating essential features such as transaction analysis, notifications, and multi-user expense splitting.

Our team's testing strategy for Brokeo primarily involves manual testing, where testers systematically verify each feature by providing various inputs and comparing the observed outputs with the expected results. Given the financial nature of the app, accuracy and reliability are critical, necessitating rigorous validation at both the frontend and backend levels. Exhaustive functional testing and database validation were conducted.

Following a structured testing approach, we ensure that Brokeo provides users with a reliable and efficient financial management tool.

Testing Period of the App

The testing was an integral part of the development process. During the implementation, the app was subjected to tests corresponding to basic functions. However, the majority of the testing of the app was done after the implementation of the app. Since the basic functionality of the app was set in stone and tested, any bugs encountered in the app were easier to tackle since they involved changes only in specific features of the app (certain functions of the source code). This made testing a lot more efficient and less time-consuming.

Testers of the App

Due to time constraints and the size of the team, the app's developers were the testers themselves. However, it was ensured that the responsibility of testing units and components was not placed on the person who developed or played a role in developing them. This ensured the neutrality of the testing process. The benefit of having the developers be the testers was that we knew the different possible inputs and hence were able to carry out extensive testing to ensure the robustness of the application.

Multiple errors were identified and corrected as and when discovered, and some of them are being worked on.

Coverage Criterion for Testing

Functional requirements were the criterion used for the testing.

Tools used in testing

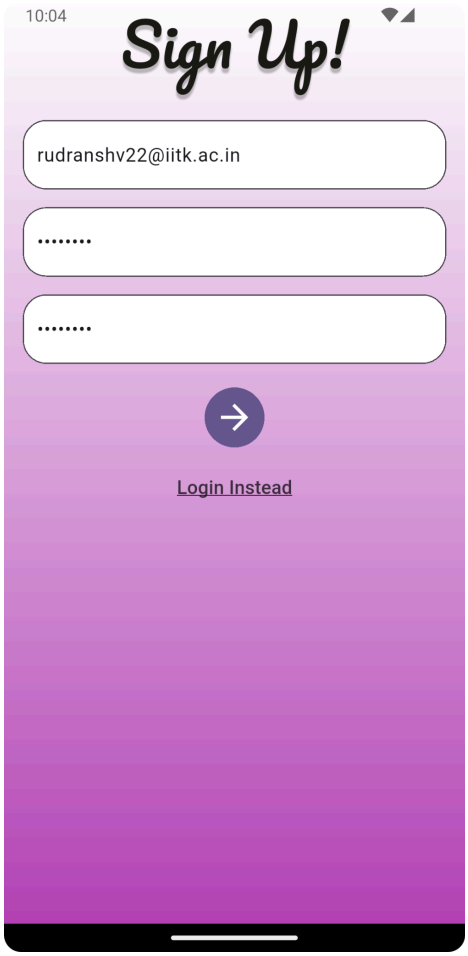
No tools were used for testing. The entire process was manual.

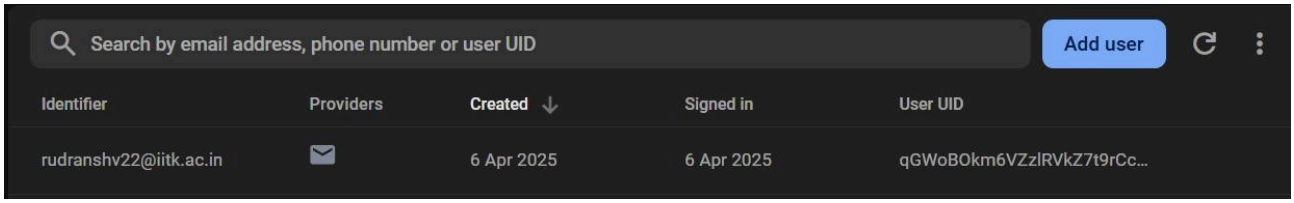
2 Unit Testing

1. Signing up:

This module handles user registration through the Sign Up page. Users can create a new account by entering a valid email and matching passwords. Upon successful registration, the new user is authenticated and their account is recorded in Firebase, ensuring smooth backend integration and secure onboarding.

Unit Details: Sign up Page
Test Owner: Anjali Patra
Test Date: 31/03/25 - 31/03/25
Test Results: The Sign Up module is functioning correctly. A new user was successfully registered using a valid email and password. The account creation was verified in the Firebase Authentication console, confirming backend integration.





Successful user registration was validated via the Sign Up page, with the new account correctly reflected in the Firebase Authentication console.

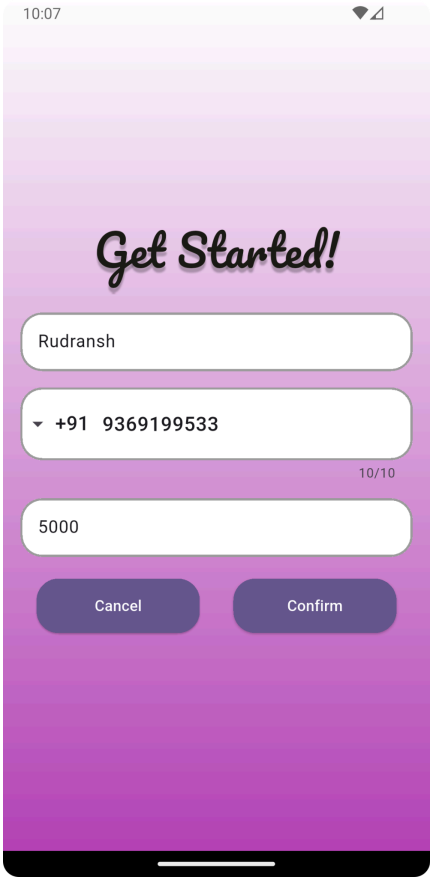
Structural Coverage: Covered user input validation (email and password), password match check, and successful account creation with database entry verification.

Additional Comments: The Sign Up screen properly validates form fields and prevents registration with mismatched or invalid inputs. The registered user data is stored securely in Firebase, and UI feedback is provided for successful registration.

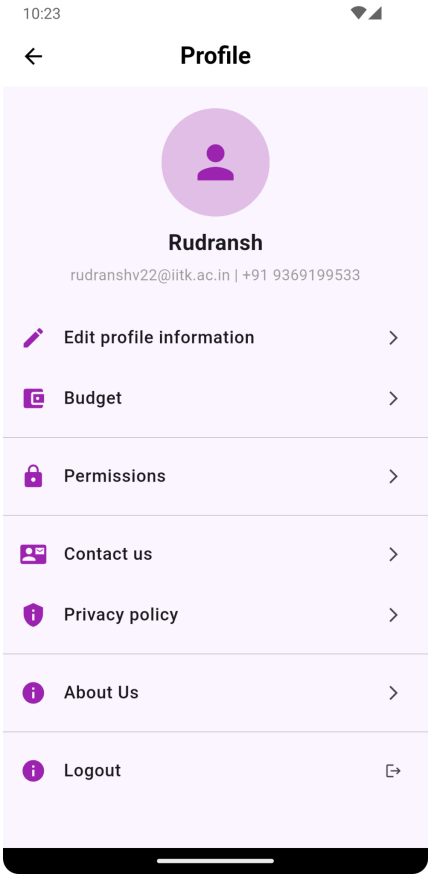
2. Login Page:

This module handles user login and initiates the onboarding process. Upon successful login, users are redirected to the Get Started page to input their name, phone number, and initial budget. The entered details, along with the user's email, are then accurately displayed on the Profile page, ensuring a smooth and consistent login-to-profile data flow.

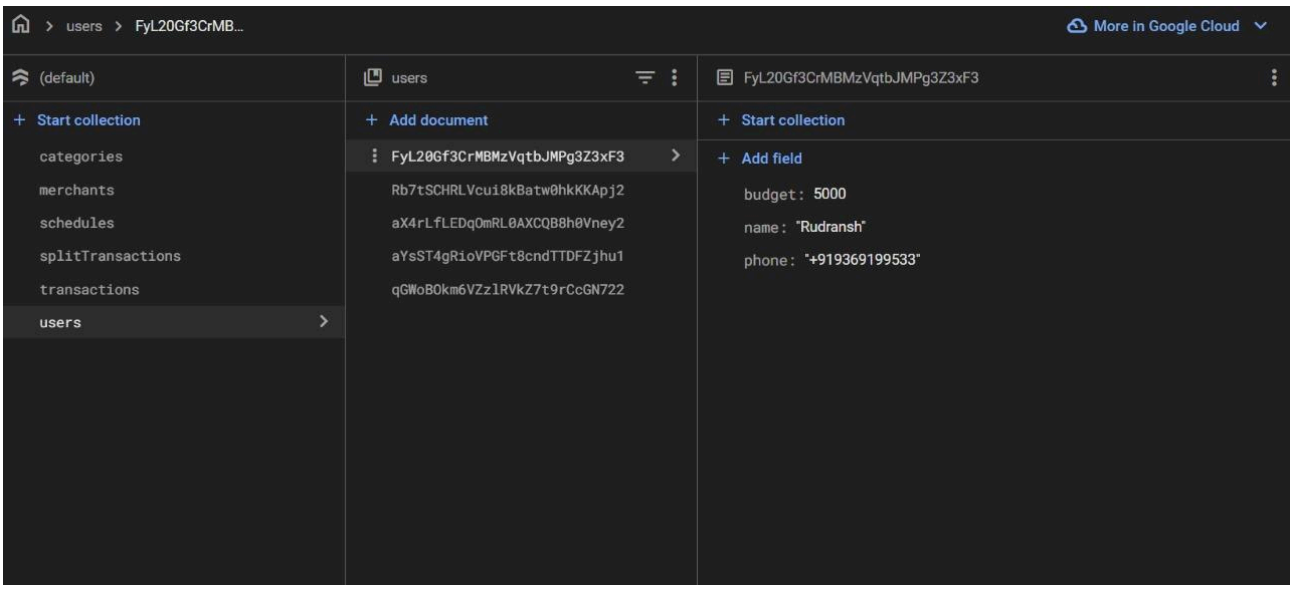
Unit Details: Login Page
Test Owner: Suryansh Verma
Test Date: 31/03/25 - 31/03/25
Test Results: The login module is fully functional.



On first login, users are correctly directed to the Get Started page to enter their phone number, name, and initial budget.



Upon successful login and input, the email, phone number, and name are accurately displayed on the Profile page.



This information to the database successfully.

Structural Coverage: All primary login flows were tested, including valid email entry, redirection to the Get Started page, and accurate storage/display of user information. Edge cases such as incomplete logins and invalid emails, were also covered to ensure proper validation and error handling.

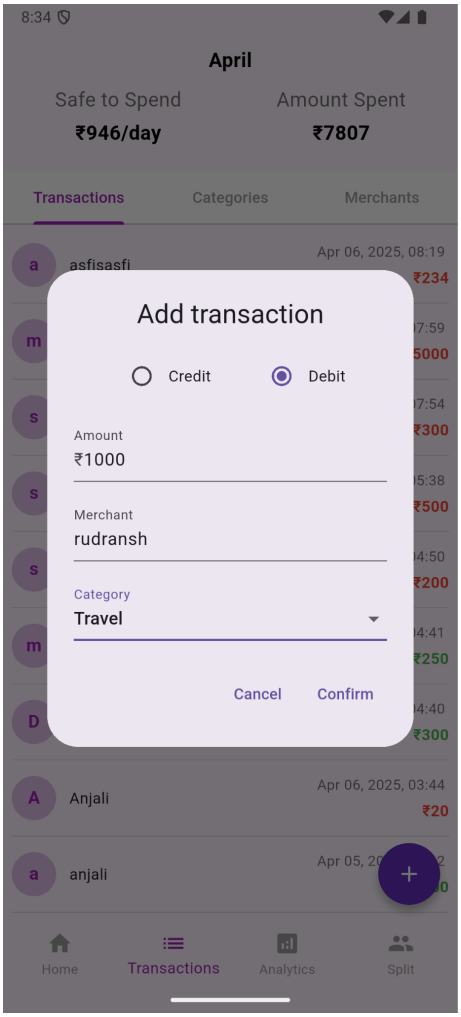
Additional Comments: The redirection and data flow from the login to the Profile page function seamlessly. All input fields behaved as expected, and validation worked correctly.

3. Safe to spend and Amount spent:

This module tests the functionality of the displayed Safe to Spend and Amount Spent functions of the transactions page upon adding/deleting a transaction.

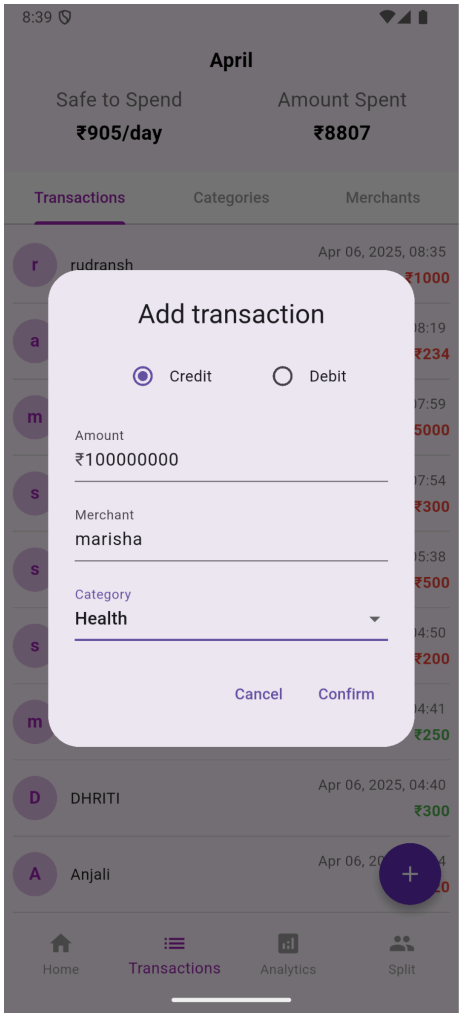
Unit Details: Transactions page
Test Owner: Sanjna S
Test Date: 31/03/25 - 31/03/25
Test Results: No bugs were found. This functionality worked properly.

Before adding the transaction:



After adding the transaction:

For cases where a transaction exceeds the set budget:



The same was done for the deletion of transactions.

Structural Coverage: The test covered calculations and UI updates for both “Safe to Spend” and “Amount Spent” sections upon adding and deleting transactions. Scenarios tested included credit, debit, and zero-value transactions. The module was verified for correct data binding, dynamic updates, and alignment with the overall budget logic.

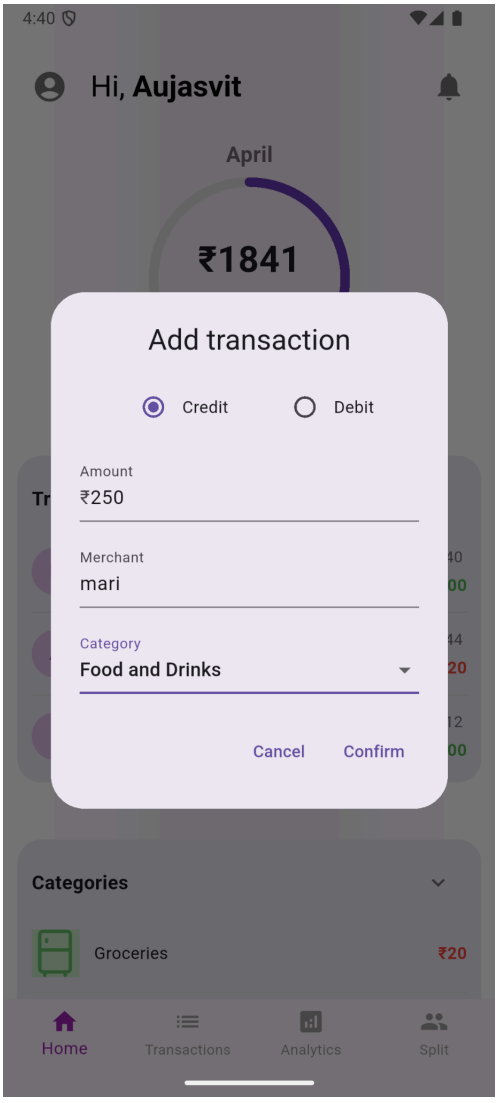
Additional comments: Values updated accurately and in real-time with each transaction. Both the visual and backend states remained consistent throughout the test.

4. Adding and Deleting Transactions:

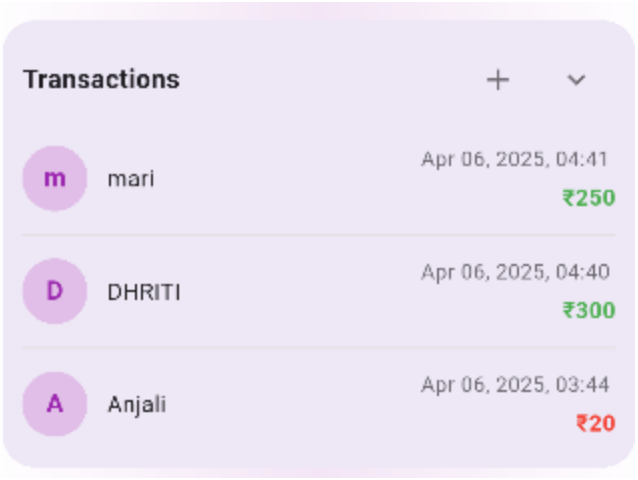
This module allows users to add both credit and debit transactions, with each entry dynamically updating the radial progress bar to reflect the current financial state. Additionally, users can delete existing transactions.

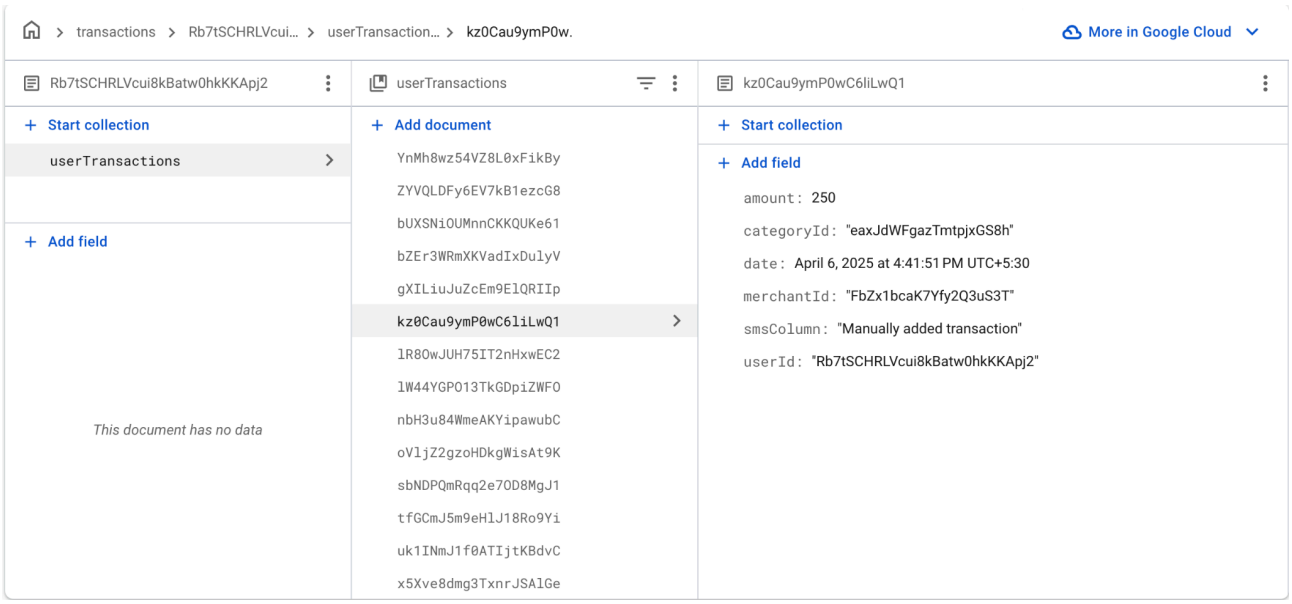
Unit Details: Transaction
Test Owner: Dhriti Barnwal
Test Date: 31/03/25 - 31/03/25
Test Results: This transaction module of Brokeo is fully functional, with users being able to easily add and delete transactions

Add Transaction:
We added a new credit transaction

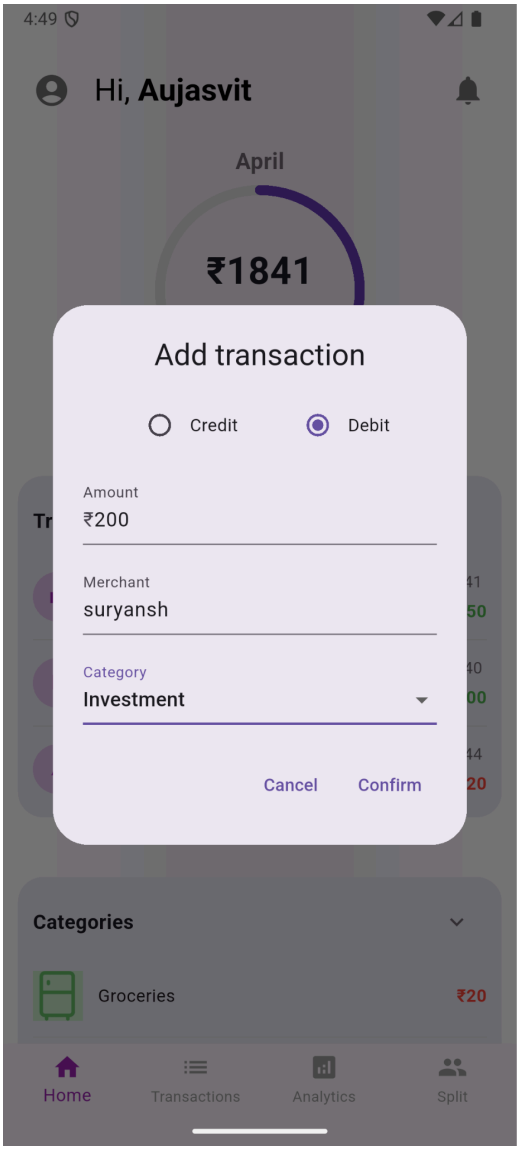


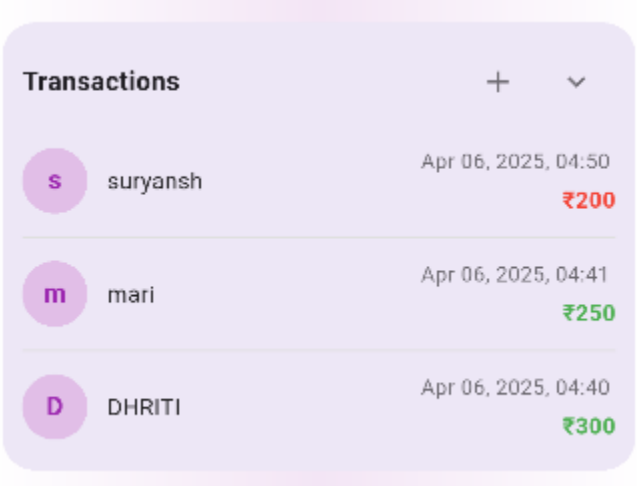
After confirmation, the added transaction is shown and also reflected in the database

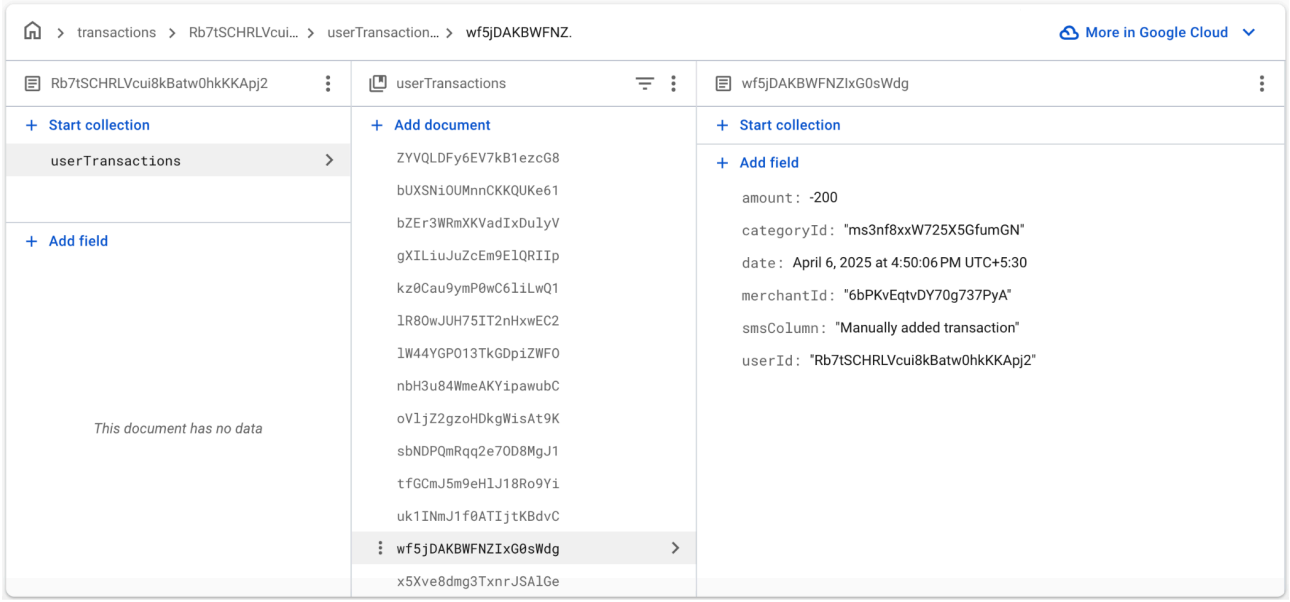


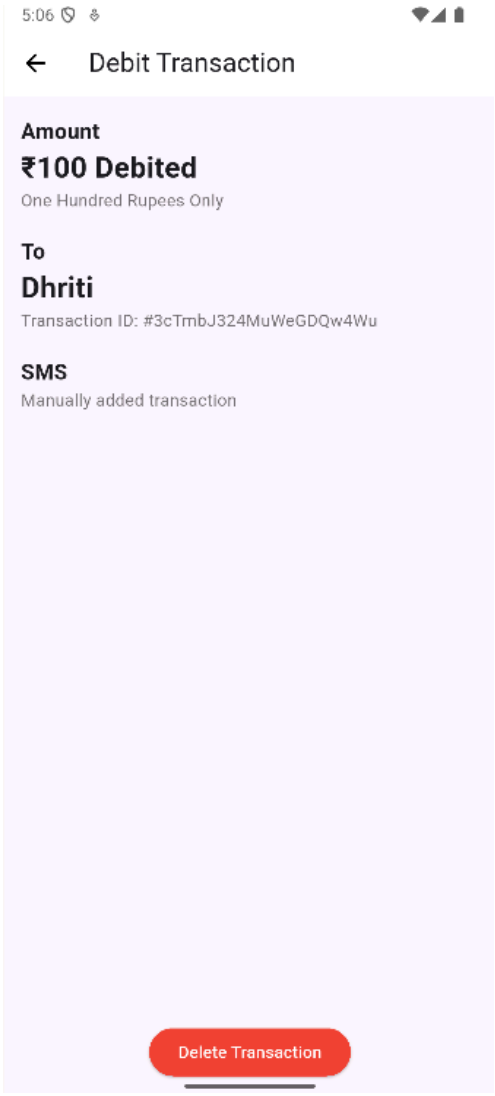


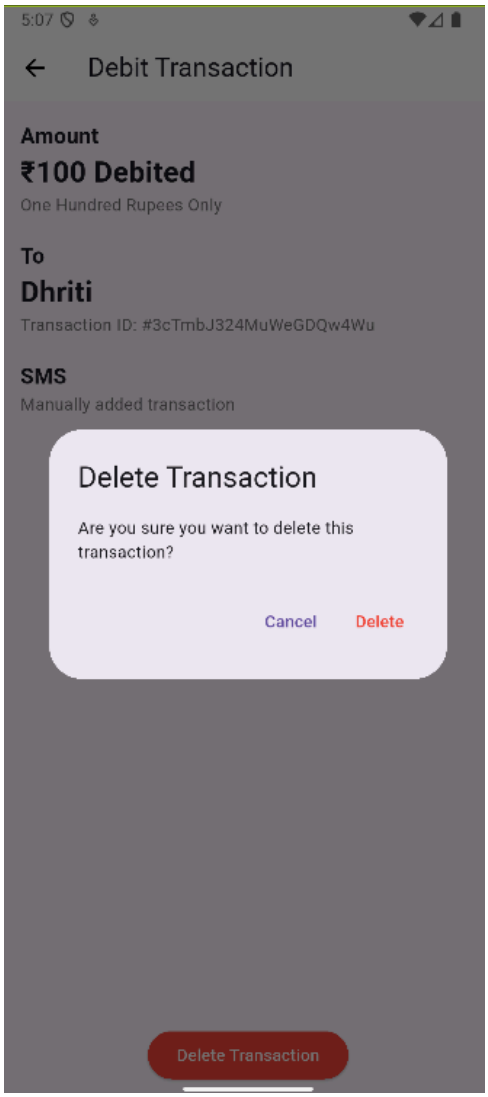
We added a new debit transaction, which was successfully reflected in the database.


4:49
Hi, Aujasvit
April
₹1841
Add transaction
Credit
Debit
Amount
₹200
Merchant
suryansh
Category
Investment
Cancel
Confirm
Categories
Groceries
₹20
Home
Transactions
Analytics
Split


Transactions
s
suryansh
Apr 06, 2025, 04:50
₹200
m
mari
Apr 06, 2025, 04:41
₹250
D
DHRITI
Apr 06, 2025, 04:40
₹300


transactions
Rb7tSCHRLVcui...
userTransaction...
wf5jDAKBWFNZ.
More in Google Cloud
Rb7tSCHRLVcui8kBatw0hkKKApj2
userTransactions
wf5jDAKBWFNZIxG0sWdg
Start collection
Add document
Start collection
userTransactions
Add field
Add field
This document has no data
ZYVQLDFy6EV7kB1ezcG8
bUXSNiOUMnnCKKQUKe61
bZEr3WRmXKVadIxDulyV
gXILiuJuZcEm9ElQRIIp
kz0Cau9ymP0wC6liLwQ1
lR8OwJUH75IT2nHxwEC2
lW44YGPO13TkGDpiZWFO
nbH3u84WmeAKYipawubC
oVljZ2gzoHDkgWisAt9K
sbNDPQmRqq2e7OD8MgJ1
tfGCmJ5m9eHlJ18Ro9Yi
uk1INmJ1f0ATIjtKBdvC
wf5jDAKBWFNZIxG0sWdg
x5Xve8dmg3TxnrJSAlGe
amount: -200
categoryId: "ms3nf8xxW725X5GfumGN"
date: April 6, 2025 at 4:50:06 PM UTC+5:30
merchantId: "6bPKvEqtvDY70g737PyA"
smsColumn: "Manually added transaction"
userId: "Rb7tSCHRLVcui8kBatw0hkKKApj2"





We then navigated to the transaction detail page and successfully deleted a transaction. This functionality is working as expected.

Structural Coverage: This unit test covers the core logic for adding and deleting both credit and debit transactions. It verifies successful transaction creation with correct data binding and ensures each addition accurately updates the radial progress bar. The deletion flow was tested for proper removal of transactions and corresponding UI updates.

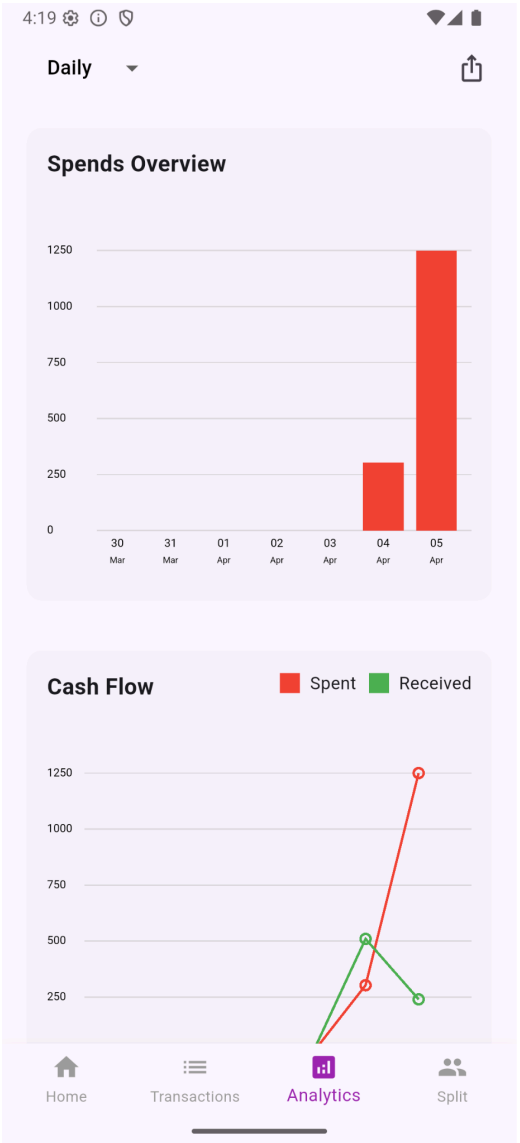
Additional Comments: The "Add Transaction" feature worked as expected for various input types and amounts. Transactions were immediately visible in the transaction list, and the progress bar reflected changes correctly.

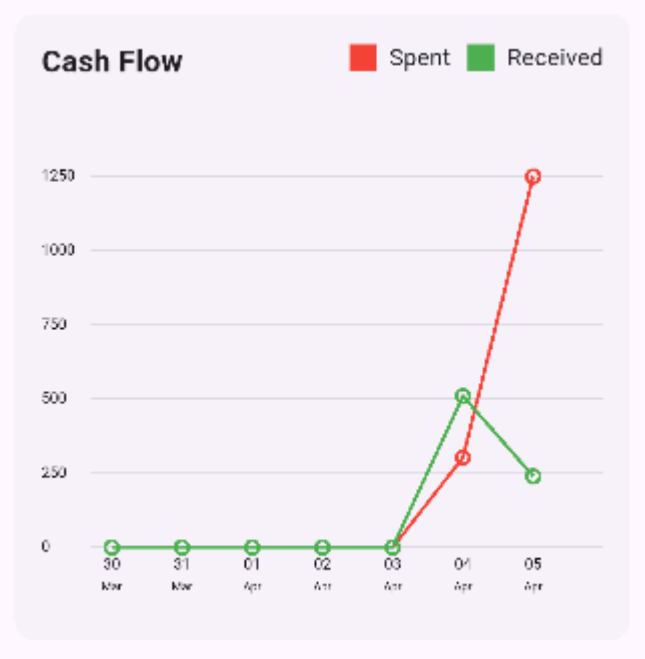
5. Analytics Page:

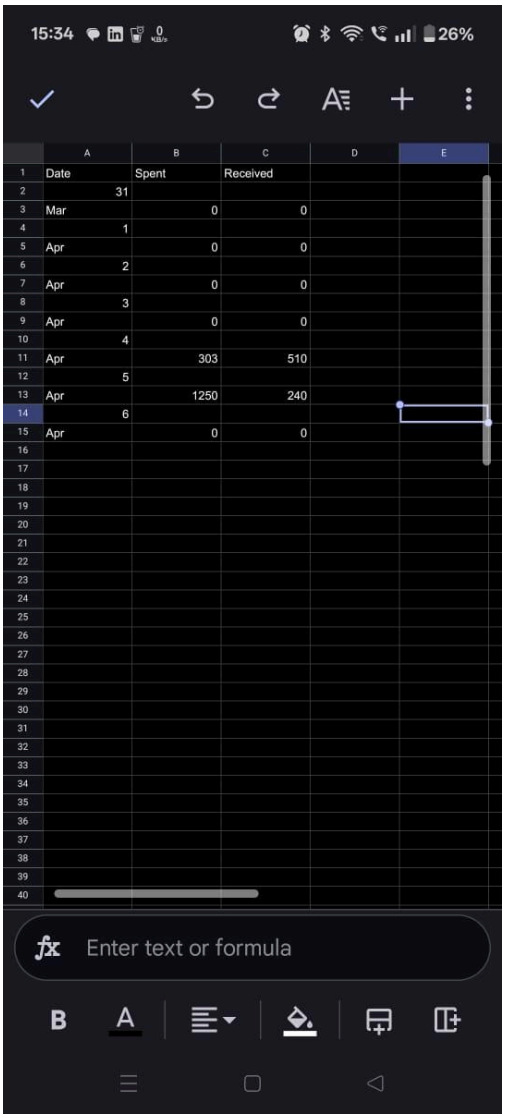
The Analytics Page enables the user to monitor their weekly, monthly and yearly spending in detail and allows the user to export their weekly, monthly and yearly transaction.

Unit Details: Analytics
Test Owner: Anjali Patra
Test Date: 31/03/25 - 31/03/25
Test Results: This page is fully functional, with users being able to view and export analytics. No bugs were found.

The drop-down button will allow the user to view daily, weekly and monthly analytics.
Daily analytics are displayed below.



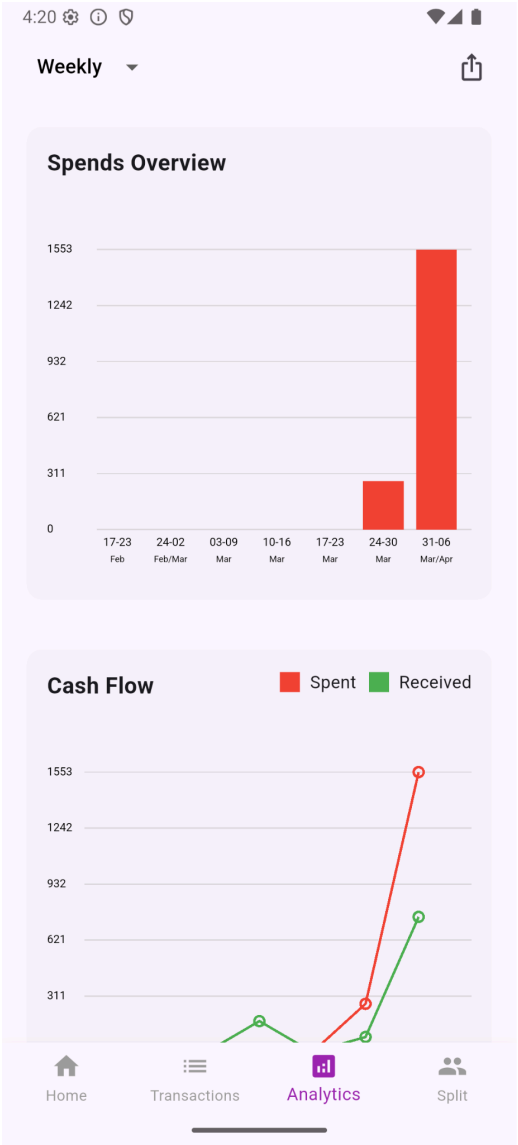


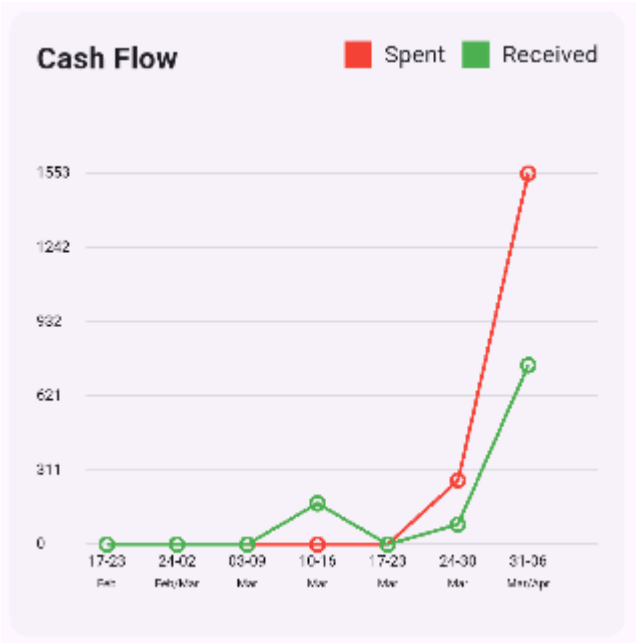


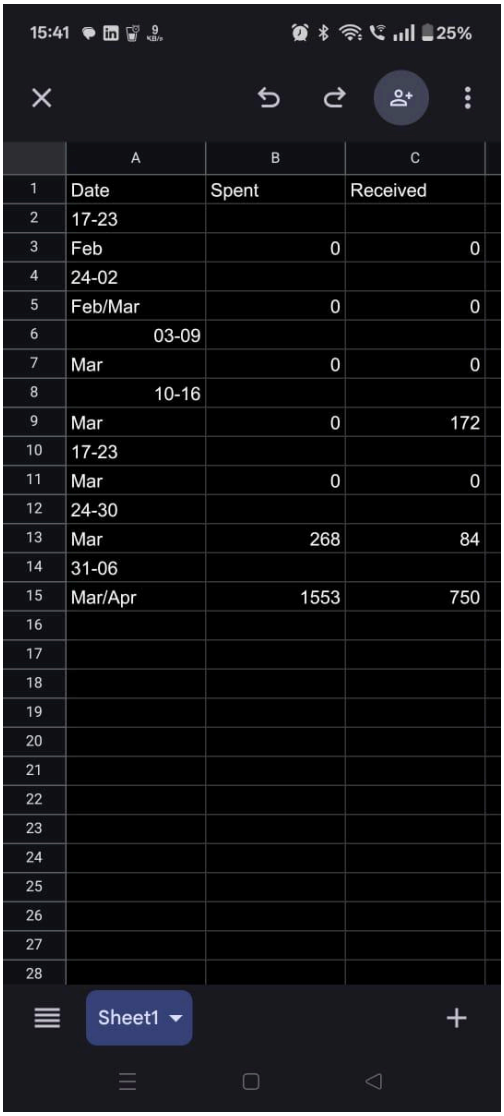
The CSV file was generated successfully.

The same has been verified for weekly analytics and monthly analytics.

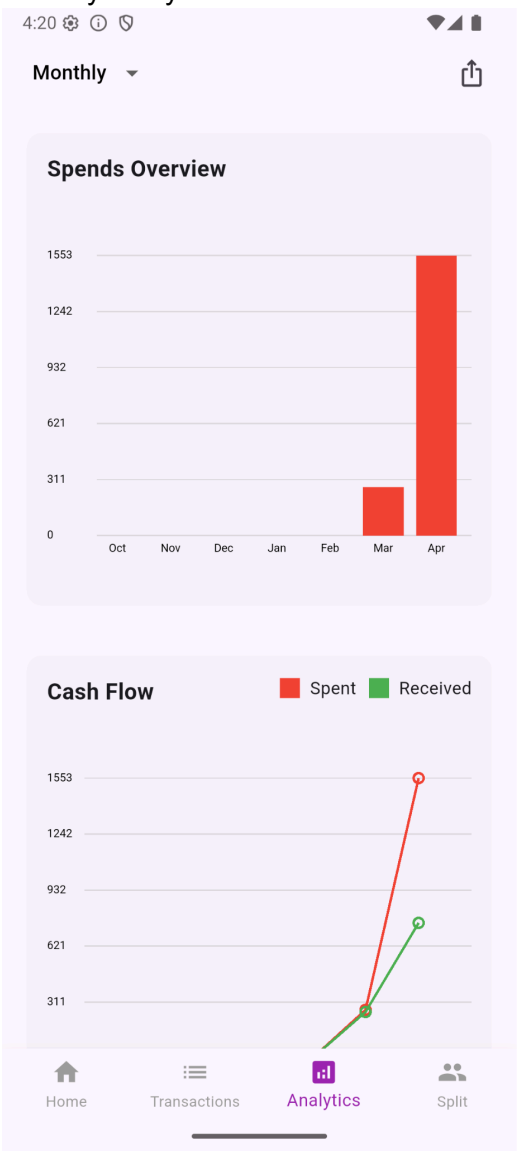
Weekly analytics

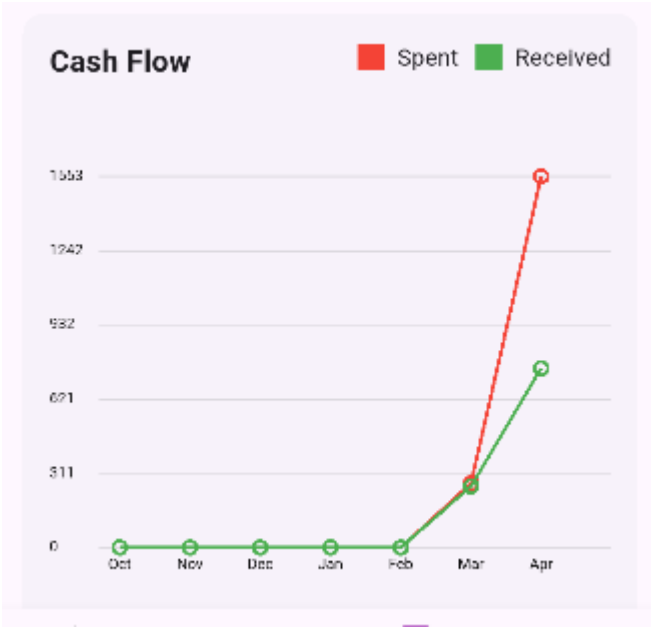


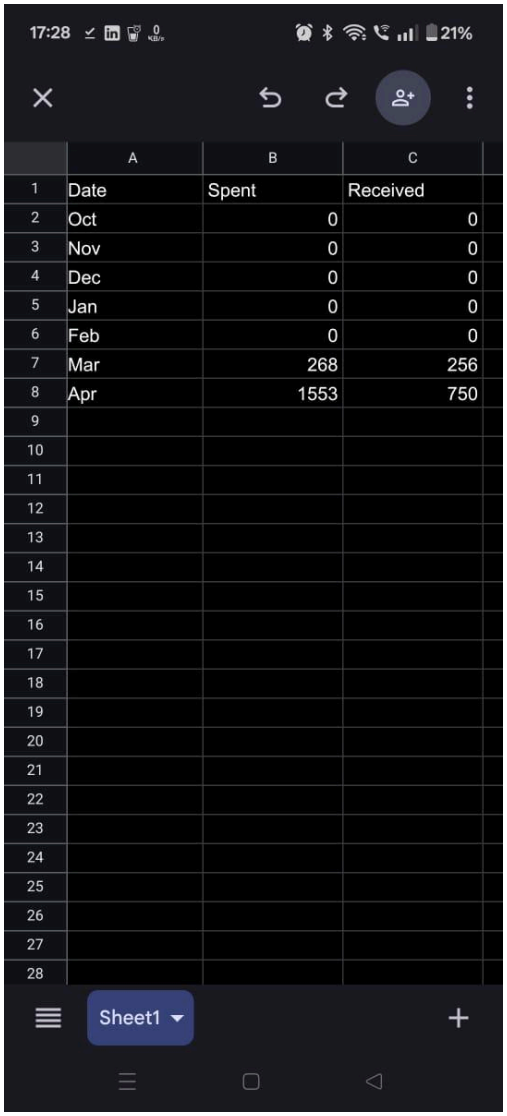


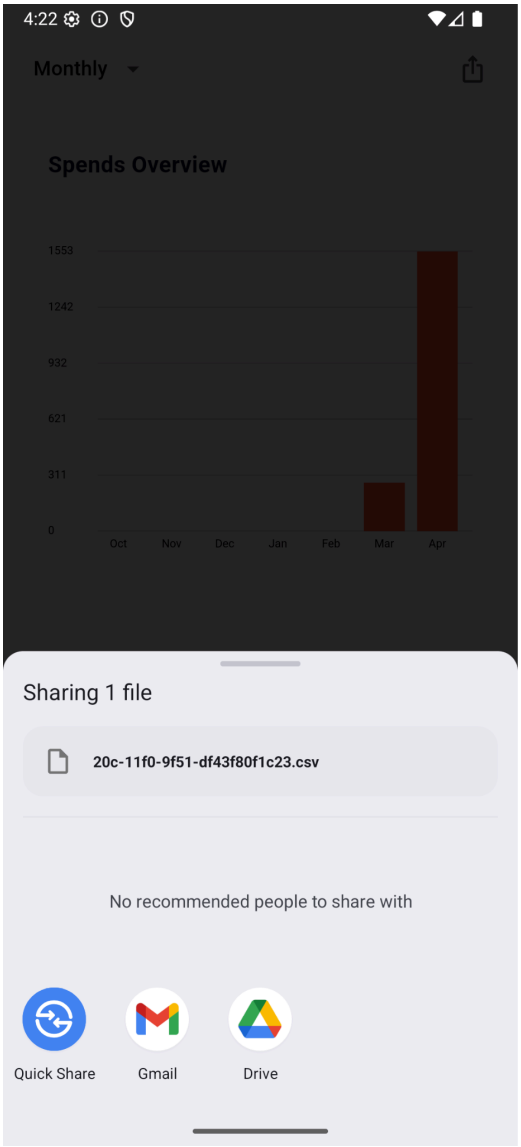

15:41
25%
A
B
C
1 Date Spent Received
2 17-23
3 Feb 0 0
4 24-02
5 Feb/Mar 0 0
6 03-09
7 Mar 0 0
8 10-16
9 Mar 0 172
10 17-23
11 Mar 0 0
12 24-30
13 Mar 268 84
14 31-06
15 Mar/Apr 1553 750
16
17
18
19
20
21
22
23
24
25
26
27
28
Sheet1

Monthly analytics






17:28
21%
A
B
C
1 Date Spent Received
2 Oct 0 0
3 Nov 0 0
4 Dec 0 0
5 Jan 0 0
6 Feb 0 0
7 Mar 268 256
8 Apr 1553 750
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
Sheet1



The sharing function works properly, too.

Structural Coverage: The test covers rendering of analytics for all supported durations—daily, weekly, monthly, and yearly. The dropdown functionality was tested to ensure correct data loads for each selected period. Export functionality for weekly, monthly, and yearly transactions was also verified, confirming successful file generation and download triggers.

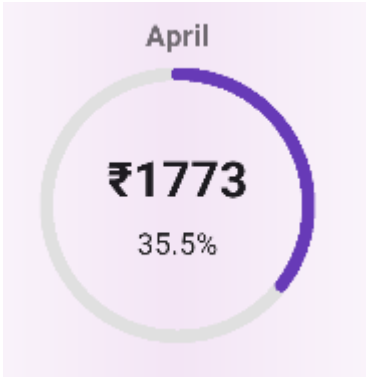
Additional Comments: Daily analytics were displayed accurately, with charts and summaries rendering correctly. The dropdown control was responsive and intuitive. Exported files matched the displayed data, with no formatting issues observed.

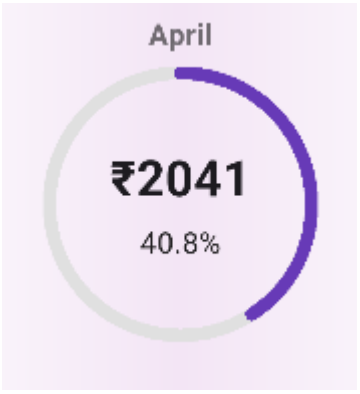
6. Radial Progress Bar:

The radial progress bar displays the amount spent in the given month and the percentage of the budget that has been spent. It provides a comprehensive summary of the money spent that month visually.

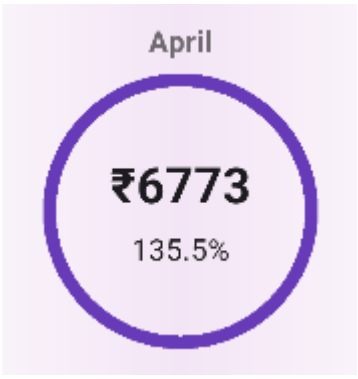
Unit Details: Radial progress bar
Test Owner: Rudransh Verma
Test Date: 31/03/25 - 31/03/25
Test Results: A minor bug was discovered and later fixed. Upon testing, it was discovered that the total money spent was calculated based on all transactions and not just those of the given month.

Once the total money spent exceeds the total budget, the radial progress bar will appear fully covered, as shown below:

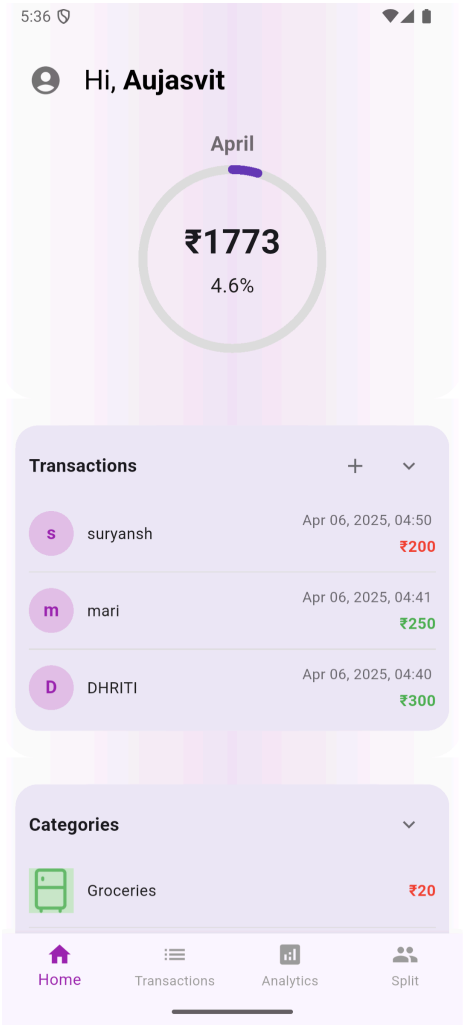


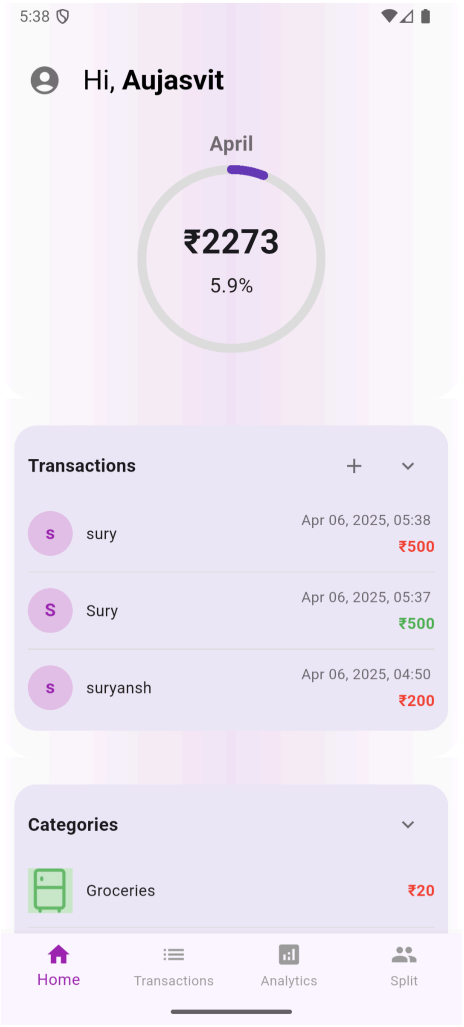


The image on the left shows the initial total expenditure but it is a bug as it includes the expenditure which happened in previous months also, later we fixed it so it records only the current month's expenditure.



The first image displays the radial progress bar when we exceeded the budget





The above pictures were obtained after the application records a debit transaction of 500 rupees with the budget set to 5000 rupees.

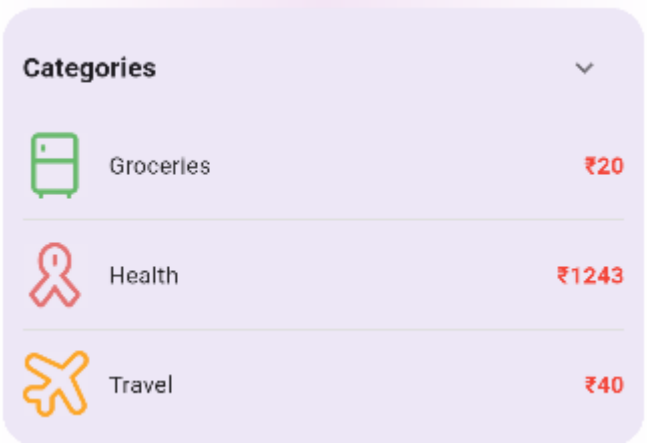
Structural coverage: All logical paths were tested, including scenarios with no transactions, transactions within the current month, and those outside the selected month. Edge cases, such as zero budget and 100%+ spending, were also covered. Full test coverage was achieved, ensuring accurate calculation and visual feedback in all conditions.

Additional Comments: The initial bug related to the incorrect inclusion of out-of-month transactions was promptly resolved. Post-fix, the progress bar accurately reflects spending for the selected month only. The UI updates smoothly, and the percentage display aligns with backend calculations. Visual clarity and responsiveness were maintained throughout.

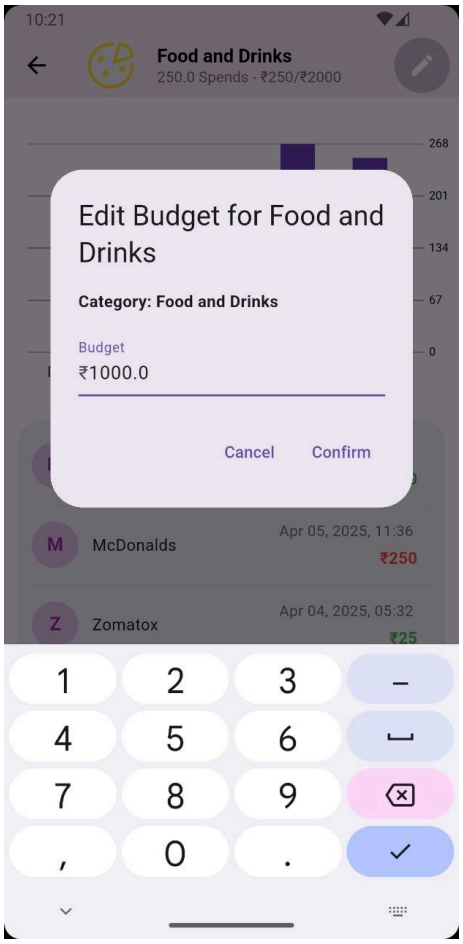
7. Editing Category Budget :

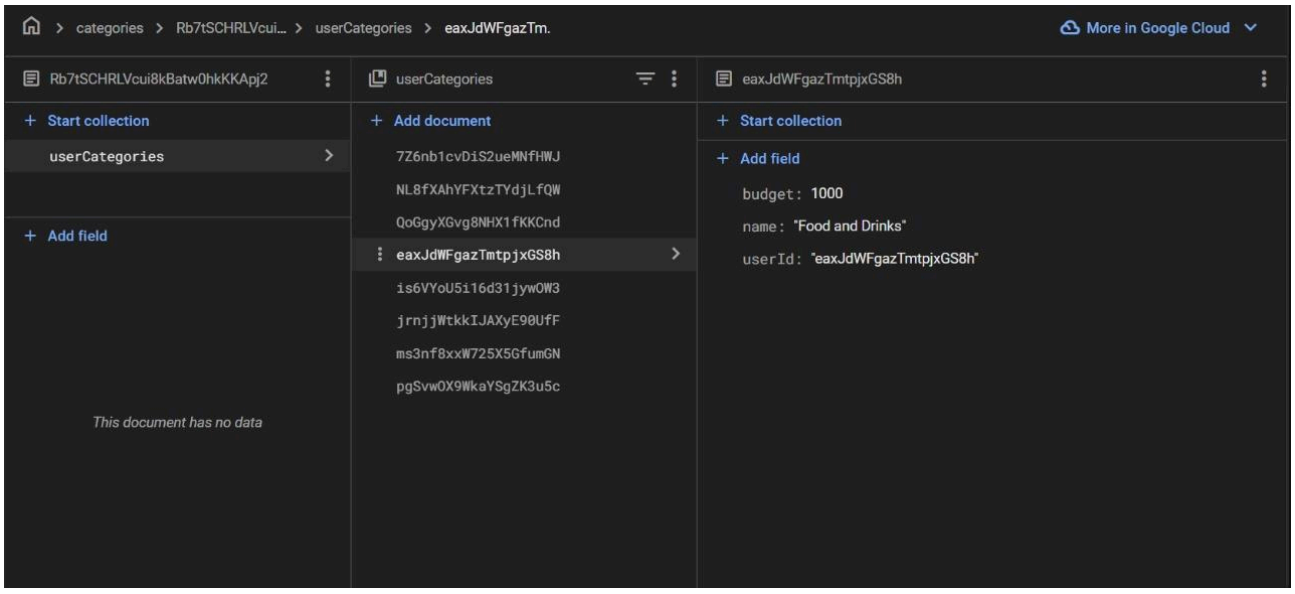
This module allows users to modify the budget allocated to a specific category. Upon editing, the updated budget is dynamically reflected across all relevant views, including beneath the category name and on the Categories page. This ensures real-time synchronisation between user input and the visual representation of budget allocations.

Unit Details: Categories
Test Owner: Dhriti Barnwal
Test Date: 31/03/25 - 31/03/25
Test Results: Few bugs were found and then fixed. This transaction module of Brokeo is fully functional with users being able to easily edit the category budget. While editing the budget, the changes were reflected in the database but not in the UI. It was then later resolved by using category ids as a reference instead. The pop-up accepted null values as inputs but did not modify the values stored as null values. Both of these issues were successfully fixed later.




10:20
Food and Drinks
250.0 Spends - ₹250/₹2000
268
201
134
67
0
Dec
Jan
Feb
Mar
Apr
M mari Apr 06, 2025, 04:41 ₹250
M McDonalds Apr 05, 2025, 11:36 ₹250
Z Zomatox Apr 04, 2025, 05:32 ₹25
A Anjali Apr 04, 2025, 01:05 ₹23
Z Zing Apr 04, 2025, 12:16 ₹202
Z Zomato Mar 24, 2025, 09:11 ₹268


10:21
Food and Drinks
250.0 Spends - ₹250/₹2000
Edit Budget for Food and Drinks
Category: Food and Drinks
Budget
₹1000.0
Cancel
Confirm
M McDonalds Apr 05, 2025, 11:36 ₹250
Z Zomatox Apr 04, 2025, 05:32 ₹25


categories > Rb7tSCHRLVcui... > userCategories > eaxJdWFgazTm.
More in Google Cloud
Rb7tSCHRLVcui8kBatw0hkKKApj2
userCategories
eaxJdWFgazTmtpjxGS8h
Start collection
userCategories
Add field
This document has no data
Add document
7Z6nb1cvDiS2ueMNfHWJ
NL8fXAhYFXtzTYdjLfQW
QoGgyXGvg8NHX1fKKCnd
eaxJdWFgazTmtpjxGS8h
is6VYoU5i16d31jywOW3
jrnjjWtkkIJAXyE90UfF
ms3nf8xxW725X5GfumGN
pgSvwOX9WkaYSgZK3u5c
Start collection
Add field
budget: 1000
name: "Food and Drinks"
userId: "eaxJdWFgazTmtpjxGS8h"

After using the function, we end up with a modified budget that has been displayed in categories section page, category page and database.

Structural coverage: All primary and edge cases were tested, including valid budget edits, null input handling, and real-time UI synchronization. Test coverage includes updating category-specific budgets, ensuring correct propagation across the UI and database, and verifying data consistency using category IDs. Input validation for empty or invalid values was also assessed and corrected.

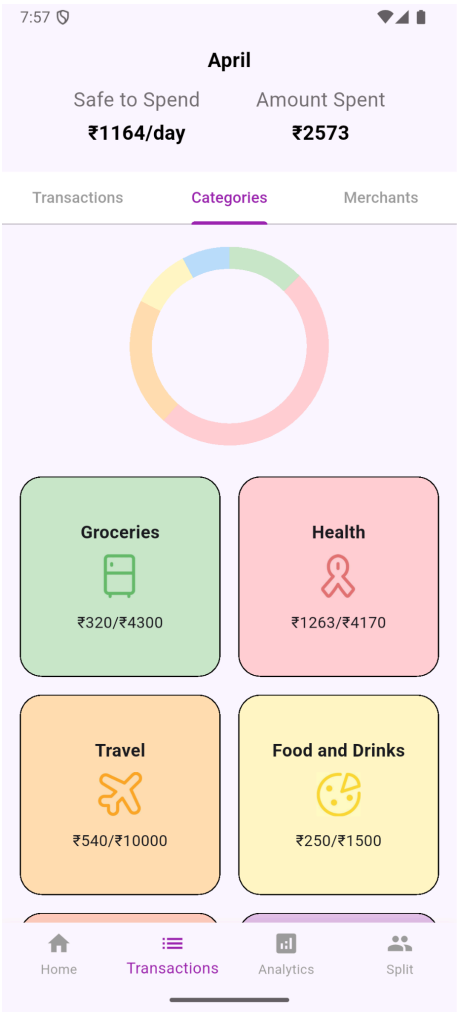
Additional Comments: Initial issues with UI not reflecting updates and handling of null inputs were resolved effectively. Post-fix, category budgets update smoothly in both the UI and database with accurate real-time feedback. The use of category IDs significantly improved state tracking and data binding.

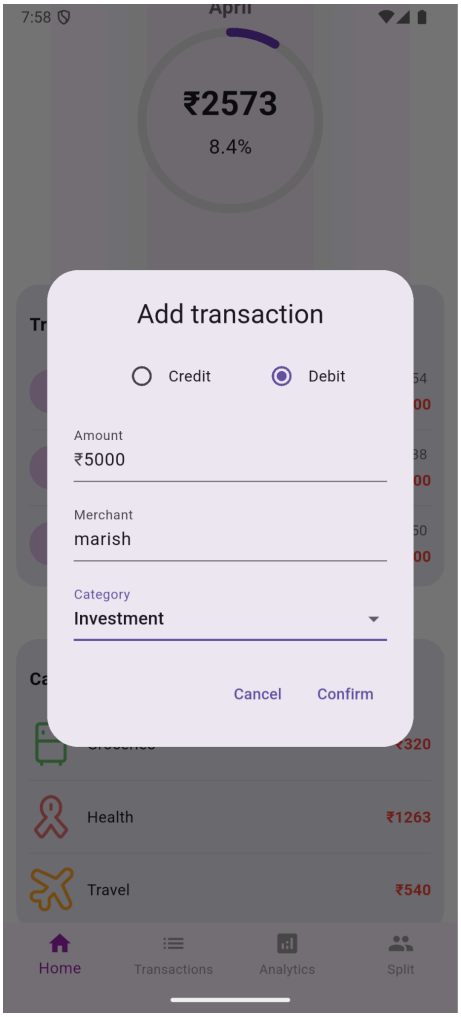
8. Radial Category chart:

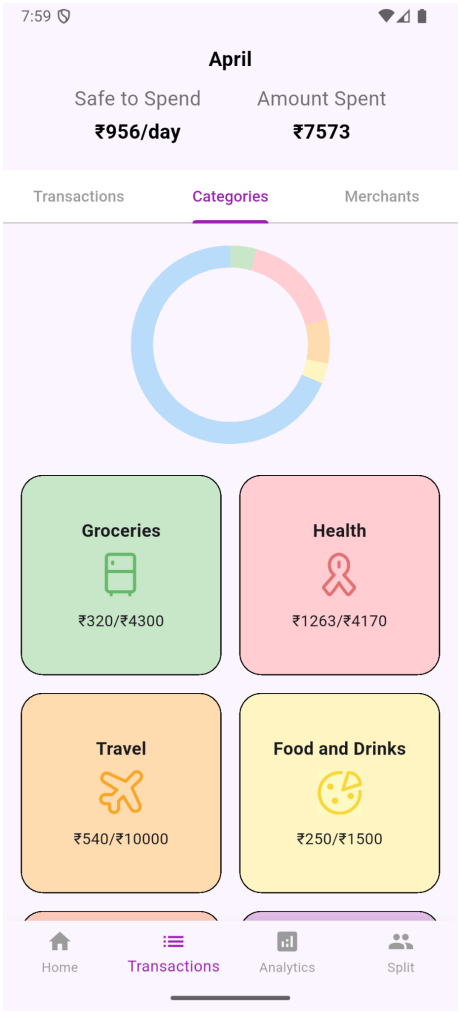
This module tests if the radial category chart of the category tab properly reflects new transactions.

Unit Details: Transactions page
Test Owner: Bhavnoor Singh
Test Date: 31/03/25 - 31/03/25
Test Results: No bugs were found. This functionality works properly

Before the addition of the transaction:







The new chart successfully reflects the addition of the new transaction.

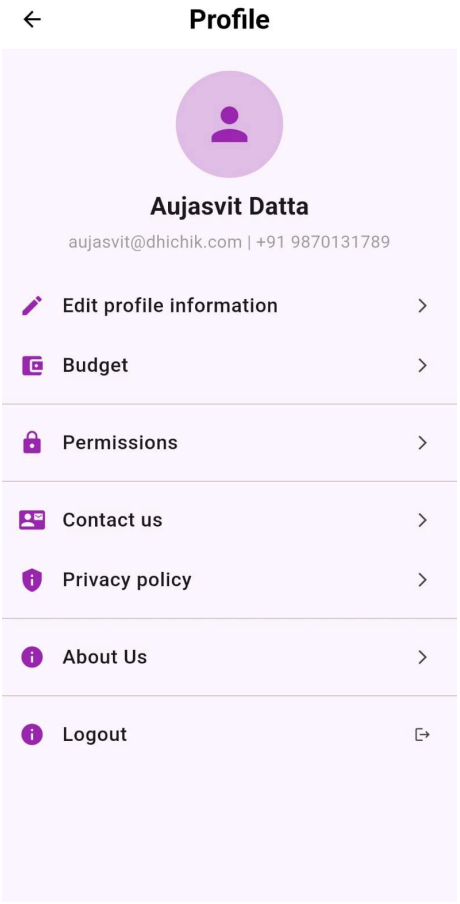
Structural Coverage: Tests were conducted to ensure the chart accurately reflects transaction data across various categories. Scenarios included adding, deleting, and updating transactions, and observing real-time changes in the radial chart. Edge cases such as zero transactions and high-value inputs were also tested to confirm consistent chart behavior and rendering.

Additional Comments: The chart responded correctly to all tested inputs, with smooth visual updates and accurate data representation. The radial segments were proportionate to category-wise spending.

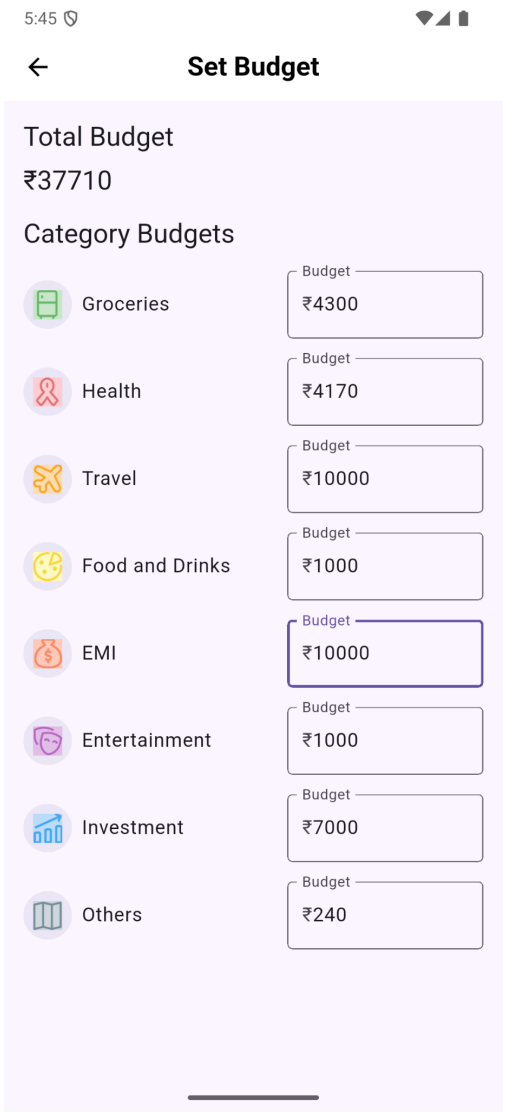
9. Setting Category-wise Budgets in the profile section:

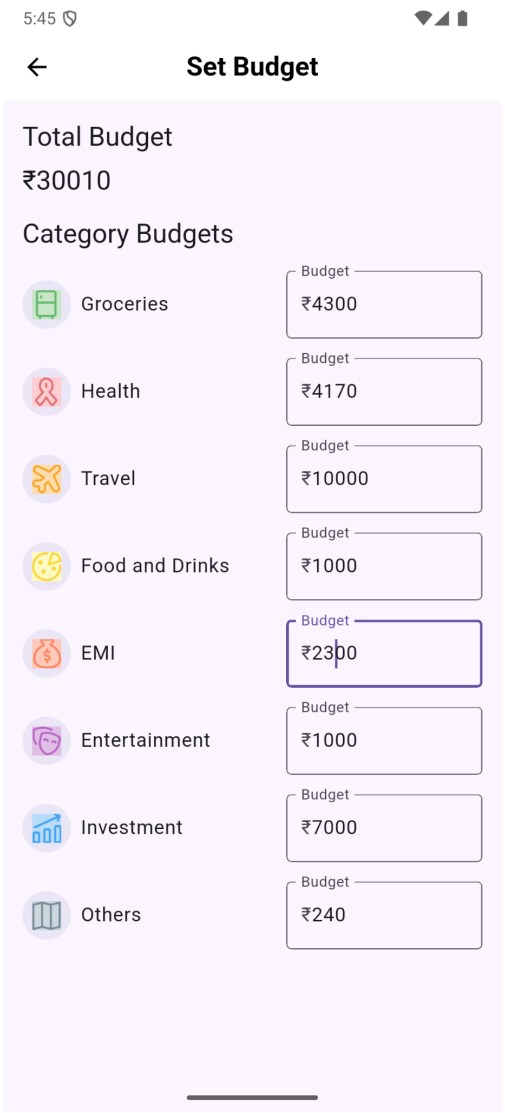
This module tests the features that allow the modification of category-wise budgets.

Unit Details: Profile section
Test Owner: Shrey Solanki
Test Date: 31/03/25 - 31/03/25
Test Results: No bugs were found. This module works perfectly fine.

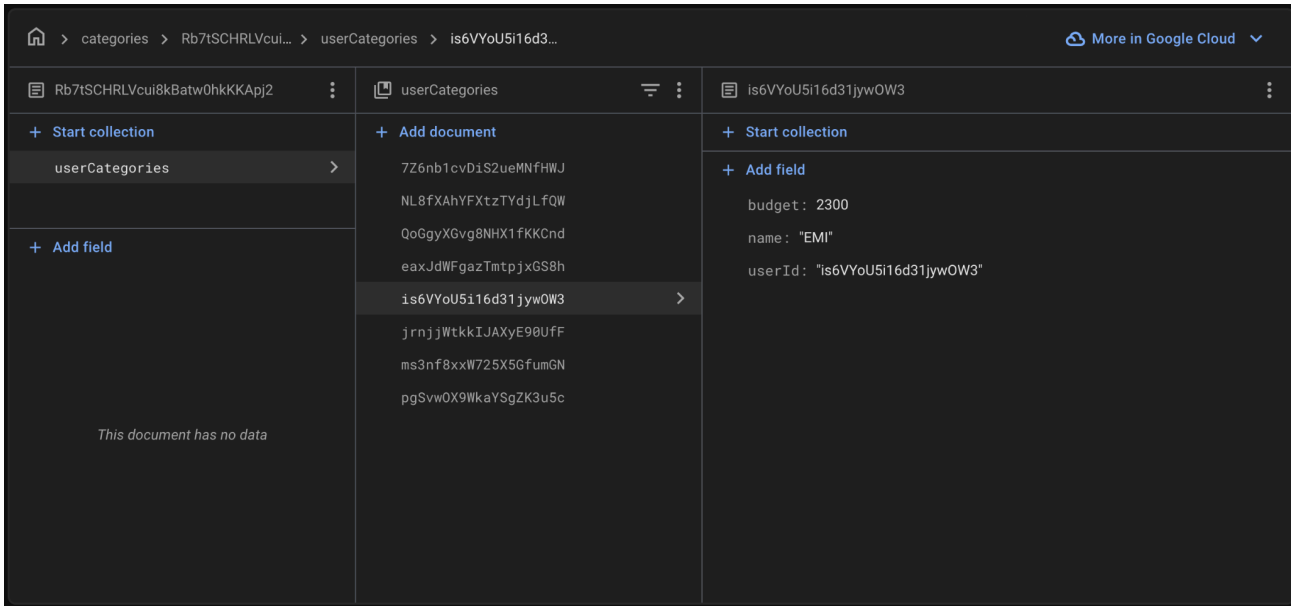


The budget option of the profile page is expected to allow the modification of category-wise budgets.





Category-wise modification of the budget was successfully reflected on the total budget.



The database also reflects the performed modifications.

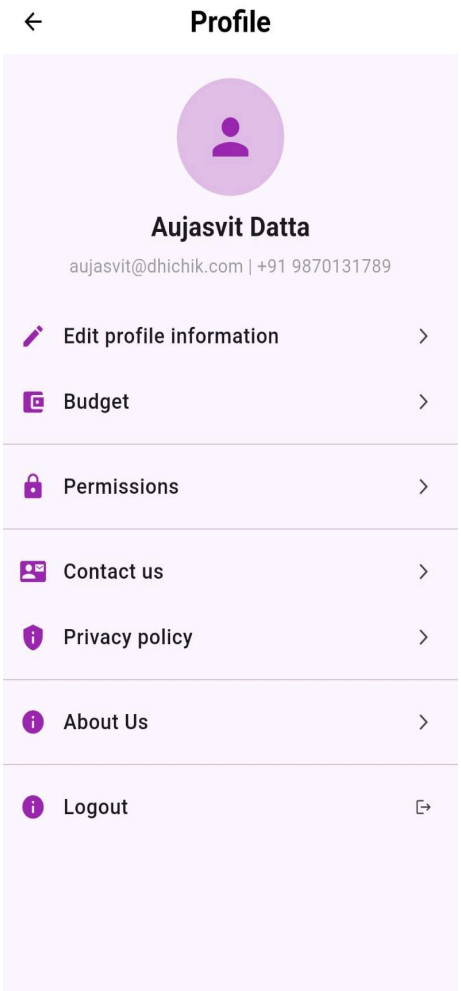
Structural Coverage: All functionalities related to modifying category-wise budgets from the Profile section were tested. This includes editing individual category budgets, saving changes, and verifying immediate updates in both the UI and database. Input validation and state retention across navigation were also tested to ensure consistent behavior.

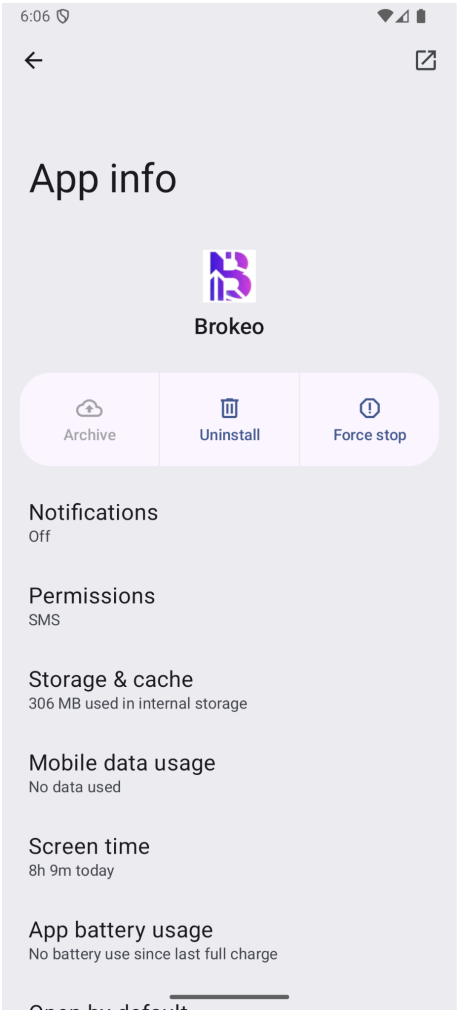
Additional Comments: The module performed as expected, with changes correctly updating in real-time and persisting after app navigation.

10. Profile page Permissions:

This module allows users to manage app permissions directly from the Profile section. By selecting the 'Permissions' option, users are seamlessly redirected to the App Info screen, where they can grant or modify the necessary permissions. This ensures a streamlined and intuitive process for maintaining app functionality and user control.

Unit Details: Profile page
Test Owner: Aujasvit Dutta
Test Date: 01/04/2025 - 01/04/2025
Test Results: This permissions module of Brokeo is fully functional. Users can access the App Info screen directly from the Profile section and successfully manage app permissions without any issues.





Clicking on 'Permissions' in the Profile section successfully navigates to the App Info screen, where the required permissions can be granted.

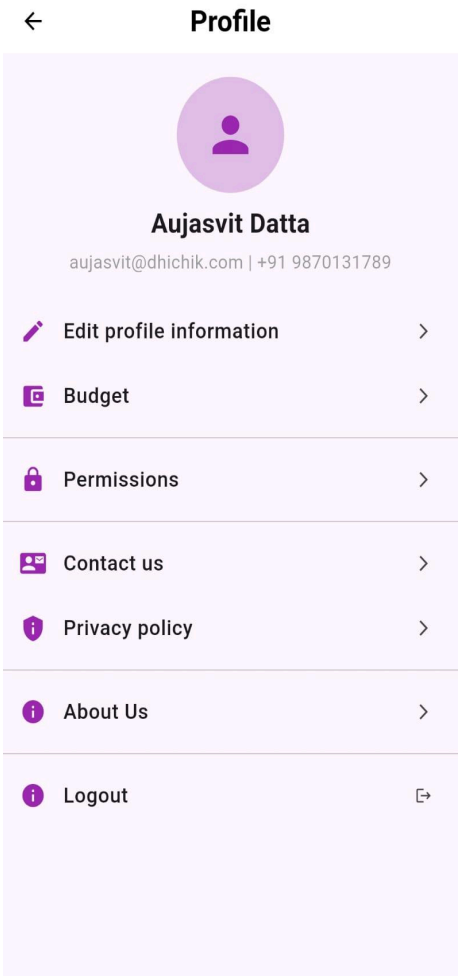
Structural coverage: All navigation and interaction paths were tested, including tapping the 'Permissions' button, redirection to the App Info screen, and the ability to grant or modify permissions. Edge cases such as denied permissions, toggling permissions multiple times, and device-level permission restrictions were also tested to ensure reliable behavior.

Additional Comments: The redirection was smooth and consistent across devices. The module provided a direct and effective way for users to manage app permissions without needing to leave the app context. No bugs were encountered.

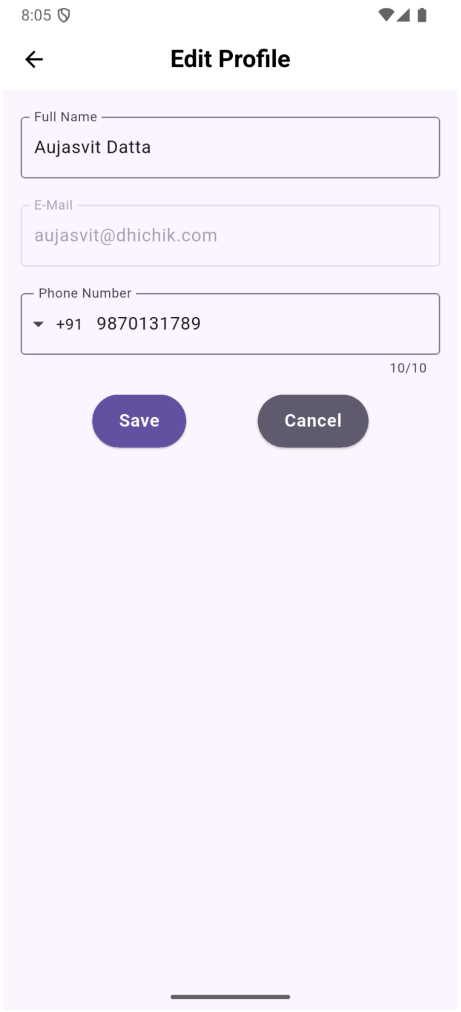
11. Editing Profile Information:

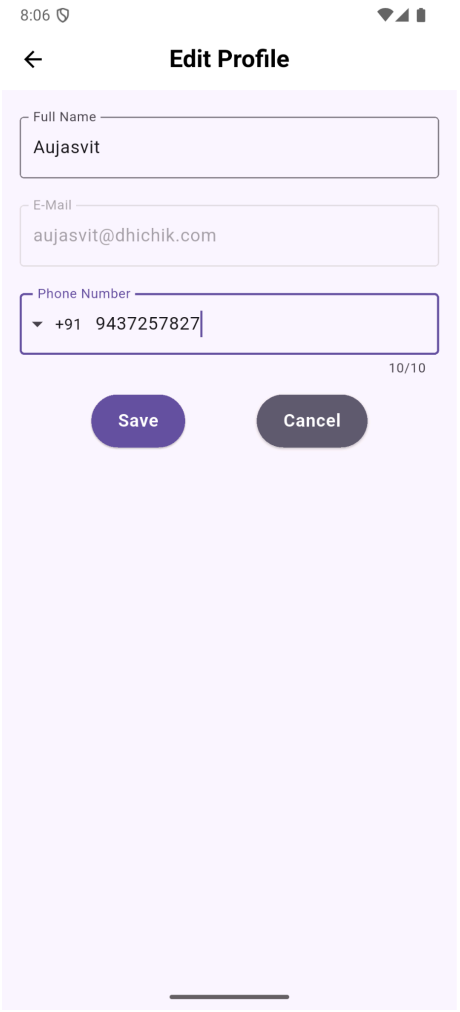
This module allows users to edit their personal information, specifically their name and phone number, through the Profile page. Upon clicking 'Edit Profile Information,' users can input the desired details. The system enforces validation to ensure the phone number entered is in a valid format. Once submitted, the updated information is displayed in real-time on the Profile page and accurately updated in the database, ensuring consistency across the application.

Unit Details: Profile page
Test Owner: Darshan Sethia
Test Date: 01/04/2025 - 01/04/2025
Test Results: The profile editing module is fully functional.

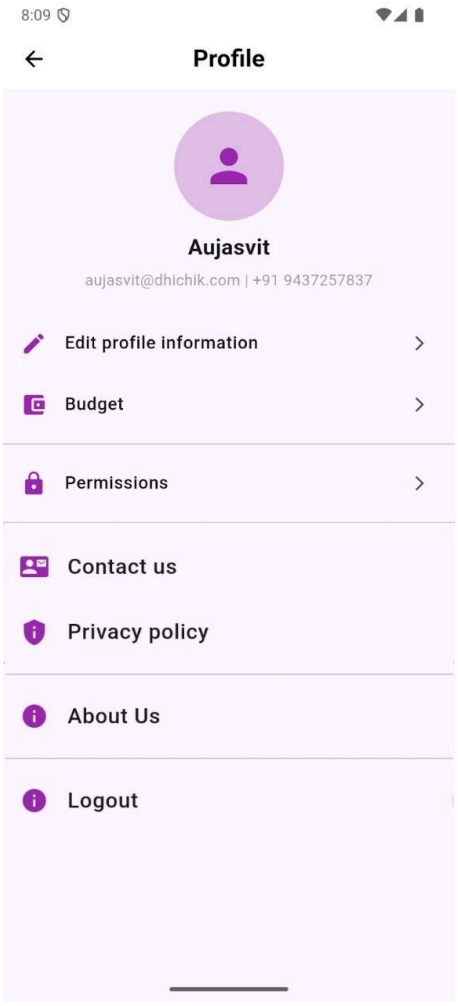


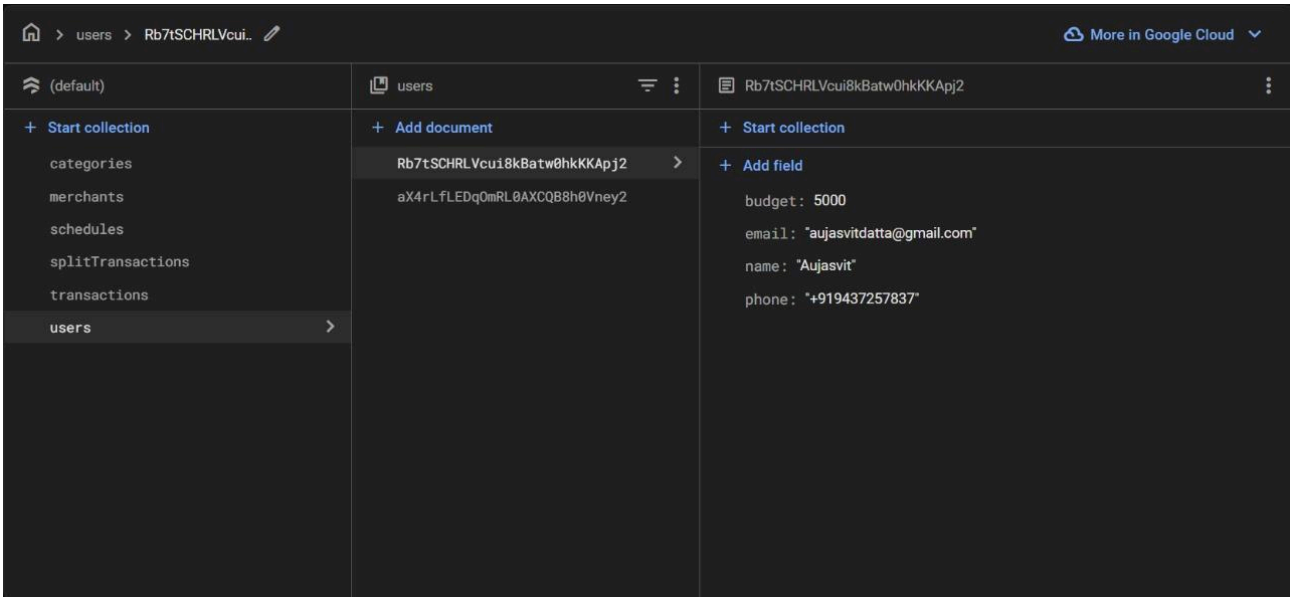
Users can edit their name and phone number by clicking on 'Edit Profile Information' within the Profile page.





Users must enter the desired name and phone number, and the app prevents progression unless a valid phone number is provided.


8:09
Profile
Aujasvit
aujasvit@dhichik.com | +91 9437257837
Edit profile information
Budget
Permissions
Contact us
Privacy policy
About Us
Logout


users
Rb7tSCHRLVcui..
More in Google Cloud
(default)
Start collection
categories
merchants
schedules
splitTransactions
transactions
users
users
Add document
Rb7tSCHRLVcui8kBatw0hkKKApj2
aX4rLfLEDqOmRL0AXCQB8h0Vney2
Rb7tSCHRLVcui8kBatw0hkKKApj2
Start collection
Add field
budget: 5000
email: "aujasvitdatta@gmail.com"
name: "Aujasvit"
phone: "+919437257837"

The updated information is correctly displayed on the Profile page and successfully updated in the database.

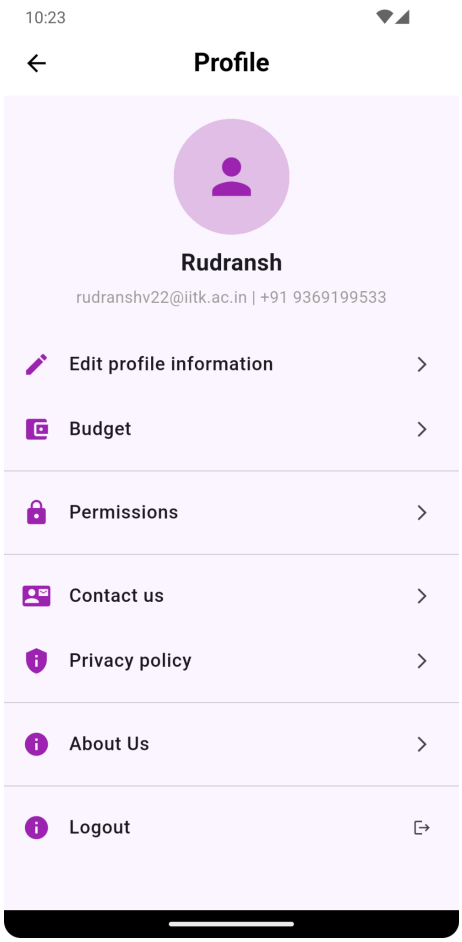
Structural coverage: All relevant flows were tested, including editing name and phone number, validating phone number format, and confirming updates in both the UI and database. Scenarios included valid input, empty fields, invalid phone numbers, and repeated edits. The UI's responsiveness and data consistency across sessions were also verified.

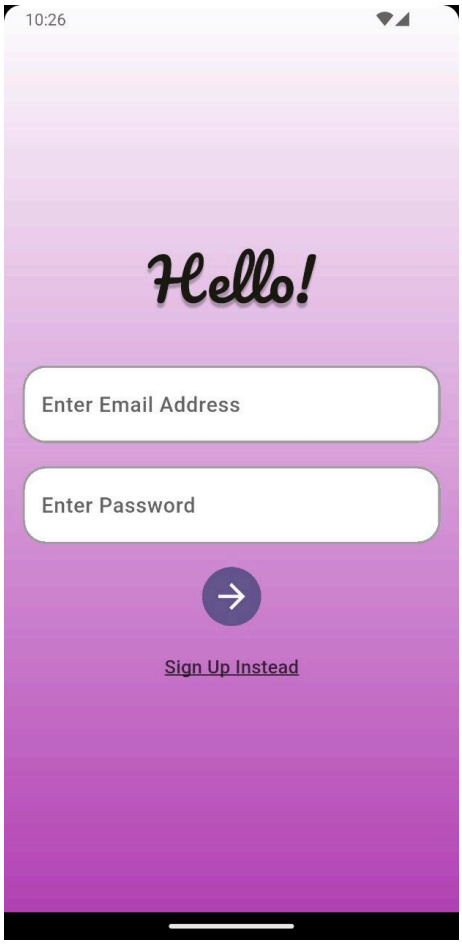
Additional Comments: The input validation effectively prevented invalid submissions, and updates were reflected instantly on the Profile page. Data persisted correctly after app restarts, indicating successful database integration.

12. Logging Out:

This module enables users to securely log out of the application through the Profile page. Upon selecting the 'Logout' option, the user session is terminated and the app redirects them to the Login page, ensuring a smooth and secure exit from the account.

Unit Details: Logout page
Test Owner: Marisha Thorat
Test Date: 01/04/2025 - 01/04/2025
Test Results: The logout functionality is fully operational. Users are able to successfully log out of the application from the Profile page without any issues. Session data is cleared, and users are redirected to the login screen.





The user can log out by navigating to the Profile page and selecting the 'Logout' option, which successfully redirects them to the Login page.

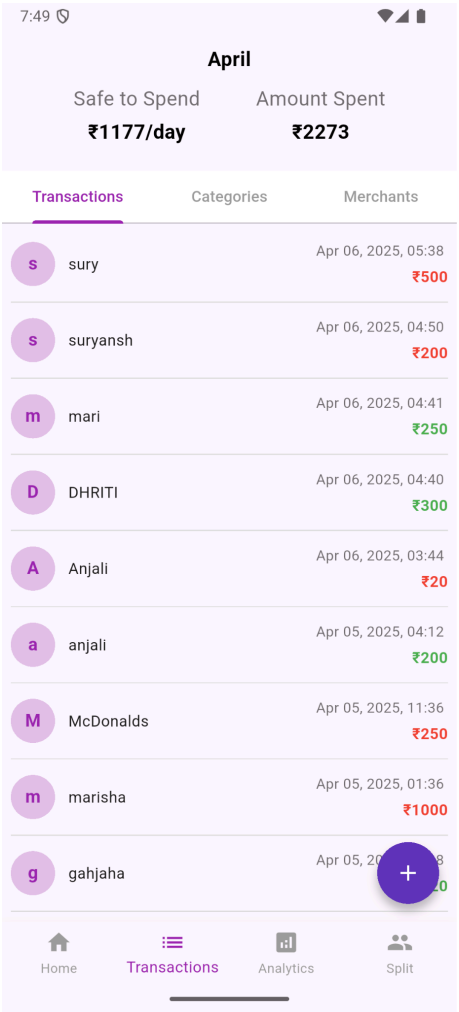
Structural Coverage: The test covered all major flows including navigating to the Profile page, triggering the logout action, and confirming session termination. Edge cases such as repeated logouts were also tested.

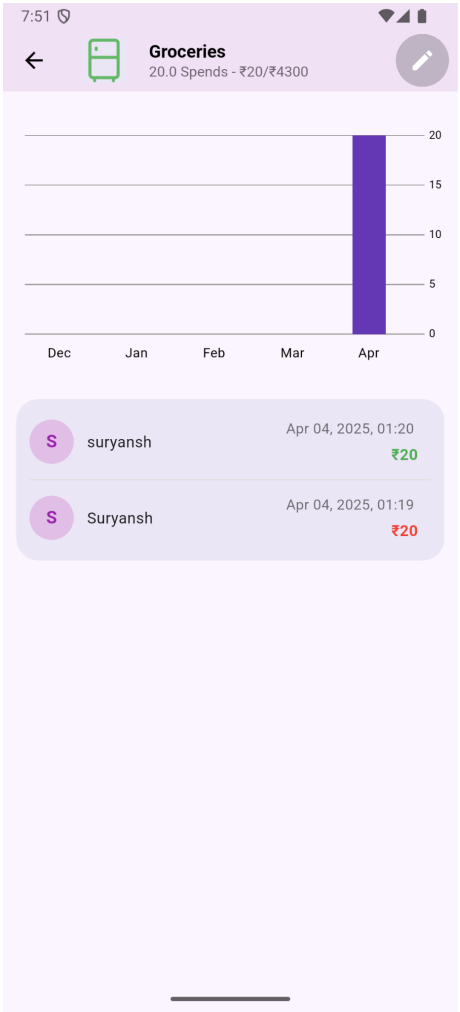
Additional Comments: The user can log out by navigating to the Profile page and selecting the 'Logout' option. The app handled session cleanup and redirection effectively.

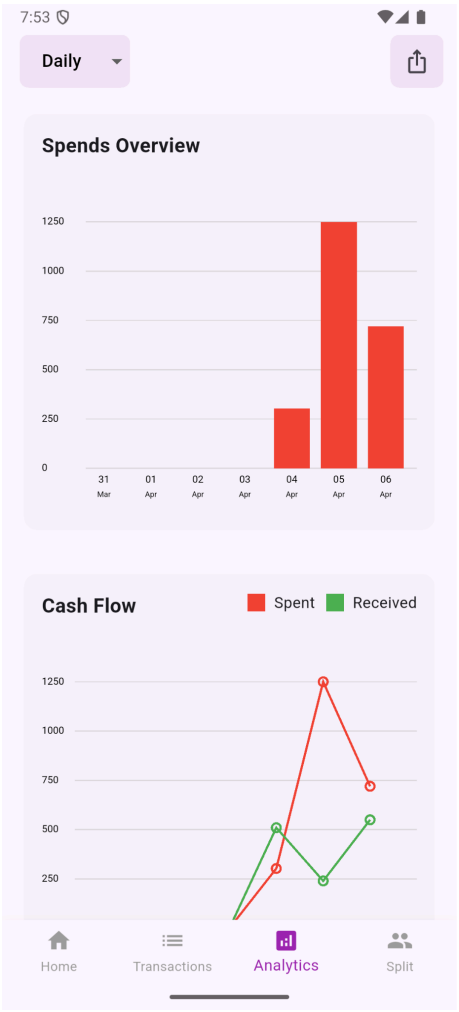
3 Integration Testing

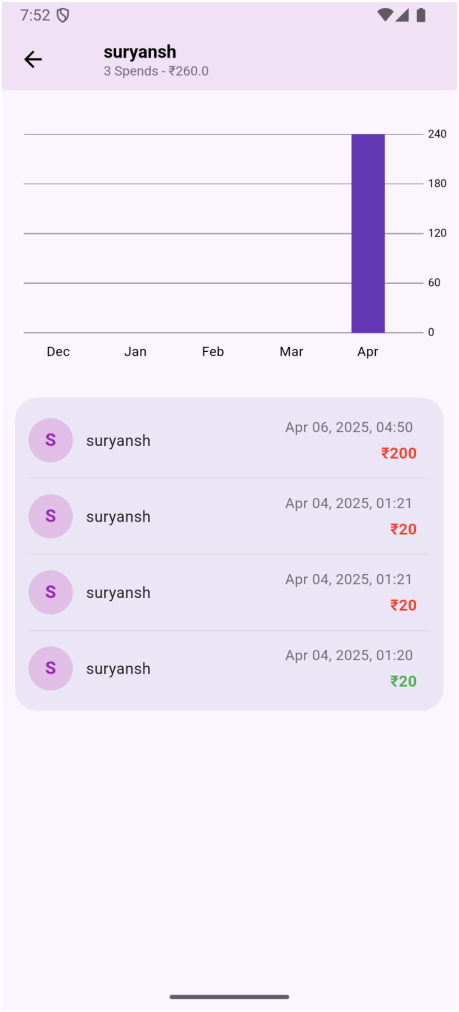
Reflection of modification of transactions across the Merchants, Analytics, and Category tabs:

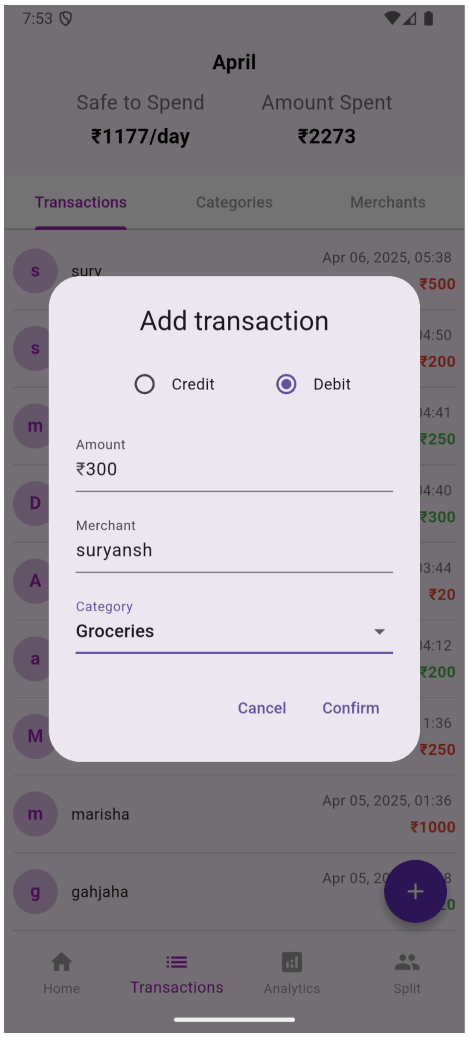
Module Details: This module tests the reflection of addition and deletion of transactions across the merchants, analytics and category tabs.
Test Owner: Rudransh Verma
Test Date: 01/04/2025 - 01/04/2025
Test Results: This functionality works perfectly fine. All changes are being properly reflected on their particular tabs.
Transactions, category,analytics and category tabs before adding the transaction:



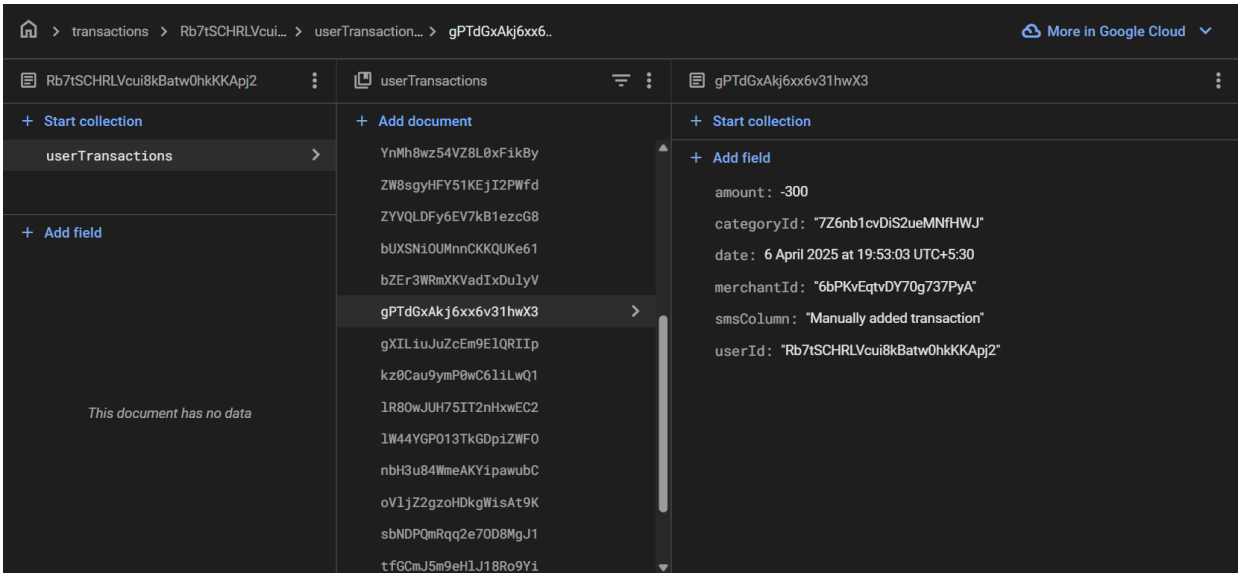



7:53
Daily
Spends Overview
1250
1000
750
500
250
0
31 Mar
01 Apr
02 Apr
03 Apr
04 Apr
05 Apr
06 Apr
Cash Flow
Spent
Received
1250
1000
750
500
250
Home
Transactions
Analytics
Split

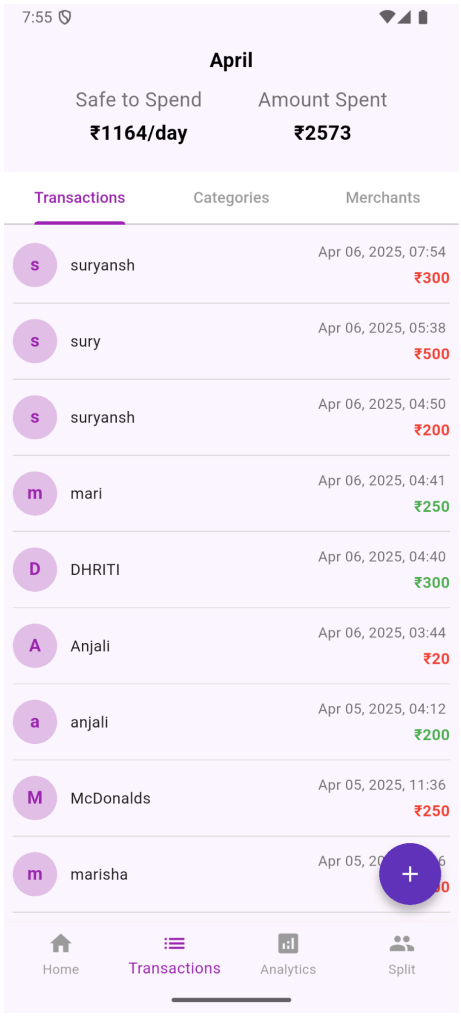

7:52
suryansh
3 Spends - ₹260.0
240
180
120
60
0
Dec
Jan
Feb
Mar
Apr
S
suryansh
Apr 06, 2025, 04:50
₹200
S
suryansh
Apr 04, 2025, 01:21
₹20
S
suryansh
Apr 04, 2025, 01:21
₹20
S
suryansh
Apr 04, 2025, 01:20
₹20

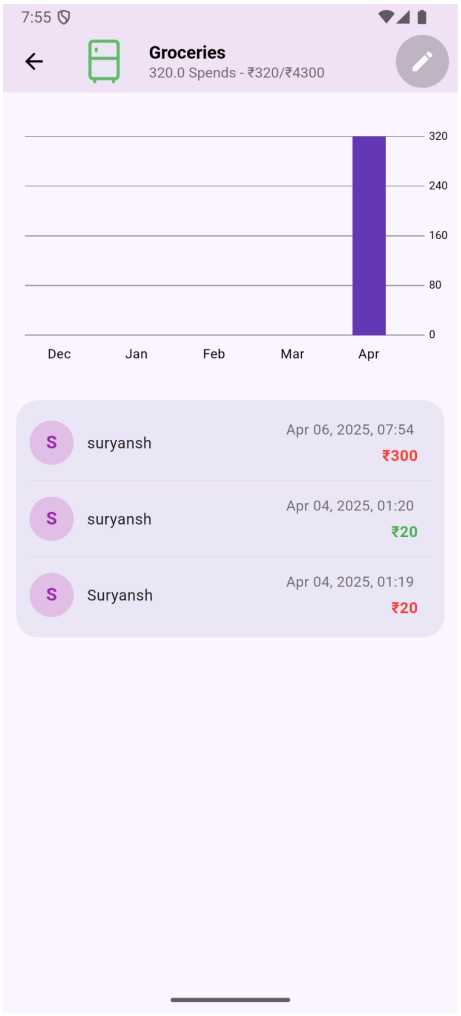


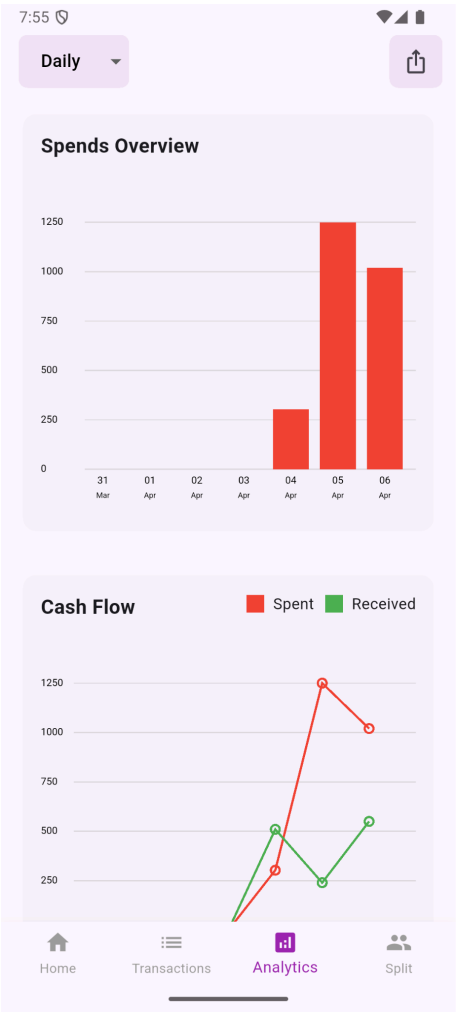
Adding the transaction
Reflection in the database:



The transactions, merchants, analytics and category pages after adding the transaction:







The weekly, monthly analytics were verified too.

Structural Coverage: This test covers the end-to-end reflection of transaction modifications (addition and deletion) across interconnected modules: Merchants, Analytics, and Category tabs. The integration test ensures that updates are dynamically and accurately propagated to all relevant components. Validation was performed on both UI and backend (database) states. Weekly and monthly analytics views were explicitly tested to verify correct aggregation post-transaction updates.

Additional Comments: All changes made to transactions are correctly reflected in the database and the corresponding UI modules. The system maintains real-time consistency across all affected tabs.

However, integration testing for SMS parsing and Split functionality is pending due to unresolved bugs in those modules.

4 System Testing

1. **Requirement:** Users can create an account using the sign-up page

   The test owner created a new account using the signup page, clicked on the verification link which was sent to their email address, and their account was added to the database.

Test Owner: Suryansh Verma
Test Date: 04/02/2025 - 04/02/2025
Test Results: Test passed.
Additional Comments: Upon opening the app for the first time, users can enter their name, email address and password to make an account on Brokeo. To verify the email address of the user, the system will send a verification link to the user's email address, which they can click on to verify.

2. **Requirement:** Users can view their details, including their name, phone number, and email address, and edit their name, and phone number.

   The test owner tried modifying the name and phone number, and it was reflected in the database successfully.

Test Owner: Rudransh Verma
Test Date: 05/02/2025 - 05/02/2025
Test Results: Test passed.

3. **Requirement:** The application includes a help section containing a user manual to guide users in understanding and utilizing its features, and an "About Brokeo" section, providing information about the application and the development team behind it.

   The test owner tried to view the help and "About Brokeo" section in the profile and successfully accessed it.

Test Owner: Sanjna S
Test Date: 04/02/2025 - 04/02/2025
Test Results: Test passed.

4. **Requirement:** The user should have the option to sign out of their account

   The test owner tried signing out of the account using the sign-out button on the profile section and successfully signed out

Test Owner: Aujasvit Dutta
Test Date: 04/02/2025 - 04/02/2025
Test Results: Test passed.

5. **Requirement:** Automatic Detection of transactions through SMS

   The test owner tried sending an sms on the phone, an api call was sent to the backend where the gemini api server was running, got a response, but the transaction did not get added to the database, which it was supposed to.

Test Owner: Darshan Sethia
Test Date: 05/02/2025 - 05/02/2025

Test Results: Test failed.
Additional Comments: When an sms was sent to the user's phone, it was supposed to be added to the database automatically. We are currently working on resolving this.

6. **Requirement:** Users can add and delete transactions

   The test owner successfully added both credit and debit transactions, with each entry dynamically updating the radial progress bar to reflect the current financial status. An option to assign a relevant spending category to each transaction was also provided. The deletion of existing transactions was accurately reflected in the radial progress bar.

Test Owner: Dhriti Barnwal
Test Date: 04/02/2025 - 04/02/2025
Test Results: Test Passed

7. **Requirement:** Users can view their spending across various categories. They can also set a monthly budget for each spending category and access detailed insights into budget usage, including the amount spent and the remaining budget.

   This feature was tested by adding budgets to various categories, modifying them, and verifying the display of spending across the respective categories.

Test Owner: Anjali Patra
Test Date: 04/02/2025 - 04/02/2025
Test Results: Test Passed

8. **Requirement:** Users can modify a category to any merchant.

   The test owner tried to edit the category of a merchant and was successful in their attempt.

Test Owner: Suryansh Verma
Test Date: 04/02/2025 - 04/02/2025
Test Results: Test Passed

9. **Requirement:** Users can view transactions organized by merchants, with a breakdown of all transactions associated with each merchant.

   Upon selecting a merchant, the system displays the total money paid to the merchant and the number of spends made during the current month. These metrics are also presented in a graphical format.

Test Owner: Shrey Solanki
Test Date: 04/02/2025 - 04/02/2025
Test Results: Test Passed

10. **Requirement:** Users will get a detailed analytics of their spending

    Successfully accessed analytics of spending in the analytics page in the form of bar graphs and line graphs.

Test Owner: Marisha Thorat
Test Date: 04/02/2025 - 04/02/2025

Test Results: Test Passed

11. Requirement: Users will be allowed to split transactions with other users.

Split of a transaction is partially written to the database. As a result, some users see the transaction while others don’t, leading to mathematically incorrect balances.

Test Owner: Bhavnoor Singh
Test Date: 05/02/2025 - 05/02/2025
Test Results: Test Failed.
Additional Comments: This feature is currently being worked on resolving so we did not test it thoroughly.

12. Requirement: Users will be able to export their expense report

Tried to export the expense report using the export button in the analytics page, and the system downloaded a CSV file which contained a summary of their spending.

Test Owner: Dhriti Barnwal
Test Date: 04/02/2025 - 04/02/2025
Test Results: Test Passed.

5 Conclusion

As far as we can see, the testing was sufficiently exhaustive and very effective. There are a few major bugs in the automatic detection of transactions through sms and the split page, which we are currently diligently working on resolving, so it will be tested after being implemented. We also encountered some errors when implementing multiple users, but it was fixed after a few iterations.

The testing process can be improved by having a larger number of students in the team and dividing the team into groups so as to reduce the amount of communication between the developers and testers of each feature.

There is also a conflict of interest in the testing process if the testers and developers are part of the same group, so no developer tested a feature that they had developed. Separating the two groups and the assessments of their work would incentivise testers to think as end-users rather than custodians of the software. In fact, this is standard practice in the certification of safety-critical software, i.e. independent regulators certify systems as safe.

Giving more time to the testing process would also have improved the testing process. However, as we had limited time, implementation was given priority, with as exhaustive testing as possible.

Appendix A - Group Log

Date	Timings	Duration	Minutes
29th Mar	16:00-17:00	1 hr	• Started preparing User Manual
30th Mar	16:00-17:00	1 hr	• Fixed the bugs found on individual checks
31st Mar	18:00-20:30	2 hrs 30 mins	• Fixed bugs in the application • Started Unit Testing of the application
1st Apr	21:00-23:00	2 hrs	• Unit Testing and Integration Testing
2nd Apr	21:30-04:00	6 hrs 30 mins	• Integration Testing and System Testing of the Application • Fixed bugs • Completion of User Manual • Final Proofreading of User Manual
3rd Apr	22:00-01:00	3 hrs	• Completion of User Manual • Final Proofreading of User Manual
4th Apr	22:00-01:00	3 hrs	• Started working on Test Document
5rd Apr	20:00-23:00	3 hrs	• Fixed some minor bugs
6th Apr	20:00-23:00	3 hrs	• Updated pictures in the Test Document with final testing • Completion of Test Document • Final Proofreading of the Document