Verification and Validation Report: Housemates

Team #9, Housemates
Justin Dang - dangj15
Harris Hamid - hamidh1
Fady Morcos - morocof2
Rizwan Ahsan - ahsanm7
Sheikh Afsar - afsars

March 6, 2024

1 Revision History

Date	**Version**	**Notes**
March 6	1.0	Version for Rev 0

2 Symbols, Abbreviations and Acronyms

symbol	**description**
API	Application Programming Interface
P	Pass
F	Fail

Contents

List of Tables

List of Figures

3 Purpose

This document describes the results of verification and validation plan that was created earlier on in the software lifecycle of Housemates. The purpose of this verification and validation is

1. Ensure that the requirements listed in the SRS for the Housemates application are correct.
2. Ensure that the design of the Housemates application satisfies the requirements listed in the SRS.
3. Ensure the usability of the Housemates application for its prospective users.

To find the the details of each test look at the VnV Plan and search for the associated test id.

4 Functional Requirements Evaluation

4.1 System Tests

The details of the system tests can be found in section 3.1 of the VnV Plan.

4.1.1 Bill Management

Test ID	P	F
test-BM1-1/2-1	×	
test-BM3-1	×	
test-BM4-1	×	
test-BM5-1	×	
test-BM6-1	×	
test-BM7-1	×	

Table 1: **Bill Management System Tests**

All bill management system tasks passed as expected.

4.1.2 Task Management

Test ID	P	F
test-TM1-1	×	
test-TM2-1	×	
test-TM3-1	×	
test-TM4-1	×	
test-TM5-1	×	

Table 2: **Task Management System Tests**

All task management system tasks passed as expected.

4.1.3 Scheduling

Test ID	P	F
test-SS1-1	×	
test-SS2-1	×	
test-SS3-1	×	

Table 3: **Scheduling System Tests**

All scheduling system tasks passed as expected.

4.1.4 Account

Test ID	P	F
test-AS-1-1	×	
test-AS-1-2	×	
test-AS-1-3	×	
test-AS-2	×	
test-AS-3	×	
test-AS-4	×	
test-AS-5	×	

Table 4: **Account System Tests**

All account system tasks passed as expected.

4.2 Unit Testing

The details of the unit tests can be found in section 4 of the VnV Plan.

4.2.1 Automated Testing

Automated testing was conducted using jest and GitHub actions. Automated testing mainly covers the unit testing of Housemates and is performed automatically whenever a commit is added to the main branch. The purpose of automated testing of Housemates is to insure that any changes made to the main branch do not break the overall functionality of Housemates.

4.2.2 Bill Management

Test ID	P	F
UT-B1	×	
UT-B2	×	
UT-B3	×	

Table 5: **Bill Management Unit Tests**

All unit tests for bill management module passed as expected.

4.2.3 Task Management

Test ID	P	F
UT-T1	×	
UT-T2	×	
UT-T3	×	
UT-T4	×	

Table 6: **Task Management Unit Tests**

All unit tests for task management module passed as expected.

4.2.4 Scheduling

Test ID	P	F
UT-S1	×	
UT-S2	×	
UT-S3	×	
UT-S4	×	
UT-S5	×	

Table 7: **Scheduling Unit Tests**

All unit tests for scheduling module passed as expected.

4.2.5 Account

Test ID	P	F
UT-A1	×	
UT-A2	×	
UT-A3	×	
UT-A4	×	

Table 8: **Account Unit Tests**

All unit tests for account module passed as expected.

5 Nonfunctional Requirements Evaluation

5.1 Usability

The usability of the app was evaluated by conducting a survey with five participants who were asked to complete basic tasks within the app and provide feedback on their experience. This survey can be found in section 5.2 of the VnV Plan. The survey consisted of 13 questions, including multiple-choice, rating scale, and open-ended questions. The main findings from the survey are summarized below along with some key data quantified:

- User Profile: All the participants were students who lived with roommates or housemates and often split their bills with other people.

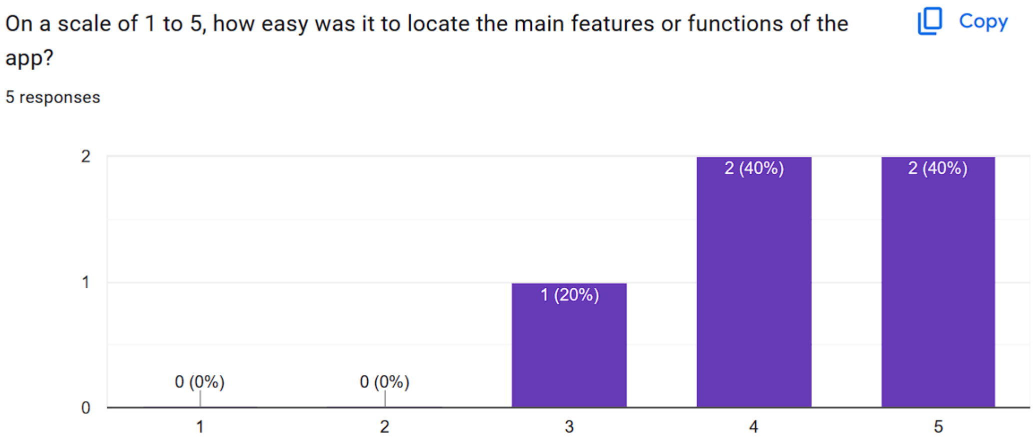
- First Impression: None of the participants found the app to be visually appealing, and most of them described it as bland, generic, or unpolished. They also had difficulty locating the main features or functions of the app.

- Performance: The participants were generally satisfied with the speed and performance of the app, but some of them encountered errors or bugs while using it, such as not being able to create a user or join a group.

- Feedback and Guidance: The app did not provide helpful feedback or guidance when the participants made a mistake or encountered an issue.

Most of them did not know what to do when they faced a problem and had to refresh the page or try again.

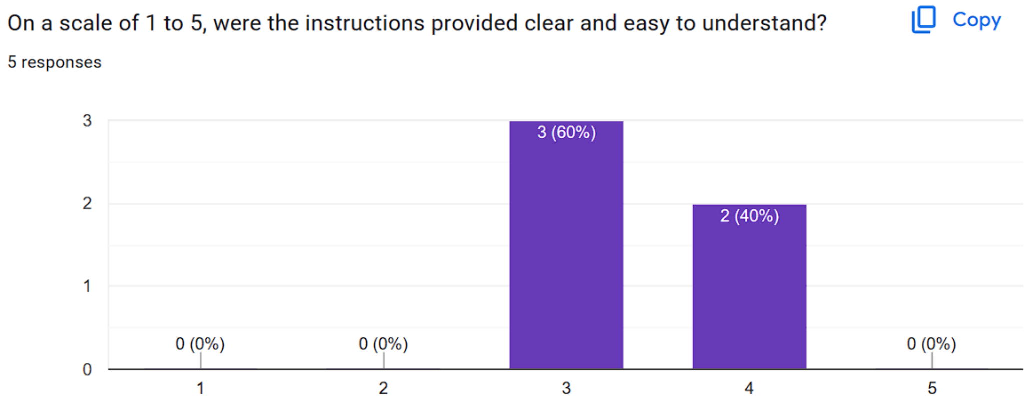
- Difficulty: The participants found some features or functions to be particularly difficult to use or understand, such as joining a group, splitting an expense, or paying off a bill. They also complained about the amount of typing required and the lack of information or options available.



On a scale of 1 to 5, how easy was it to locate the main features or functions of the app?

5 responses

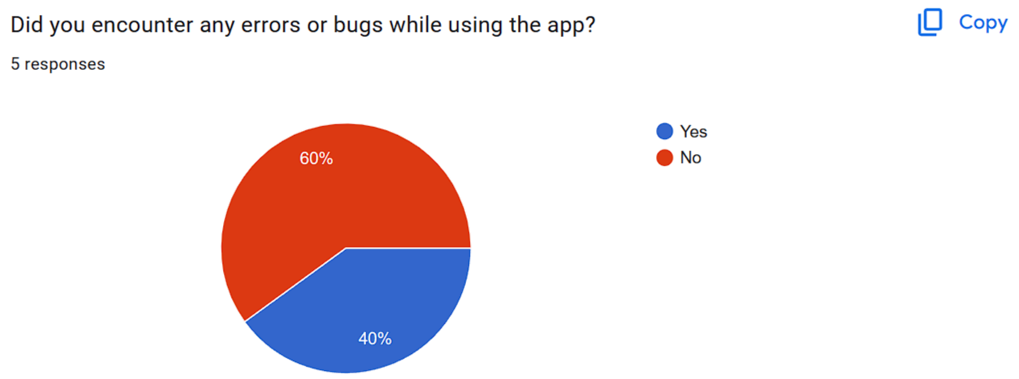
Figure 1: Ease of locating main features



On a scale of 1 to 5, were the instructions provided clear and easy to understand?

5 responses

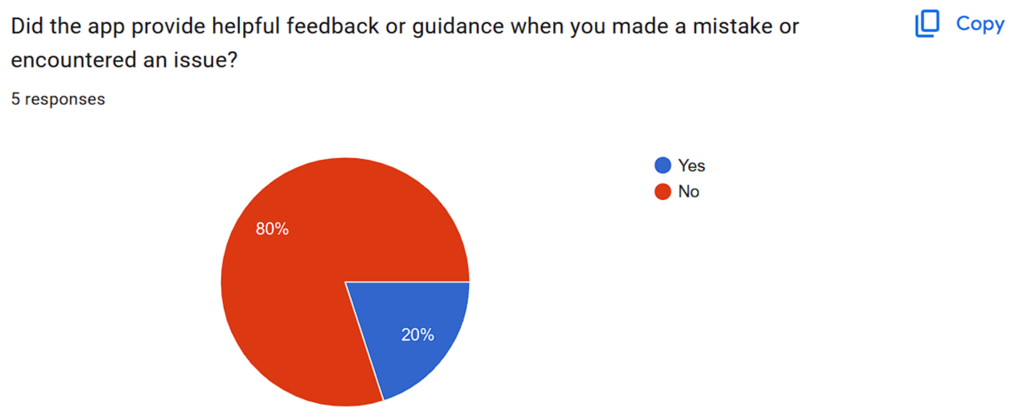
Figure 2: Clear instructions



Did you encounter any errors or bugs while using the app?

5 responses

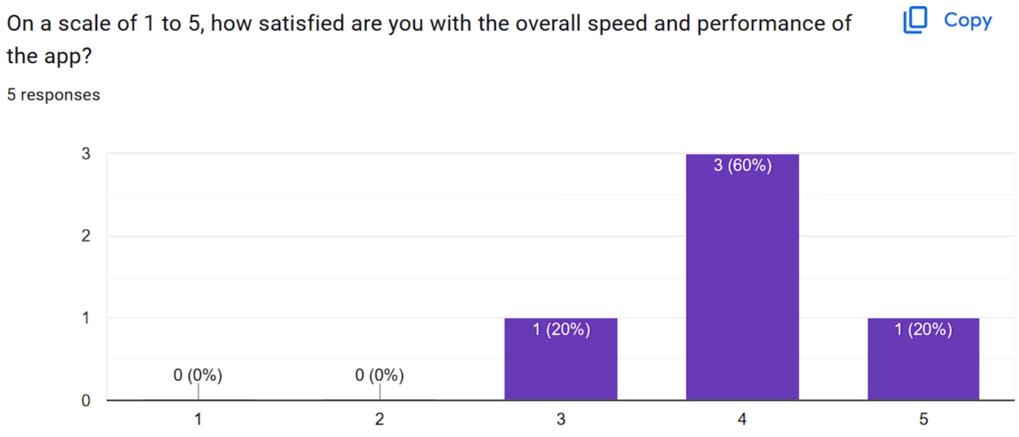
Figure 3: Bugs encountered



Did the app provide helpful feedback or guidance when you made a mistake or encountered an issue?

5 responses

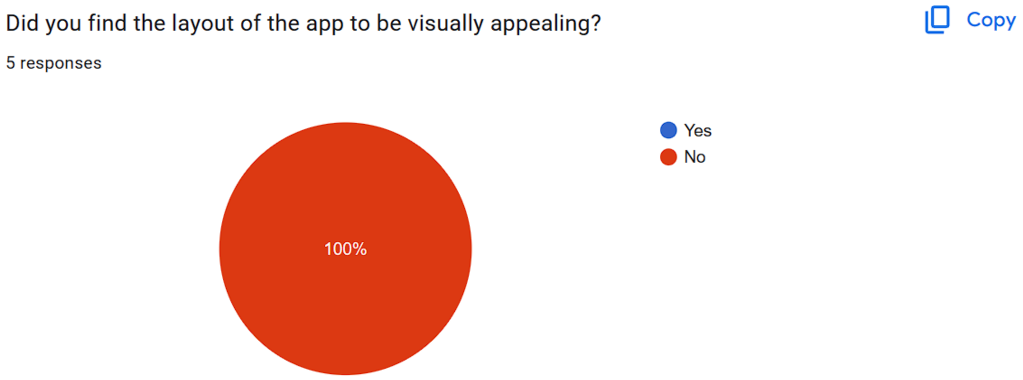
Figure 4: App feedback to user errors



On a scale of 1 to 5, how satisfied are you with the overall speed and performance of the app?

5 responses

Figure 5: Performance of app



Did you find the layout of the app to be visually appealing?

5 responses

Figure 6: Visual appeal

5.2 Performance

Performance testing for Housemates was focused testing the back-end server of the application. This was done using Jmeter, which is a load testing tool that helps analyze the performance of the backend API of Housemates. The test plan involved creating 100 users for Housemates and having them make common requests to the backend server API (e.g getting user information after logging in). The results of performance testing can be seen in the graph below.

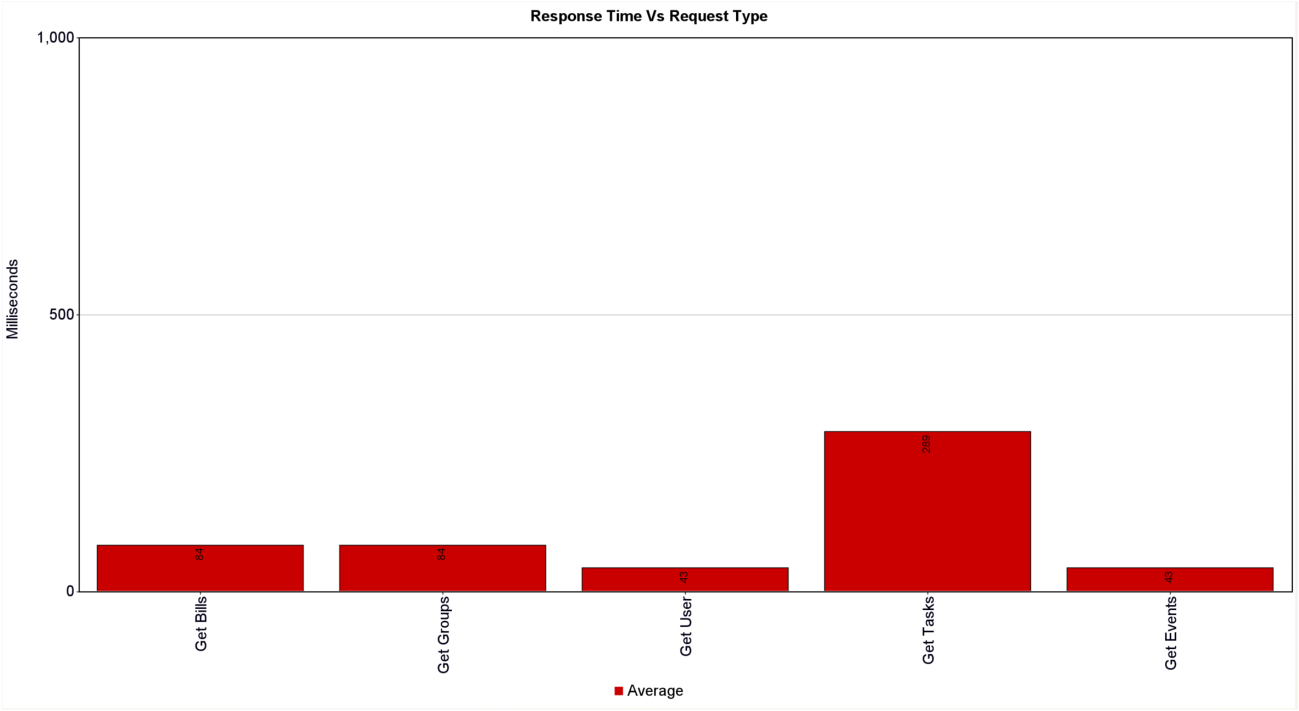


Figure 7: Average Response Times of Housemates API

As can be seen through the above graph the response times to user requests is quite fast ranging from around 40 - 300 ms. This is fast enough so that there should not be noticeable delay in Housemates on the user end. As such this indicates to us that Housemates can handle 100 concurrent users and as a result we concluded that the performance of Housemates is sufficient for our current purposes. In the future if necessary the database of Housemates could be upgraded in order to handle more users.

5.3 Other Non-functional Tests

The details of these non-functional tests can be found in section 3.2 of the VnV Plan.

Test ID	P	F
test-LF-A1-1		×
test-LF-A1-2		×
test-UH-E1-1	×	
test-UH-E1-2	×	
test-UH-P1	×	
test-UH-L1-1		×
test-UH-L1-2		×
test-UH-A1	×	
test-P-SL1	×	
test-P-PA1	×	
test-P-RFT1		×
test-P-C1	×	
test-OE-PE1	×	
test-OE-PR1		×
test-M-M1	×	
test-S-A1	×	
test-S-IN1	×	
test-S-P1	×	
test-C-SC1	×	

Table 9: **Non-Functional System Tests**

For the non-functional system tests most passed, but some did not. test-OE-PR1 failed because Housemates isnt available on Google Play Store, test-P-RFT-1 failed because Housemates doesn't work well offline yet, test-LF-A1-1, test-LF-A1-2, test-UH-L1-1, and test-UH-L1-2 failed because of reasons described in usability section of this VnV report.

6 Conclusions based off VnV Data

The main area of improvement for Housemates is in the usability department. As evidenced from the data from the usability survey most users are not satisfied with the current state of the Housemates application. The non-functional tests that failed also mainly have to do with the usability of Housemates. As such, the main focus for revision 1 of Housemates will be on improving the usability of Housemates. The changes that we plan to implement with revision 1 of the Housemates application are covered in the next section.

7 Changes Due to Testing

7.1 Planned Changes due to Revision 0 Feedback

For the Revision 0 demo, the main piece of feedback that we received was that the user interface of the Housemates felt very unpolished (e.g. dollar amounts not being rounded correctly, requiring a lot of typing which is undesirable for a mobile application) and that users were unlikely to use Housemates in this state. To help address these issues we plan to improve the UI of all the main features of Housemates (Bill Management, Task Management, Scheduling) by presenting the user information in a card layout, which is more user-digestible. This new UI will also require less typing, which should make it more efficient to the end user. Additionally, some features were suggested such as having presets for tasks and reoccurring tasks that we plan to implement in revision 1 of Housemates.

7.2 Planned Changes due to Usability Feedback

One of the major feedback we got was regarding the design and layout of the application. Comments included phrases like ”bland”, ”generic” and ”not eye catching”. So for revision 1 we plan to improve the current layout by introducing a more desirable color scheme rather than being black and white. Another pain point described by the user was that the application did not provide feedback if a mistake was made by the user. To remedy this problem, instead of providing feedback to the problem, we will reduce the chance of the user going into that state by introducing more guards in our application.

8 Trace to Requirements

Test ID	TM1	TM2	TM3	TM4	TM5	BM1	BM2	BM3	BM4	BM5
test-TM1-1	×									
test-TM2-1		×								
test-TM3-1			×							
test-TM4-1				×						
test-TM5-1					×					
test-BM1-1/2-1						×	×			
test-BM3-1								×		
test-BM4-1									×	
test-BM5-1										×

Table 10: **Functional Requirements Traceability Part 1**

Test ID	BM6	BM7	SS1	SS2	SS3	AS1	AS2	AS3	AS4	AS5
test-BM6-1	×									
test-BM7-1		×								
test-SS1-1			×							
test-SS2-1				×						
test-SS3-1					×					
test-AS-1-1						×				
test-AS-1-2						×				
test-AS-1-3						×				
test-AS-2							×			
test-AS-3								×		
test-AS-4									×	
test-AS-5										×

Table 11: **Functional Requirements Traceability Part 2**

Test ID	LF-A1	UH-E1	UH-P1	UH-L1	UH-A1	P-SL1	P-PA1	P-RFT1
test-LF-A1-1	×							
test-LF-A1-2	×							
test-UH-E1-1		×						
test-UH-E1-2		×						
test-UH-P1			×					
test-UH-L1-1				×				
test-UH-L1-2				×				
test-UH-A1					×			
test-P-SL1						×		
test-P-PA1							×	
test-P-RFT1								×

Table 12: **Non-Functional Requirements Traceability Part 1**

NFR ID	P-C1	OE-PE1	OE-PR1	M-M1	S-A1	S-IN1	S-P1	C-SC1
test-P-C1	×							
test-OE-PE1		×						
test-OE-PR1			×					
test-M-M1				×				
test-S-A1					×			
test-S-IN1						×		
test-S-P1							×	
test-C-SC1								×

Table 13: **Non-Functional Requirements Traceability Part 2**

9 Trace to Modules

Module	Task	Bill	Scheduling	Account	Interface	Database	Network	Crypto
test-TM1-1	×				×	×	×	
test-TM2-1	×				×	×	×	
test-TM3-1	×				×	×	×	
test-TM4-1	×				×	×	×	
test-TM5-1	×				×	×	×	
test-BM1-1/2-1		×			×	×	×	
test-BM3-1		×			×	×	×	
test-BM4-1		×			×	×	×	
test-BM5-1		×			×	×	×	
test-BM6-1		×			×	×	×	
test-BM7-1		×			×	×	×	
test-SS1-1			×		×	×	×	
test-SS2-1			×		×	×	×	
test-SS3-1			×		×	×	×	
test-AS-1-1				×	×	×	×	×
test-AS-1-2				×	×	×	×	×
test-AS-1-3				×	×	×	×	×
test-AS-2				×	×	×	×	×
test-AS-3				×	×	×	×	
test-AS-4				×	×	×	×	
test-AS-5				×	×	×	×	

Table 14: **Module Traceability Part 1**

Module	Task	Bill	Scheduling	Account	Interface	Database	Network	Crypto
test-LF-A1-1					×			
test-LF-A1-2					×			
test-UH-E1-1					×			
test-UH-E1-2					×			
test-UH-P1					×			
test-UH-L1-1	×	×	×	×	×			
test-UH-L1-2	×	×	×	×	×			
test-UH-A1					×			
test-P-SL1	×	×	×	×	×			
test-P-PA1		×			×	×		
test-P-RFT1	×	×	×	×	×			×
test-P-C1	×	×	×	×		×	×	×
test-OE-PE1								
test-OE-PR1								
test-M-M1								
test-S-A1				×	×	×		×
test-S-IN1					×	×		
test-S-P1					×			
test-C-SC1								

Table 15: **Module Traceability Part 2**

Appendix — Reflection

The information in this section will be used to evaluate the team members on the graduate attribute of Reflection. Please answer the following question:

1. In what ways was the Verification and Validation (VnV) Plan different from the activities that were actually conducted for VnV? If there were differences, what changes required the modification in the plan? Why did these changes occur? Would you be able to anticipate these changes in future projects? If there weren't any differences, how was your team able to clearly predict a feasible amount of effort and the right tasks needed to build the evidence that demonstrates the required quality? (It is expected that most teams will have had to deviate from their original VnV Plan.)

There were a lot of changes that we made to the original VnV plan that we made back in November. Some of the tests in the original VnV were a bit too implementation specific (e.g. expecting a certain function to exist for the system tests) and had to be changed in order to fit the actual implementation that we made. Additionally, we had to make changes to how we did usability testing and performance testing since in the original VnV plan we didn't know exactly what we were going to do. We think that in future projects these issues would be better be able to be anticipated with more experience in creating VnV plans. Specifically, focusing more on the specifics on usability testing (e.g. determining the target user for the usability surveys) would have improved the VnV process a lot, especially with usability being an important portion of this project.