Verification and Validation Report: Software Engineering

Team #13, ARC
Avanish Ahluwalia
Russell Davidson
Rafey Malik
Abdul Zulfiqar

March 10, 2025

1 Revision History

Date	Version	Notes
Date 1	1.0	Notes
Date 2	1.1	Notes

2 Symbols, Abbreviations and Acronyms

symbol	**description**
T	Test

[symbols, abbreviations or acronyms – you can reference the SRS tables if needed —SS]

Contents

1 Revision History **i**

2 Symbols, Abbreviations and Acronyms **ii**

3 Functional Requirements Evaluation **1**
3.1 Database Testing . . . 1
3.2 Custom AR Object Generation . . . 1
3.3 Uploading Objects to Inventory, Post Object Scan . . . 1
3.4 Realm Testing . . . 2
3.5 Object Placement Testing . . . 3
3.6 Interactions with User Inventory . . . 4

4 Nonfunctional Requirements Evaluation **6**
4.1 Usability Testing . . . 6
4.2 Security Testing . . . 6
4.3 Availability Testing . . . 6
4.4 Maintainability Testing . . . 6
4.5 Compliance Testing . . . 7
4.6 Reusability Testing . . . 7

5 Comparison to Existing Implementation **8**

6 Unit Testing **8**
6.1 Settings Module Testing . . . 8
6.2 Help Module Testing . . . 8
6.3 Collision Detection Module Testing . . . 8
6.4 Tour Proximity Module Testing . . . 9
6.5 Notifications Module Testing . . . 9
6.6 Access Hardware Testing . . . 9
6.7 Inventory Module Testing . . . 10
6.8 Object Placement Testing . . . 10
6.9 Restricted Area Detect Testing . . . 10
6.10 Weather Hazard Detect Testing . . . 11

7 Changes Due to Testing **11**

8 Automated Testing **12**

9 Trace to Requirements 12

10 Trace to Modules 12

11 Code Coverage Metrics 12

List of Tables

1 Functional Requirements Evaluation Results for Database Testing 1
2 Functional Requirements Evaluation Results for Custom AR Object Generation 1
3 Functional Requirements Evaluation Results for Uploading Objects to Inventory 2
4 Functional Requirements Evaluation Results for the Realm Interface 3
5 Functional Requirements Evaluation Results for Object Placement Features 4
6 Functional Requirements Evaluation Results for Inventory Features 5
7 Usability Testing Evaluation Results 6
8 Security Testing Evaluation Results 6
9 Availability Testing Evaluation Results 6
10 Maintainability Testing Evaluation Results 7
11 Compliance Testing Evaluation Results 7
12 Reusability Testing Evaluation Results 7
13 Settings Module Unit Test Results 8
14 Help Module Unit Test Results 8
15 Collision Detection Module Unit Test Results 9
16 Tour Proximity Module Unit Test Results 9
17 Access Hardware Module Unit Test Results 9
18 Inventory Module Unit Test Results 10
19 Restricted Area Detect Module Unit Test Results 10
20 Weather Hazard Detect Module Unit Test Results 11
21 Results of Usability Survey 14

List of Figures

1 ”Navigation between interfaces is intuitive” statement ratings 14

2 ”Placing objects is easy” statement ratings 15

3 ”Generating objects is easy” statement ratings 15

4 ”It is easy to start a tour” statement ratings 15

5 ”Changing settings is easy” statement ratings 16

6 ”The app is generally satisfying to use” statement ratings 16

7 ”Using the app distracts from the surroundings” statement ratings . . 16

This document ...

3 Functional Requirements Evaluation

3.1 Database Testing

The following section presents the results of the our database testing

Table 1: **Functional Requirements Evaluation Results for Database Testing**

Id	**Type**	**Inputs**	**Expected Result**	**Actual Result**	**Result**
Test-DB1	Automated	Periodic backup run is completed.	Automated monitor verifies that the database backup is present and correct.	Same as expected	Pass
Test-DB2	Automated	Command to check encryption status is inputted into DBMS for all databases	DBMS response shows that all databases are encrypted	Same as expected	Pass

3.2 Custom AR Object Generation

The following section presents the results of our custom AR object generation testing.

Table 2: **Functional Requirements Evaluation Results for Custom AR Object Generation**

Id	**Control**	**Inputs**	**Expected Result**	**Actual Result**	**Result**
Test-POG1	Automatic	Enter prompts of various lengths, with and without profanity.	Prompt is restricted to 200 characters, real-time character count is displayed, profanity is flagged and rejected.	Same as expected	Pass
Test-POG5	Manual	Rotate the AR object to inspect all sides.	The AR object rotates smoothly, allowing inspection from all angles.	Same as expected	Pass

3.3 Uploading Objects to Inventory, Post Object Scan

The following section presents the results of our testing for uploading objects to inventory after scanning.

Table 3: **Functional Requirements Evaluation Results for Uploading Objects to Inventory**

Id	**Control**	**Inputs**	**Expected Result**	**Actual Result**	**Result**
Test-OUI1	Manual	Display the scanned object and allow for user interaction in editing mode.	Render of scanned object is displayed, user can edit and finalize the object.	Same as expected	Pass
Test-OUI2	Manual	Provide a name for the object and save it with metadata.	Object name is stored (ASCII only), and all metadata is correctly saved.	Same as expected	Pass
Test-OUI3	Manual	Select specific portions of the object and apply color changes.	Color changes are applied accurately and reflected in the final render.	Same as expected	Pass

3.4 Realm Testing

The following section presents the results of our testing of the realm interface.

Table 4: **Functional Requirements Evaluation Results for the Realm Interface**

Id	Control	Inputs	Expected Result	Actual Result	Result
Test-RI1	Manual	Tester changes their position and angle in relation to an AR object.	The AR object adjusts perspective appropriately, reflecting the new camera position and angle.	Same as expected	Pass
Test-RI2	Manual	Tester moves camera over a crowded area where multiple AR objects are present.	The interface selectively displays a manageable number of AR objects without overwhelming the user's view.	Same as expected	Pass
Test-RI3	Manual	Test AR object instance is placed with a known alignment in the real world, and reference screenshots.	Test AR object appears in correct position and orientation as expected, matches stored object instance data.	Same as expected	Pass
Test-RI6	Manual	Tester attempts to access the object placement workflow via the provided control.	Tester is successfully redirected to the object placement workflow.	Same as expected	Pass
Test-RI8	Manual	Tester moves within range of the tour start point.	The interface displays a clear indication of the nearby tour and a link to the tour preview.	Same as expected	Pass
Test-RI9	Manual	Tester moves closer to a hazard in real space.	Interface displays a clear warning when the user approaches the hazard.	Same as expected	Pass

3.5 Object Placement Testing

The following section presents the results of our object placement testing.

Table 5: **Functional Requirements Evaluation Results for Object Placement Features**

Id	**Control**	**Inputs**	**Expected Result**	**Actual Result**	**Result**
Test-OP1	Manual	Tester selects object from inventory or prompt generation.	Interface successfully proceeds to the placement interface with the selected object.	Same as expected	Pass
Test-OP3	Manual	Tester rotates, resizes, and translates the object in real space.	Object is placed accurately in real space with correct orientation.	Same as expected	Pass
Test-OP4	Manual	Tester checks the AR object instance database.	Object instance is present with correct details (type, position, orientation).	Same as expected	Pass
Test-OP5	Automated and Manual	Tester attempts to place another object in an area with placement limit reached.	System prevents additional placements, displaying a warning.	Same as expected	Pass
Test-OP6	Automated and Manual	Tester attempts to place another object within a short period after the time-based limit is reached.	System restricts further placements, displaying a warning.	Same as expected	Pass
Test-OP7	Automated and Manual	Tester places an object, but the initial storage attempt fails.	System automatically retries storage until success or retry limit is reached.	Same as expected	Pass

3.6 Interactions with User Inventory

The following section presents the results of our testing of interactions with the user inventory.

Table 6: **Functional Requirements Evaluation Results for Inventory Features**

Id	Control	Inputs	Expected Result	Actual Result	Result
Test-IV1	Manual	Tester selects an object and chooses the delete option.	The selected object is removed from the inventory.	Same as expected	Pass
Test-IV2	Manual	Tester adds a new object to the inventory.	The new object appears in the inventory.	Same as expected	Pass
Test-IV3	Automatic	Tester opens the inventory.	Inventory contains the preloaded application-provided objects.	Same as expected	Pass
Test-IV4	Automatic	Tester attempts to add an additional object.	The object is successfully added, but adding another would be prevented.	Same as expected	Pass
Test-IV5	Manual	Tester opens the inventory and inspects object origins.	Each personal object is present.	Same as expected	Pass
Test-IV6	Automatic	Tester views the total count of objects.	The app displays the correct total number of objects.	Same as expected	Pass
Test-IV7	Manual	Tester adds both 2D and 3D AR objects to their inventory.	Both 2D and 3D objects are correctly stored in inventory.	Same as expected	Pass
Test-IV9	Manual	Tester sorts objects by usage or size.	Objects are sorted as per user selection.	Same as expected	Pass
Test-IV10	Automatic	Tester selects option to view a 3D AR object.	3D objects are displayed in a continuous rotating state.	Same as expected	Pass

4 Nonfunctional Requirements Evaluation

4.1 Usability Testing

The following section presents the results of our usability testing.

Table 7: **Usability Testing Evaluation Results**

Id	**Type**	**Inputs**	**Expected Result**	**Actual Result**	**Result**
Test-QS-U1	Manual	Language setting is changed to English, Mandarin, Hindi, Spanish, and French.	Text updates correctly in all tested languages with understandable translations.	Same as expected	Pass
Test-QS-U2	Manual	New users perform core app workflows without guidance.	80% of testers complete tasks and rate the app as intuitive and satisfying.	Same as expected	Pass

4.2 Security Testing

The following section presents the results of our security testing.

Table 8: **Security Testing Evaluation Results**

Id	**Type**	**Inputs**	**Expected Result**	**Actual Result**	**Result**
Test-QS-SC3	Manual	Code sections displaying private data are checked for identity verification.	All sections contain identity verification checks.	Same as expected	Pass

4.3 Availability Testing

The following section presents the results of our availability testing.

Table 9: **Availability Testing Evaluation Results**

Id	**Type**	**Inputs**	**Expected Result**	**Actual Result**	**Result**
Test-QS-A1	Automated	Monitor server uptime over one week.	Server uptime recorded at 99% or higher.	Same as expected	Pass

4.4 Maintainability Testing

The following section presents the results of our maintainability testing.

Table 10: **Maintainability Testing Evaluation Results**

Id	**Control**	**Inputs**	**Expected Result**	**Actual Result**	**Result**
Test-DI-M1	Manual and Automated	Simulate common errors like database connection failure, invalid input data, service timeout in internal APIs.	Error messages clearly indicate the source and nature of the error (90% of the cases).	Same as expected	Pass

4.5 Compliance Testing

The following section presents the results of our compliance testing.

Table 11: **Compliance Testing Evaluation Results**

Id	**Control**	**Inputs**	**Expected Result**	**Actual Result**	**Result**
Test-CO1	Manual	App is checked against the Personal Information and Electronic Documents Act (PIPEDA).	The app complies with all sections of PIPEDA.	Same as expected	Pass
Test-CO2	Manual	The app's revenue records are checked for purchases and ad-revenue spanning at least 6 years.	The records go back at least 6 years.	N/A	
Test-CO3	Manual	App is checked against the Google Play Developer Policy.	The app complies with all sections of the Google Play Developer Policy.	Same as expected	Pass
Test-CO4	Manual	App is checked against the App Store Review Guidelines.	The app complies with all sections of the App Store Review Guidelines.	Same as expected	Pass

4.6 Reusability Testing

The following section presents the results of our reusability testing.

Table 12: **Reusability Testing Evaluation Results**

Id	**Control**	**Inputs**	**Expected Result**	**Actual Result**	**Result**
Test-DI-R1	Static	All code is sent to a static analyzer that detects code duplication.	The analysis shows metrics related to code sections with a high amount of duplication, suggesting areas for refactoring.	Some duplicate code was found. Refactoring to fix this issue.	Fail

5 Comparison to Existing Implementation

This section is not applicable to this project.

6 Unit Testing

This section provides the test reports for the unit tests performed on various modules of the system.

6.1 Settings Module Testing

The following section presents the results of our Settings Module testing. The tests verify that the settings module correctly validates input keys and ensures profile details match the expected schema.

Table 13: **Settings Module Unit Test Results**

Id	**Type**	**Inputs**	**Expected Result**	**Actual Result**	**Result**
Test-SM1	Functional, Automated	Valid and invalid settings keys	Returns true for valid key, false for invalid key	Same as expected	Pass
Test-SM2	Functional, Automated	Valid user settings object	Returns object matching expected schema	Same as expected	Pass

6.2 Help Module Testing

The following section presents the results of our Help Module testing. The test verifies that the search functionality correctly returns relevant help items when given partial and full keywords.

Table 14: **Help Module Unit Test Results**

Id	**Type**	**Inputs**	**Expected Result**	**Actual Result**	**Result**
Test-HM1	Functional, Automated	Partial and full keywords matching help items	Outputs match expected search results	Same as expected	Pass

6.3 Collision Detection Module Testing

The following section presents the results of our Collision Detection Module testing. The test ensures that the module correctly identifies potential collisions based on AR tracking and accelerometer data.

Table 15: **Collision Detection Module Unit Test Results**

Id	Type	Inputs	Expected Result	Actual Result	Result
Test-CD1	Functional, Automated	Mock AR tracking and accelerometer data	Returns true for collisions, false otherwise	Same as expected	Pass

6.4 Tour Proximity Module Testing

The following section presents the results of our Tour Proximity Module testing. The test ensures that the module correctly detects and returns a list of nearby tours based on GPS data.

Table 16: **Tour Proximity Module Unit Test Results**

Id	Type	Inputs	Expected Result	Actual Result	Result
Test-TP1	Functional, Automated	Mock GPS data for device and tours	Outputs list of tours within specified distance	Same as expected	Pass

6.5 Notifications Module Testing

N/A

6.6 Access Hardware Testing

Table 17: **Access Hardware Module Unit Test Results**

Id	Type	Inputs	Expected Result	Actual Result	Result
Test-AHM1	Automated	Known simulator height value compared with simulator height.	The known height value should match the simulator's height.	Same as expected	Pass
Test-AHM2	Automated	Known simulator width value compared with simulator width.	The known width value should match the simulator's width.	Same as expected	Pass

6.7 Inventory Module Testing

Table 18: **Inventory Module Unit Test Results**

Id	Type	Inputs	Expected Result	Actual Result	Result
Test-IM1	Automated	Ensure the object count is less than or equal to the maximum allowed.	The object count should be less than or equal to the maximum object count (MAX_OBJ_COUNT).	TOTAL_OBJ_COUNT Same as expected	Pass
Test-IM2	Automated	Add an object to the inventory.	The TOTAL_OBJ_COUNT should increase by one, and the object should be added to the objects list.	Same as expected	Pass
Test-IM3	Automated	Retrieve an object from the inventory.	The object should be returned with its properties intact.	Same as expected	Pass
Test-IM4	Automated	Delete an object from the inventory.	The TOTAL_OBJ_COUNT should decrease by one, and the object should be removed from the objects list.	Same as expected	Pass
Test-IM5	Automated	Retrieve the list of all objects in the inventory.	The list should contain exactly the number of objects corresponding to the TOTAL_OBJ_COUNT.	Same as expected	Pass

6.8 Object Placement Testing

N/A

6.9 Restricted Area Detect Testing

Table 19: **Restricted Area Detect Module Unit Test Results**

Id	Type	Inputs	Expected Result	Actual Result	Result
Test-RADM1	Automated	GPS coordinates of a known restricted area.	The module should detect that the area is restricted.	Same as expected	Pass
Test-RADM2	Automated	GPS coordinates of a known unrestricted area.	The module should detect that the area is unrestricted.	Same as expected	Pass

6.10 Weather Hazard Detect Testing

Table 20: **Weather Hazard Detect Module Unit Test Results**

Id	Type	Inputs	Expected Result	Actual Result	Result
Test-WHDM1	Automated	Make an external call to the weather API for the Toronto area.	The weather data returned by the external API request should match the data returned by the module.	Same as expected	Pass

7 Changes Due to Testing

[This section should highlight how feedback from the users and from the supervisor (when one exists) shaped the final product. In particular the feedback from the Rev 0 demo to the supervisor (or to potential users) should be highlighted. —SS]

The following Tests for Functional Requirements (3.1 of VnVPlan) subsections where removed due to focus the project on tours instead of social media as advised in the Rev 0 demo:

- 3.1.1
- 3.1.2
  - Test-RI4
  - Test-RI5
  - Test-RI7
  - Test-RI10
- 3.1.3
  - Test-OP2
- 3.1.7
- 3.1.8
- 3.1.9
- 3.1.11
- 3.1.12
  - Test-IV8

8 Automated Testing

9 Trace to Requirements

10 Trace to Modules

11 Code Coverage Metrics

References

Appendix — Usability Survey Results

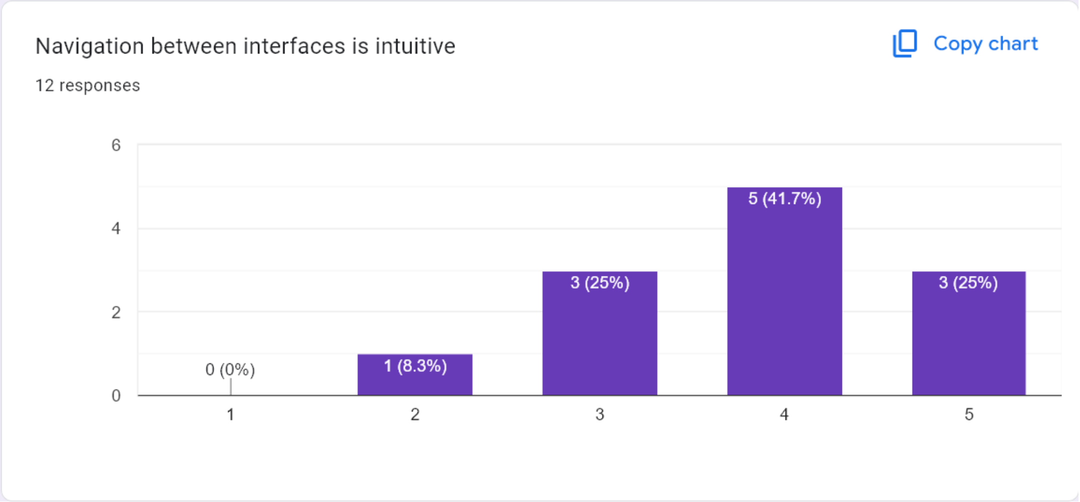
Link to view survey: here

Table 21 below showing the results of the Usability survey

Table 21: **Results of Usability Survey**

Statement	**Average Rating of Statement Accuracy / 5**	**Analysis**
Navigation between interfaces is intuitive	3.833	Most users found the navigation to be intuitive, although navigation seems to be the lowest rated aspect of the functional user experience
Placing objects is easy	3.917	No ratings below a three and an average rating of ”Agree” says that this was well recieved
Generating objects is easy	3.917	Again, no ratings below a three and an average rating of ”Agree” indicates that the design works for most users
It is easy to start a tour	4.167	A good indication that the touring experience was designed well
Changing settings is easy	4.417	Somewhat expected, users generally did not have issues finding and changing settings as it was a straightforward feature
The app is generally satisfying to use	3.667	This was the lowest rating of all our positive statements. We recieved relevant feedback on the non-uniform look and feel of the app making the app feel like a rushed development
Using the app distracts from the surroundings	3.167	More found the app distracting than not, but the results are somewhat inconclusive given the variance

Figure 1: ”Navigation between interfaces is intuitive” statement ratings


Navigation between interfaces is intuitive

12 responses

Figure 2: ”Placing objects is easy” statement ratings

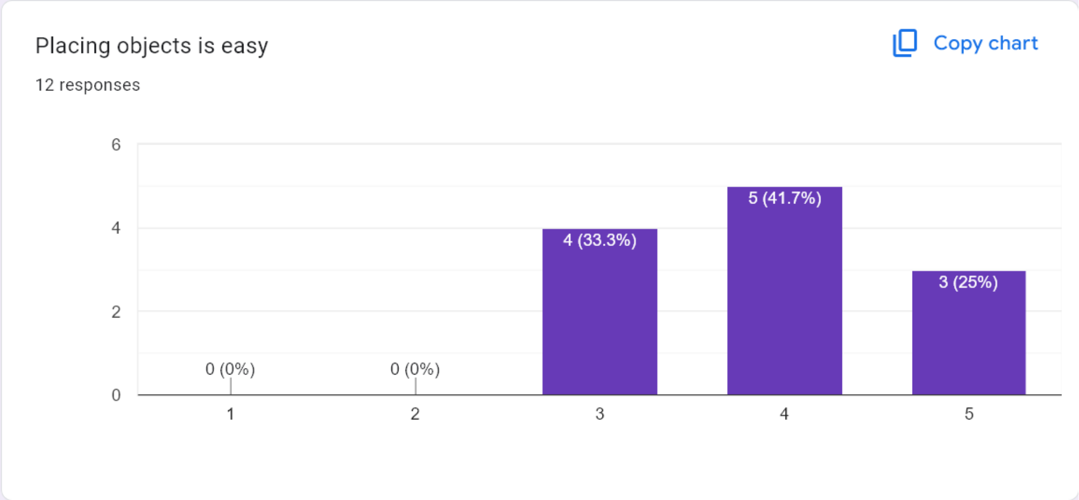


Figure 3: ”Generating objects is easy” statement ratings

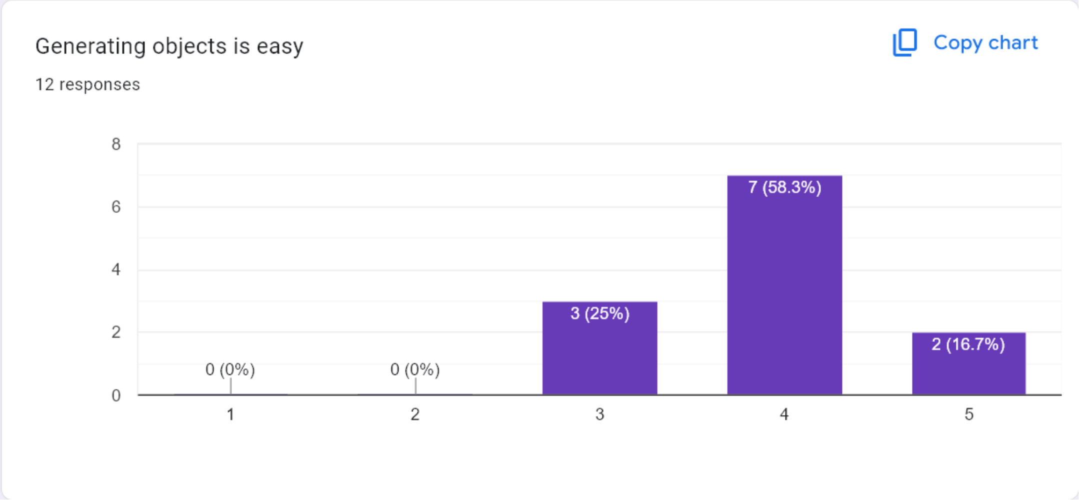


Figure 4: ”It is easy to start a tour” statement ratings

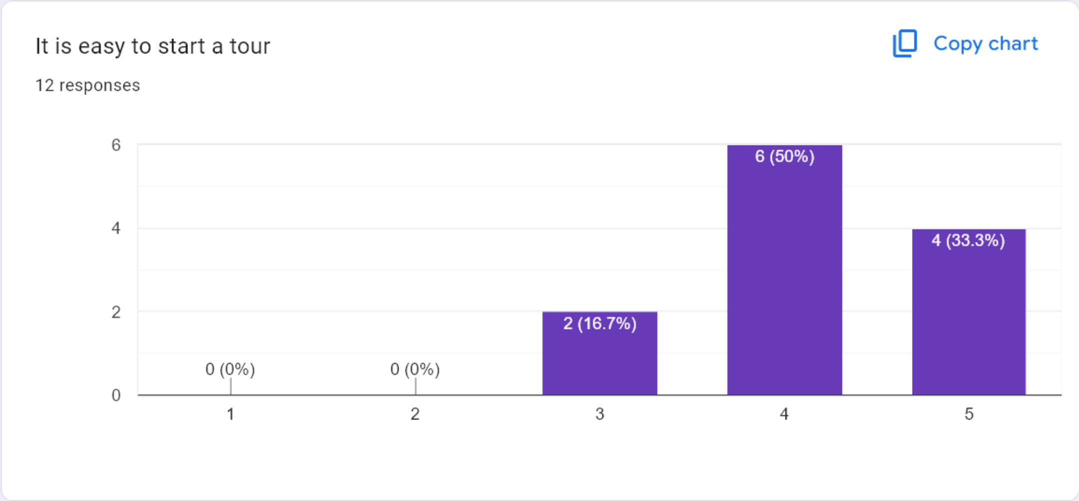


Figure 5: ”Changing settings is easy” statement ratings

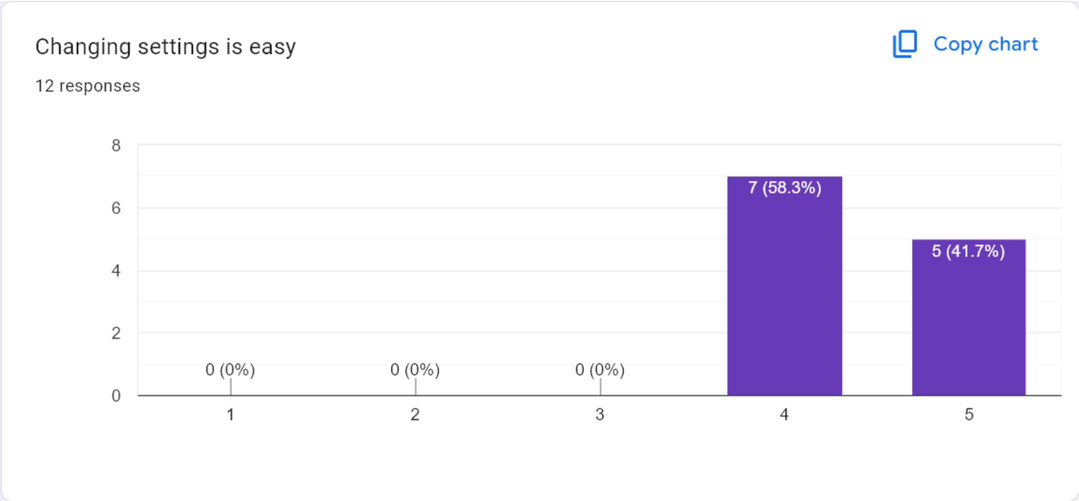


Figure 6: ”The app is generally satisfying to use” statement ratings

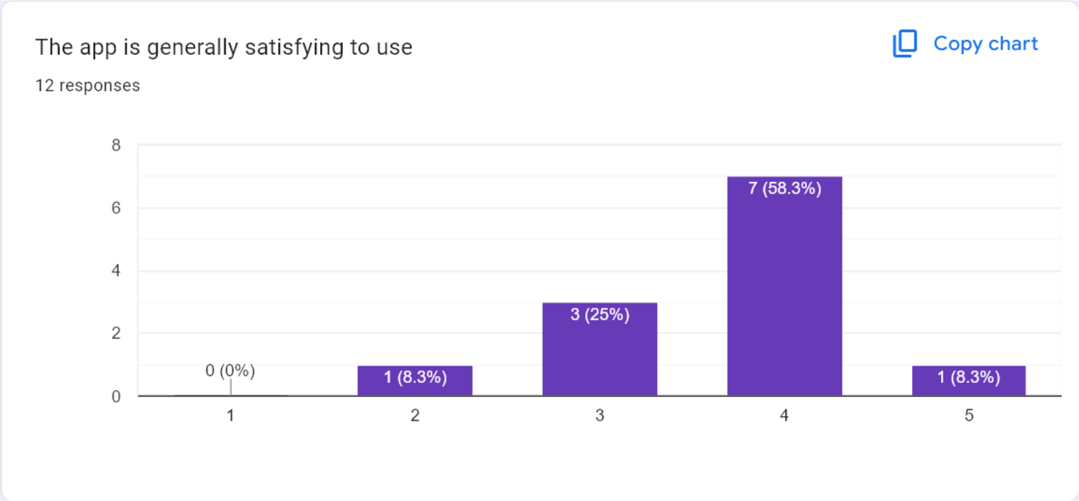
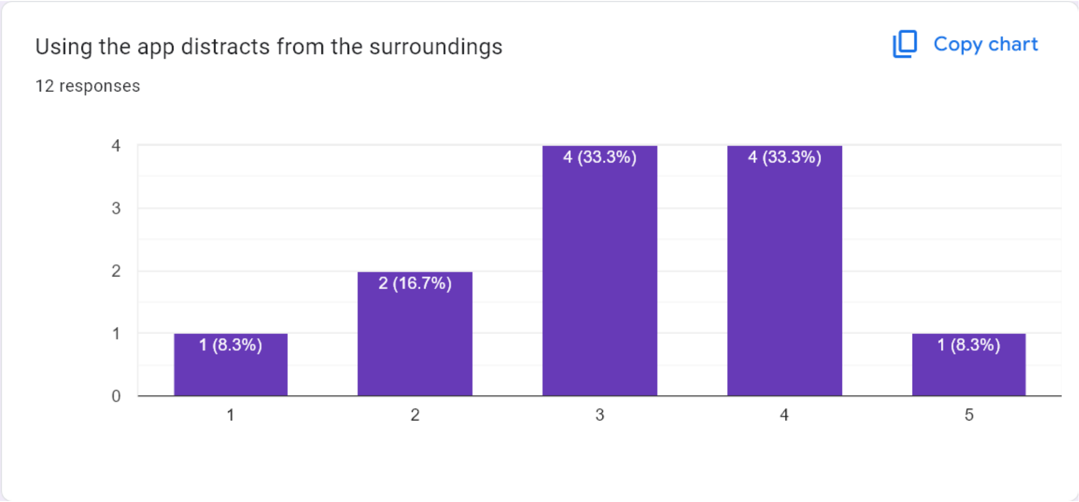


Figure 7: ”Using the app distracts from the surroundings” statement ratings



Appendix — Reflection

The information in this section will be used to evaluate the team members on the graduate attribute of Reflection.

The purpose of reflection questions is to give you a chance to assess your own learning and that of your group as a whole, and to find ways to improve in the future. Reflection is an important part of the learning process. Reflection is also an essential component of a successful software development process.

Reflections are most interesting and useful when they're honest, even if the stories they tell are imperfect. You will be marked based on your depth of thought and analysis, and not based on the content of the reflections themselves. Thus, for full marks we encourage you to answer openly and honestly and to avoid simply writing "what you think the evaluator wants to hear."

Please answer the following questions. Some questions can be answered on the team level, but where appropriate, each team member should write their own response:

1. What went well while writing this deliverable?

2. What pain points did you experience during this deliverable, and how did you resolve them?

3. Which parts of this document stemmed from speaking to your client(s) or a proxy (e.g. your peers)? Which ones were not, and why?

4. 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.)