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
2025-03-10	1.0	Initial report
2025-03-28	1.1	Modifications made based on review

2 Symbols, Abbreviations and Acronyms

See SRS Documentation here for a full list.

symbol	description
T	Test

Contents

List of Tables

List of Figures

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

3.7 Profile Testing

The following section presents the results of the profile testing

Table 7: **Functional Requirements Evaluation Results for Profile Testing**

Id	**Type**	**Inputs**	**Expected Result**	**Actual Result**	**Result**
Test-PS1	Manual	User enters valid credentials.	The user successfully logs in and is redirected to their profile page.	Same as expected	Pass
Test-PS2	Manual	User inputs new password and confirms.	System updates the password and provides a confirmation message.	Same as expected	Pass
Test-PS3	Automated	None	Profile information (username, password, status) is displayed correctly.	Same as expected	Pass
Test-PS4	Manual	User navigates to help page.	A help page with FAQs and ad- ditional help information is displayed.	Same as expected	Pass

3.8 Touring

The following section presents the results of the general users experience of using a tour.

Table 8: **Functional Requirements Evaluation Results for Touring**

Id	**Type**	**Inputs**	**Expected Result**	**Actual Result**	**Result**
Test-TR1	Manual	General user attempts to navigate to the touring screen.	The touring screen is reachable.	Same as expected	Pass
Test-TR2	Manual	Organization user attempts to navigate to touring screen.	The touring screen is hidden from user.	Same as expected	Pass
Test-TR3	Manual	General user finds a tour and attempts to preview it	User can see the information described in TR-FR3.	Same as expected	Pass
Test-TR4	Manual	General user navigates to the tour list interface, and searches for a tour belonging to an organization.	The tour has been found.	Same as expected	Pass
Test-TR5	Manual	General user goes close to a tour area in the real-world	A push notification appears on the user's phone indicating that a tour is nearby and prompts them to preview it.	Same as expected	Pass
Test-TR6	Manual	General user scans the QR code through the camera app.	The camera app opens Realm to the preview of the corre- sponding tour.	Same as expected	Pass
Test-TR8	Manual	General user selects map view	The user can see the map with the properties described in TR-FR4.1.	Same as expected	Pass

3.9 Tour management

The following section presents the results of the organizational users side of managing a tour.

Table 9: **Functional Requirements Evaluation Results for Managing Tours**

Id	Control	Inputs	Expected Result	Actual Result	Result
Test-TM1	Manual	Organization user attempts to navigate to tour management screen.	The tour management screen is reachable.	Same as expected	Pass
Test-TM2	Manual	General user attempts to navigate to tour management screen.	The tour management screen is hidden from user.	Same as expected	Pass
Test-TM3	Manual	User attempts to create a tour by inputting all the information described in TM-FR4 and placing one of each type of object in the environment.	The tour is successfully created with the correct data.	Same as expected	Pass
Test-TM4	Manual	User attempts to create a tour by inputting all the information described in TM-FR4 and selects the option to save as a draft.	The tour is successfully created as a draft.	Tours aren't being saved as a draft	Fail
Test-TM5	Manual	Organization user attempts to create a tour by inputting all the information described in TM-FR4 and selects the option to publish the tour.	The tour is successfully created and published.	Same as expected	Pass
Test-TM6	Manual	User navigates to the draft tour and selects publish option.	The tour is successfully published.	Draft not published since tours aren't being saved as a draft.	Fail
Test-TM7	Manual	User navigates to the tour and selects the preview option.	The tour can be previewed through the lens of what a Gen- eral User would see.	Same as expected	Pass
Test-TM8	Manual	User navigates to the tour they wish to edit, selects the edit option and changes all the inputs described in TM-FR4.	The tour is successfully edited with the correct data.	Same as expected	Pass

3.10 Maps Interface Testing

The following table presents the results of the our maps interface testing:

Table 10: **Functional Requirements Evaluation Results for Maps Interface Testing**

Id	**Type**	**Inputs**	**Expected Result**	**Actual Result**	**Result**
Test-MP1	Manual	Tester is present on the Map interface and is viewing the complete map and check displayed information.	1. User's current location is displayed on the map. 2. Location markers appear for AR object clusters. 3. Markers show the count of objects in each cluster. 4. All sub-realm objects associated with the user are indicated on the map.	Same as expected	Pass
Test-MP2	Manual	Select a marker and initiate navigation. Terminate navigation mid-route.	1. Directions to the selected marker are provided. 2. Navigation terminates when requested by the user.	Same as expected	Pass
Test-MP3	Automated	Zoom in and out on the map and navigate toward restricted areas.	1. Objects are grouped to reduce clutter on the map. 2. Restricted areas are identified, and navigation to these areas is disallowed.	Same as expected	Pass

3.11 Settings Testing

The following table presents the results of the our settings testing:

Table 11: **Functional Requirements Evaluation Results for Settings Testing**

Id	**Type**	**Inputs**	**Expected Result**	**Actual Result**	**Result**
Test-S1	Manual	User adjusts text size, enables/disables viewing of object names, or changes language.	The expected result is that accessibility settings apply as configured by the user.	Same as expected	Pass
Test-S2	Manual	User changes display settings such as light/dark mode or AR object visibility.	The expected result is that display settings reflect user preferences.	Same as expected	Pass
Test-S3	Manual	User modifies privacy settings to control visibility of profile, friends list, and AR interactions.	The expected result is that privacy settings update based on user preferences.	Same as expected	Pass
Test-S4	Manual	User changes username, password, profile picture, or status.	The expected result is that profile settings are updated and saved.	Same as expected	Pass
Test-S5	Manual	User navigates to sub-realm settings to modify sub-realm options.	The expected result is that sub-realm settings are accessible and configurable.	Same as expected	Pass

4 Nonfunctional Requirements Evaluation

4.1 Usability Testing

The following section presents the results of our usability testing.

Table 12: **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 13: **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 14: **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 15: **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 16: **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 17: **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

4.7 Portability Testing

The following table presents the results of the portability testing:

Table 18: **Portability Testing Evaluation Results**

Id	Type	Inputs	Expected Result	Actual Result	Result
Test-PT1	Non-Functional, Manual	Run the app on iOS and Android devices.	The app is functional and displays correctly on both platforms.	App works correctly on both devices	Pass
Test-PT2	Non-Functional, Code Review	Inspect the codebase to ensure shared files are correctly configured with minimal platform-specific files.	Codebase only differs in configuration files for platform-specific settings.	Same as expected	Pass
Test-PT3	Non-Functional, Automated	Initiate automated builds for both iOS and Android.	Both builds succeed without errors.	Same as expected	Pass

4.8 Safety Testing

The following table presents the results of the safety testing:

Table 19: **Safety Testing Evaluation Results**

Id	Type	Inputs	Expected Result	Actual Result	Result
Test-SA1	Non-Functional, Survey-Based	Conduct a user survey after users engage with the app in a controlled environment.	Survey results show that users do not find the app dangerously distracting them from their surroundings while using it.	Based on the survey results shown in the table below, users find the app somewhat distracting.	Fail
Test-SA2	Non-Functional, Manual Inspection	Navigate through all screens and interactions within the app.	No bright flashes or loud noises are present in any of the app interfaces.	No bright flashes were noticed. Loud Noises are not heard on any user interface.	Pass

4.9 Installation Testing

The following table presents the results of the installation testing:

Table 20: **Installation Testing Evaluation Results**

Id	Type	Inputs	Expected Result	Actual Result	Result
Test-I1	Non-Functional, Manual	Search for the app on the Apple App Store and Google Play Store.	The app is available for download on both app stores.	Same as expected	Fail
Test-I2	Non-Functional, Manual	Attempt to install the app on a device from both the Apple App Store and Google Play Store.	The app installs directly without any additional steps or configurations.	Same as expected	Fail

4.10 Performance Testing

The following section presents the results of our performance testing.

Table 21: **Performance Testing Evaluation Results**

Id	Type	Inputs	Expected Result	Actual Result	Result
Test-QS-PE1	Automatic	User performs actions to navigate to the Map interface.	The map and its overlays are completely visible and can be interacted with.	Same as expected	Pass
Test-QS-PE2	Automatic	User selects the option to view their entire Inventory.	The Inventory loads completely and can be interacted with within 1-10 seconds depending on the number of objects present in the Inventory.	Same as expected	Pass
Test-QS-PE4	Manual	Tester initiates the generation of an AR object.	The AR object is fully generated and visible.	Same as expected	Pass
Test-QS-PE5	Manual	Tester attempts to view AR objects within the app.	The app renders AR objects with minimal lag or provides a fallback mode (low-resolution objects) for accessibility.	Same as expected	Pass

4.11 Reliability Testing

The following section presents the results of our reliability testing.

Table 22: **Reliability Testing Evaluation Results**

Id	Type	Inputs	Expected Result	Actual Result	Result
Test-QS-RE1	Automatic	Inject random data or errors into the test database to trigger failure.	(a) Database recovers automatically or through a set of recovery steps. (b) If there is user data loss, only 2% of user data will be lost after the recovery. (c) System returns to normal operation, allowing all users to access their data without issues.	Data recovers, but more than 2% of user data is being lost.	Fail

4.12 Distribution Testing

The following section presents the results of our distribution testing.

Table 23: **Distribution Testing Evaluation Results**

Id	Type	Inputs	Expected Result	Actual Result	Result
Test-DI-D1	Automated	Attempt to download and install the app on the user device.	(a) The app installs successfully on devices running iOS 16.0+ and Android 12+. (b) The app functions as expected post-installation on the device.	Same as expected	Fail
Test-DI-D2	Manual	Review server locations where user data is stored.	All user data is stored within North America.	Same as expected	Pass

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 24: **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 25: **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 26: **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 27: **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 28: **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 29: **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 30: **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 31: **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

6.11 Authentication Module

Table 32: **Unit Testing Evaluation Results for Authentication Module**

Id	Type	Inputs	Expected Result	Actual Result	Result
Test-AM-SN	Manual	User ID, notification message	Bool - Success or Fail	Success	Pass
Test-AM-FN	Manual	User ID	List of notifications	Same as expected	Pass

6.12 Object Render Module

Table 33: **Unit Testing Evaluation Results for Object Render Module**

Id	Type	Inputs	Expected Result	Actual Result	Result
Test-AM-FRS	Manual	-, Dictionary of current render settings	Bool - Success or Fail	Success	Pass

6.13 Touring Module

Table 34: **Unit Testing Evaluation Results for Touring Module**

Id	Type	Inputs	Expected Result	Actual Result	Result
Test-TM-ST	Manual	Tour ID, Dictionary of current render settings	Bool - Success or Fail	Success	Pass
Test-TM-ET	Manual	-, Dictionary of current render settings	Bool - Success or Fail	Success	Pass

7 Changes Due to Testing

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

Our automated test suite is divided into unit and integration tests. Unit tests focus on testing individual modules such as the Settings Module, Notifications Module, or Tour Management Module, in complete isolation. These unit tests rely on mocks and stubs to simulate external dependencies, allowing us to verify that each module behaves as expected under a variety of conditions.

The second part of our suite is integration tests, which ensure that modules work together correctly. These tests use the real implementations of our modules and often involve end-to-end scenarios (e.g., creating a tour, fetching its details, and then updating it). This helps us catch issues that might arise only when modules communicate with each other.

We run these tests automatically on our continuous integration (CI) pipeline. This setup makes it easy to catch regressions early and confirm that recent changes have not broken existing functionality. We also collect coverage metrics—both line coverage and branch coverage—for each module to track our progress and identify areas that may need more thorough testing.

9 Trace to Requirements

The following table shows the traceability matrix for the functional requirements:

Test-ID	**Test Name**	**Requirements**
Test-RI1	Validate AR Object Perspective Adjustment	RI-FR1.1
Test-RI2	Validate AR Object Clutter Management	RI-FR1.2
Test-RI3	Validate AR Object Placement Accuracy	RI-FR1.2
Test-RI6	Validate Object Placement Workflow Control	RI-FR3
Test-RI8	Validate Nearby Tour Indication	RI-FR5
Test-RI9	Validate Hazard Warning	RI-FR6
Test-RI10	Validate Offline Mode for Interactive Components	RI-FR7
Test-OP1	Validate Object Selection Stage	OP-FR2.1
Test-OP3	Validate Object Placement Stage	OP-FR2.3
Test-OP4	Validate Object Instance Storage	OP-FR1
Test-OP5	Validate Area Based Placement Limit	OP-FR3.1
Test-OP6	Validate Time Based Placement Limit	OP-FR3.2
Test-OP7	Validate Automated Retry for Failed Object Storage	OP-FR1
Test-DB1	Validate Periodic Database Backup	DB-FR1
Test-DB2	Validate Database Encryption	DB-FR2
Test-QS-U1	Validate Localization	QS-U1
Test-QS-U2	Validate User Intuitiveness and Satisfaction	QS-U2
Test-QS-A1	Automated Server Availability Monitoring	QS-A1
Test-QS-A2	User Feedback on Server Availability	QS-A1
Test-DI-M1	Validate API Error Message Clarity	DI-M1

Table 35: Mapping of Tests to Requirements (I)

Test-ID	**Test Name**	**Requirements**
Test-PS1	Validate User Authentication	PS-FR1
Test-PS2	Password Change Functionality	PS-FR3
Test-PS3	View Profile Information	PS-FR4
Test-PS4	Access Help Page	PS-FR6
Test-S1	Modify Accessibility Settings	S-FR1
Test-S2	Adjust Display Settings	S-FR2
Test-S3	Manage Privacy Settings	S-FR3
Test-S4	Update Profile Settings	S-FR4
Test-PT1	Validate Cross-Platform Compatibility	DI-P1
Test-PT2	Common Codebase Validation	DI-P2
Test-PT3	Build Verification on iOS and Android	DI-P1
Test-SA1	Distraction Level Assessment	QS-SA1
Test-SA2	No Bright Flashes or Loud Noises	QS-SA2
Test-I1	Verify App Store Availability	DI-I1
Test-I2	Simple Installation Process	DI-I2

Table 36: Mapping of Tests to Requirements (II)

Test-ID	**Test Name**	**Requirements**
Test-TU1	Tutorial opens on account creation	TU-FR2
Test-TU2	Tutorial can be opened through the settings screen	TU-FR4
Test-TU3	Tutorial involves all major features	TU-FR1
Test-TU4	User can exit the tutorial at any time	TU-FR3
Test-TU5	The tutorial is interactive in a sandbox environment	TU-FR5
Test-TM1	*Organization Users* can access tour management screen	TM-FR1
Test-TM2	*General Users* can NOT access the tour management screen	TM-FR1
Test-TM3	*Organization Users* can create a customized tour	TM-FR4
Test-TM4	*Organization Users* can create a tour as a draft	TM-FR2
Test-TM5	*Organization Users* can create a tour and directly publish it	TM-FR3
Test-TM6	*Organization Users* can publish a draft tour	TM-FR3
Test-TM7	*Organization Users* can preview one of their tours	TM-FR5
Test-TM8	*Organization Users* can edit one of their tours	TM-FR6
Test-TR1	*General Users* can access the touring screen	TR-FR1
Test-TR2	*Organization Users* can NOT access the touring screen	TR-FR1
Test-TR3	*General Users* can preview a tour	TR-FR3
Test-TR4	*General Users* can find a tour through the tour list interface	TR-FR2.1
Test-TR5	*General Users* can find a tour through a push notification when in proximity to a tour area in the real-world	TR-FR2.2

Table 37: Mapping of Tests to Requirements (III)

Test-ID	**Test Name**	**Requirements**
Test-TR6	*General Users* can find a tour through a QR code	TR-FR2.3
Test-TR7	*General Users* can switch between the map and AR view in a tour	TR-FR4
Test-TR8	*General Users* can see the map tour view	TR-FR4.1
Test-TR9	*General Users* can see the AR tour view	TR-FR4.2
Test-QS-SC1	Encryption implementation message reading	QS-SC1
Test-QS-SC2	Encryption implementation algorithm check	QS-SC1
Test-QS-SC3	Verify identity before transmitting private data	QS-SC2
Test-CO1	Check Personal Information and Electronic Documents Act (PIPEDA) compliance	CO1
Test-CO2	Tax records check going back six years	CO2
Test-CO3	Check *Google Play* developer policy compliance	CO3
Test-CO4	Check *App Store* review guidelines compliance	CO4
Test-DI-R1	Reusable components check	DI-R1
Test-IV1	Delete Object from Inventory	IV-FR1
Test-IV2	Add Object to Inventory	IV-FR2
Test-IV3	Application-Provided AR Objects in Inventory	IV-FR3
Test-IV4	Inventory Capacity for *Organization Users* Objects	IV-FR4
Test-IV5	Personal Object Source Verification	IV-FR5

Table 38: Mapping of Tests to Requirements (IV)

Test-ID	**Test Name**	**Requirements**
Test-IV6	Total Object Count in Inventory	IV-FR6
Test-IV7	Storage of 2D and 3D AR Objects	IV-FR7
Test-IV8	Add to Favourite Group	IV-FR8
Test-IV9	Sort Objects	IV-FR9
Test-IV10	Continuous Rotation for 3D Objects	IV-FR10
Test-MP1	Map Location and Display of Overlays	MP-FR1, MP-FR2, MP-FR3, MP-FR4, MP-FR5
Test-MP2	Navigation and Directions on Map	MP-FR6, MP-FR7, MP-FR9
Test-MP3	Clutter Management and Restricted Area Identification	MP-FR8, MP-FR10
Test-POG1	Prompt Entry and Validation	POG-FR1, POG-FR2, POG-FR3, POG-FR4
Test-POG2	Object Type Selection and Confirmation	POG-FR5, POG-FR6
Test-POG3	AR Object Generation and Selection	POG-FR7, POG-FR8
Test-POG4	Add to Inventory	POG-FR9
Test-POG5	Generated AR Object Preview	POG-FR10
Test-OUI2	Object Naming and Storage	OUI-FR5, OUI-FR6, OUI-FR7, OUI-FR8, OUI-FR9, OUI-FR10, OUI-FR11

Table 39: Mapping of Tests to Requirements (V)

Test-ID	**Test Name**	**Requirements**
Test-QS-PE1	Map Rendering	QS-PE1
Test-QS-PE2	Inventory Load	QS-PE2
Test-QS-PE3	Real-Time Render Delay	QS-PE3
Test-QS-PE4	AR Object Generation	QS-PE4
Test-QS-PE5	AR Object Fallback Mode	QS-PE5
Test-QS-RE1	Database Failure/Corruption	QS-RE1
Test-DI-D1	Device Compatibility	DI-D1
Test-DI-D2	Regional Availability	DI-D2
Test-DI-D3	Recommended Age Requirement Display	DI-D3
Test-DI-D4	User Data Storage in North America	DI-D4

Table 40: Mapping of Tests to Requirements (VI)

10 Trace to Modules

The following table shows the traceability matrix for unit tests to modules:

Test ID	**Test Name**	**Module**
Test-AM-SN	SendNotification	Authentication Module
Test-AM-FN	FetchNotifications	Authentication Module
Test-AM-MNR	MarkNotificationRead	Authentication Module
Test-AM-DN	DeleteNotification	Authentication Module
Test-AM-UNS	UpdateNotificationSettings	Authentication Module
Test-AM-FRS	FetchRenderSettings	Object Render Module
Test-TM-ST	StartTour	Touring Module
Test-TM-PT	PauseTour	Touring Module
Test-TM-ET	EndTour	Touring Module
Test-TM-FTD	FetchTourDetails	Touring Module
Test-TM-FTL	FetchTourList	Tour List Module
Test-TM-STQ	SearchTours	Tour List Module
Test-TM-CT	CreateTour	Tour Management Module
Test-TM-UT	UpdateTour	Tour Management Module
Test-TM-DT	DeleteTour	Tour Management Module

Table 41: Test Mapping to Modules (I)

Test ID	**Test Name**	**Module**
Test-LDM-FD	fetchData	Local Database Manager
Test-LDM-SD	saveData	Local Database Manager
Test-LDM-UD	updateData	Local Database Manager
Test-LDM-DD	deleteData	Local Database Manager
Test-LDM-SWS	syncWithServer	Local Database Manager
Test-LDM-ICS	isCacheStale	Local Database Manager
Test-SDM-FD	fetchData	Server Database Manager
Test-SDM-SD	saveData	Server Database Manager
Test-SDM-UD	updateData	Server Database Manager
Test-SDM-DD	deleteData	Server Database Manager
Test-SDM-SWL	syncWithLocal	Server Database Manager
Test-SDM-LSO	logSyncOperation	Server Database Manager
Test-RAC-SR	sendRequest	REST API Communication
Test-RAC-PR	parseResponse	REST API Communication
Test-RAC-SH	setHeaders	REST API Communication

Table 42: Test Mapping to Modules (II)

Test ID	**Test Name**	**Module**
Test-RAC-HA	handleAuthentication	REST API Communication
Test-RAC-CSS	checkServerStatus	REST API Communication
Test-RAC-BU	buildURL	REST API Communication
Test-RAC-LR	logRequest	REST API Communication
Test-RI-RO	renderObjects	Realm Interface
Test-SM1	Key Validation	Settings Module
Test-SM2	Ensure Valid Profile Details	Settings Module
Test-HM1	String Search	Help Module
Test-CD1	Detect Collision	Collision Detection Module
Test-TP1	Detect Nearby Tour	Tour Proximity Module

Table 43: Test Mapping to Modules (III)

11 Code Coverage Metrics

Module	Branch Coverage	Line Coverage
Settings Module	38%	55%
Help Module	32%	45%
Collision Hazard Module	36%	58%
Tour Proximity Module	49%	63%
Notifications Module	21%	38%
Authentication Module	28%	46%
Object Render Module	36%	50%
Touring Module	42%	61%
Tour List Module	34%	47%
Tour Management	37%	52%
Local Database Manager	39%	56%
Data Sync Module	31%	49%
Server Database Manager	46%	60%
REST API Communication Module	25%	44%
Realm Interface Module	50%	65%
AccessHardware Module	33%	48%
Inventory Module	43%	62%
Object Placement Module	27%	41%
Restricted Area Detection Module	18%	34%
Weather Hazard Detection Module	29%	45%

Table 44: Coverage Metrics by Module

The lower percentages are mostly because we focused more on testing the main features and haven't fully covered some less common cases yet. Also, some modules interact with other parts of the system, which made them harder to fully test on their own.

References

Appendix — Usability Survey Results

Link to view survey: here

Table 45 below showing the results of the Usability survey

Table 45: **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

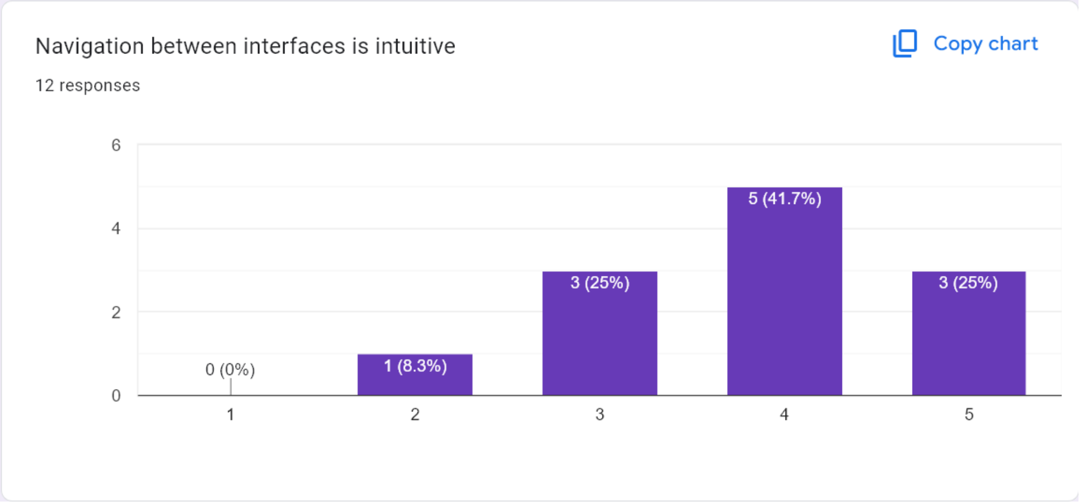


Figure 2: ”Placing objects is easy” statement ratings

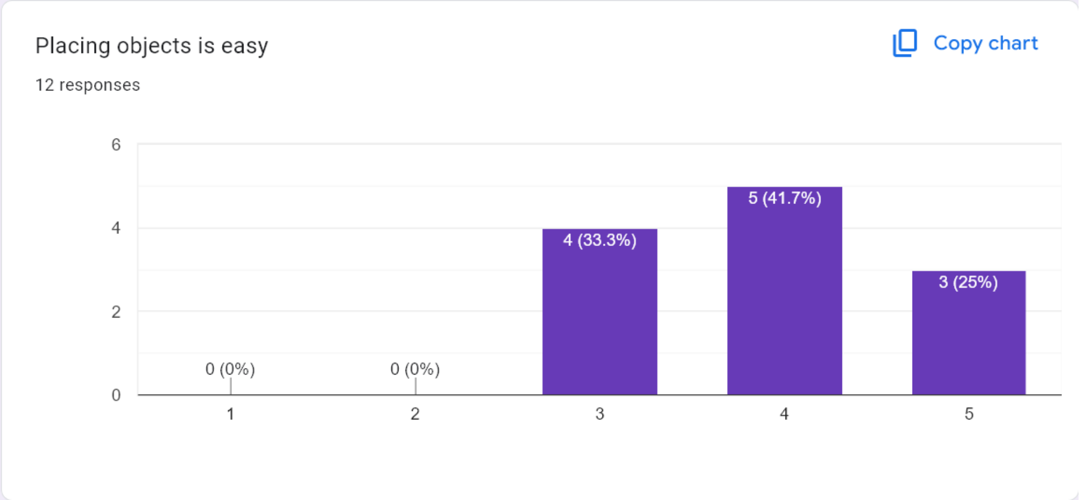


Figure 3: ”Generating objects is easy” statement ratings

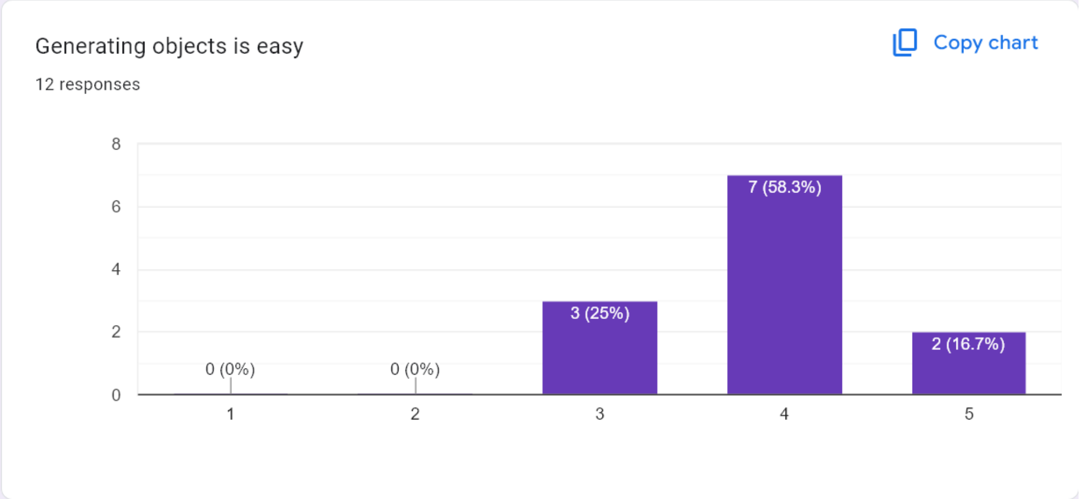


Figure 4: ”It is easy to start a tour” statement ratings

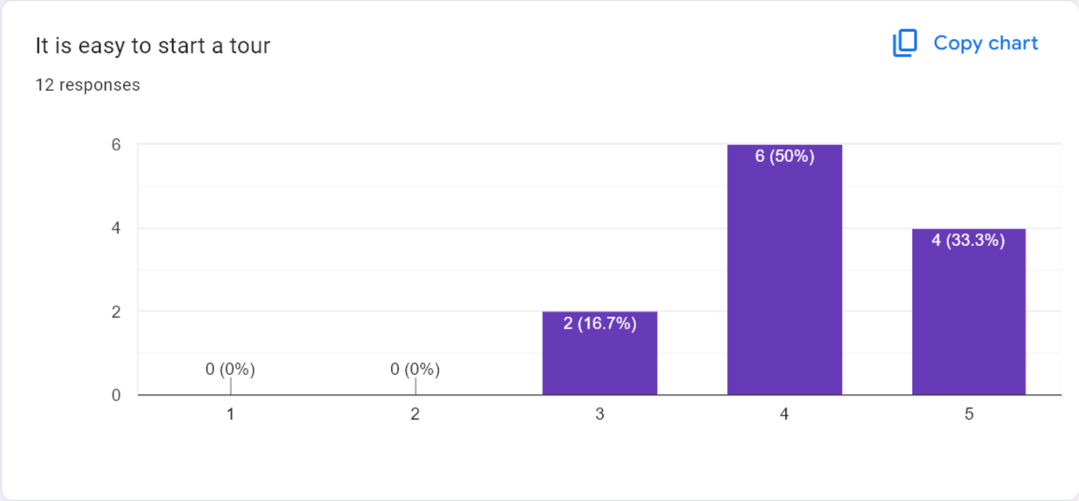


Figure 5: ”Changing settings is easy” statement ratings

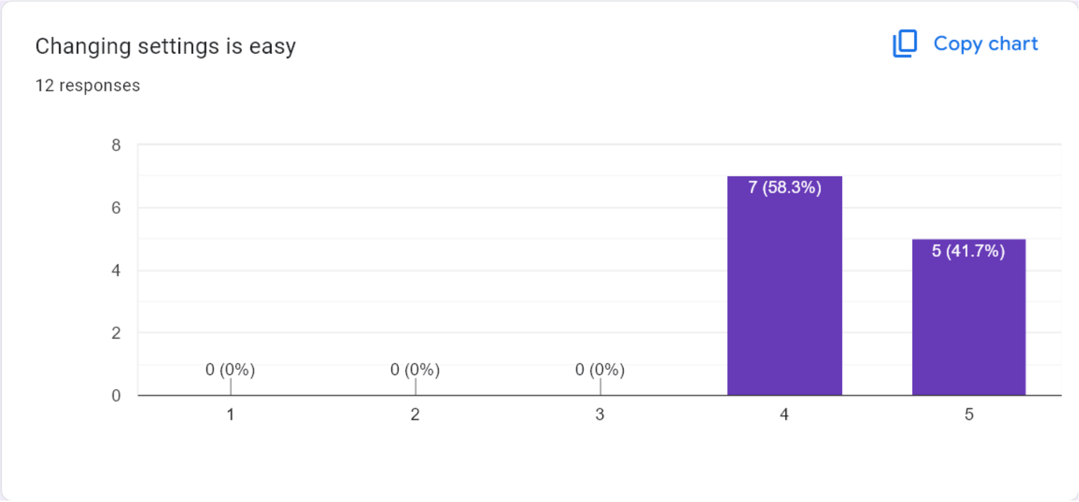


Figure 6: ”The app is generally satisfying to use” statement ratings

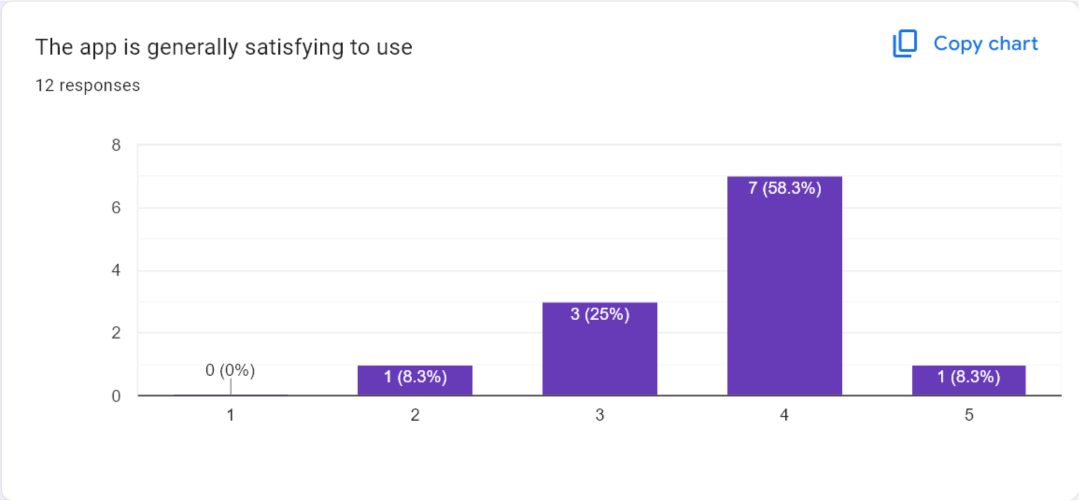
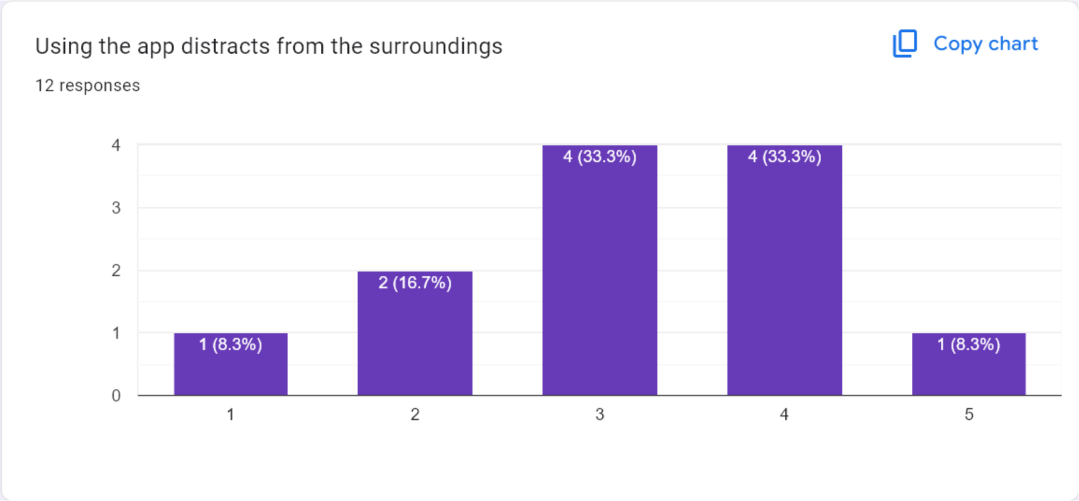


Figure 7: ”Using the app distracts from the surroundings” statement ratings



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?

   The unit tests seemed fairly simple and intuitive to do and the work was split up well between the group.

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

   Executing some of the test cases smoothly was a pain point, as well as getting them to pass, but sticking with it and sitting through them after some time, we were able to do our tests and pass 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?

   Usability survey and feedback on the app was given from peers. Since we don't have a client, a lot of our feedback was from the Professor and TA during Rev0.

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.)

   We had initially expected many of our tests to be automated, but after actually going through them, a lot of them seemed to be more manual work, such as logging in ourselves, testing out the tours, etc. We learned that many of the manual tests are moreso for the code correctness and such, and, at least for our project, the functionality had to be tested through doing.