

Software Requirements Specification: Realm Software Engineering

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

Contents

Revision History	3
1 Introduction	3
1.1 Document Purpose	3
1.2 Product Scope	3
1.3 Definitions, Acronyms and Abbreviations	3
1.4 References	5
1.5 Document Overview	5
2 Product Overview	5
2.1 Product Perspective	5
2.2 Product Functions	5
2.3 Product Constraints	5
2.4 User Characteristics	5
2.5 Assumptions and Dependencies	5
2.6 Apportioning of Requirements	5
3 Requirements	5
3.1 External Interfaces	5
3.1.1 User interfaces	5
3.1.2 Look and Feel Requirements	6
3.1.3 Hardware interfaces	6
3.1.4 Software interfaces	7
3.2 Functional	7
3.2.1 Tutorial	7
3.2.2 Tour Management	8
3.2.3 Touring	9
3.2.4 Object Placement	10
3.2.5 Realm Interface	10
3.2.6 Admin Interface	11
3.3 Use Cases	11
3.4 Quality of Service	23
3.4.1 Performance	23
3.4.2 Security	23
3.4.3 Reliability	23
3.4.4 Availability	23
3.5 Compliance	23
3.6 Design and Implementation	24
3.6.1 Installation	24
3.6.2 Distribution	24
3.6.3 Maintainability	25
3.6.4 Reusability	25
3.6.5 Portability	25
3.6.6 Cost	27

3.6.7	Deadline	27
3.6.8	Proof of Concept	27
4	Verification	27
5	Appendixes	27
5.1	Activity Diagrams	27

Revision History

Table 1: Revision History

Date	Developer(s)	Change
2024-10-07	Russell Davidson	Compliance, Installation, Distribution, and Portability
2024-10-07	Russell Davidson	Tutorial, Tour Management, and Touring
2024-10-08	Russell Davidson	UC1, UC2, UC3, and UC4
2024-10-10	Russell Davidson	UC11, UC12, and Use Cases diagram
2024-10-11	Russell Davidson	UC4 sequence diagram

1 Introduction

1.1 Document Purpose

1.2 Product Scope

1.3 Definitions, Acronyms and Abbreviations

- *AR object*: A 2D/3D projection of an entity.
- *Users*: A term for anyone who uses the app.
- *Organization users*: Users who belong to an organization that has the ability to create tours. They are affiliated with a particular approved organization who have the ability to modify tours within their organization's domain.
- *General users*: Users that can access most of the app functionalities except creating tours.
- *Admins*: People who have access to nothing else but the admin interface within the app.

1.3.1 Screen Names

- **Realm screen:** the screen in the app that provides a view of AR objects imposed on a feed of the user's camera.
- **Tour List screen:** Shows a list of the available tours and an option to start them.
- **Tour Editor screen:** Allows organization users to edit tours and publish changes.
- **Maps screen:** Map view displaying objects placed around you
- **Inventory:** A display of available AR objects (personal objects, objects mapped to a group and application preset objects)
 - **Groups screen:** Ability to add, edit and interact with group settings
 - **Friends screen:** Managing friends
 - **Settings:** Accessibility settings, Display settings, Privacy settings, Profile Settings, Group settings
 - **Help:** FAQ and request org account
- **Placement Editor:** Able to move, rotate, resize the object prior to confirming placement.

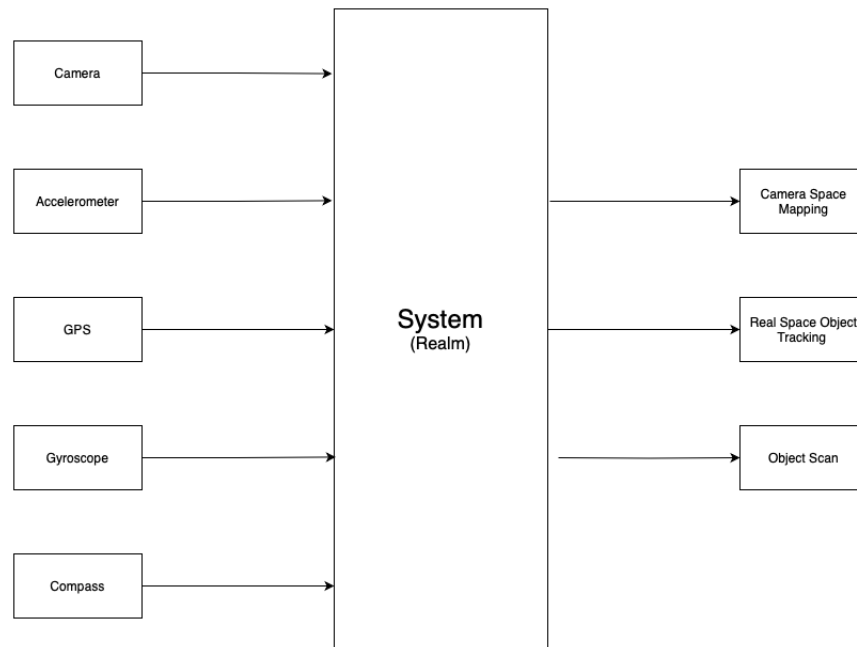
There must be a navigation menu that can be accessed from any page that allows the user to navigate to any of the four main pages: Realm, User Profile, Tours, and Map

3.1.2 Look and Feel Requirements

LF-NFR1

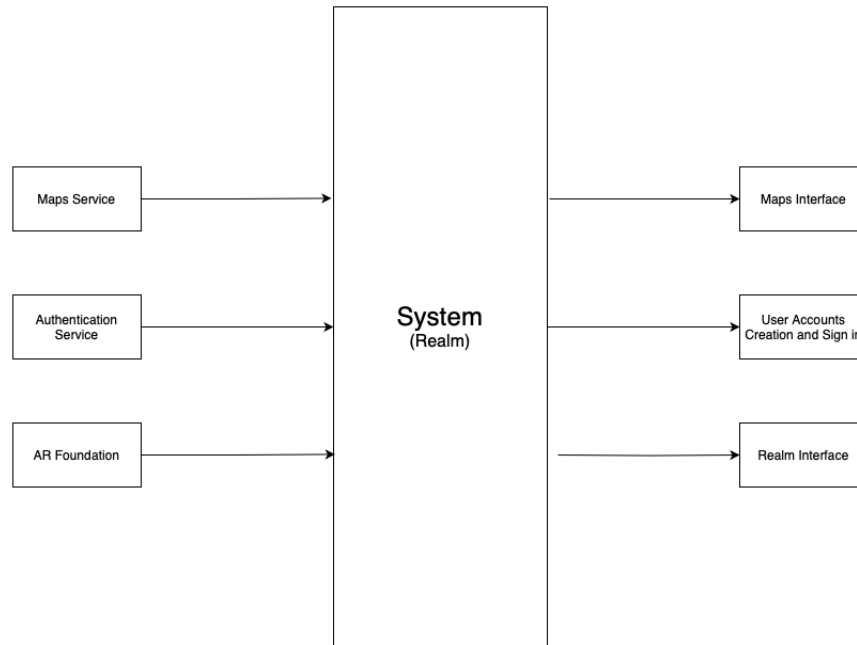
- Description: The Realm Screen must provide an immersive view of AR objects imposed on the real world, and must be minimally interrupted by other UI elements.
- Fit Criteria: Permanent UI elements besides the camera feed should take no more than 10% of the space on the screen.

3.1.3 Hardware interfaces



The above figure shows the hardware inputs of the system and the outputs of the system that directly follow from the hardware inputs.

3.1.4 Software interfaces



The above figure shows the external systems our system will interact with, and the outputs of the system that directly follow from the external software inputs

3.2 Functional

3.2.1 Tutorial

TU-FR1: The app shall have a step-by-step interactive guide of how to use all major app features.

TU-FR2: The app shall prompt the user to complete the app tutorial after they create an account.

TU-FR3: The app shall allow the user to leave the tutorial at any time.

TU-FR4: The tutorial shall be available at any time through the app's help page in the **Settings screen**.

TU-FR5: The tutorial will involve user participation to directly use functionality in a sandbox environment.

TU-NFR1: Each interaction step should not take the user more than 15 seconds to figure out.

TU-NFR2: The entire tutorial shall not take longer than 5 minutes to complete for 80% of users.

3.2.2 Tour Management

TM-FR1: The tour management functionality within the app shall only be available to [Organization users](#).

TM-FR2: Tours within the app can be created as a “draft” to make it available to other [Organization users](#) but not released to the public.

TM-FR3: Tours within the app can be published to the public from a “draft” or from directly after creation.

TM-FR4: [Organization users](#) will be able to customize the following for each of their tours:

- (a) Name
- (b) Description
- (c) Route
 - i. Will be editable on a map of the tour area
 - ii. Intended direction of travel can be set
 - iii. Estimated time of completion that will be automatically determined by the distance if not set
- (d) Objects
 - i. [AR object](#) can be placed along the route at specific geographic locations along with description text for each of the [AR object](#)
 - ii. Historical information text popups can be placed along the route
 - iii. All text will have an audio playback option with text-to-speech or pre-recorded audio (if available)
- (e) Price
 - i. Can be free or
 - ii. a price under \$10
- (f) Relevant Web Link(s)

TM-FR5: [Organization users](#) will have the ability to preview the tours that belong to their organization.

TM-FR6: [Organization users](#) will have the ability to edit tours that belong to their organization.

3.2.3 Touring

TR-FR1: The touring functionality within the app shall only be available to [General users](#).

TR-FR2: The app will have three avenues for a user to find tours:

TR-FR2.1: The app will allow [General users](#) to see a list of available tours through the **Tour List Interface**.

- The tours can be grouped in this page by organization or by location

TR-FR2.2: The app will also have tours show up in a push notification (if configured by a user) when in close proximity to a tour area.

TR-FR2.3: Locations can place QR codes at the starting location of the tour which can be scanned by a mobile camera that will open the tour preview in the app.

TR-FR3: Users will be able to preview a tour to see the following information:

- (a) Name
- (b) Description
- (c) Relevant web link(s)
- (d) Tour distance (auto-calculated from route)
- (e) Estimated time of completion
- (f) A map that will show the route and locations of [AR objects](#)
- (g) Price

TR-FR4: Once a user starts a tour they will have two main views they can switch between:

TR-FR4.1: One of the tour views is a map

- The designated tour area will be outlined
- The user's current location will be shown
- Intended route and direction will be overlaid
- Location of [AR objects](#) will be marked

TR-FR4.2: One of the tour views is an AR view

- Will be very similar to the interface of the **Realm Interface**
- Have an added indicator of the intended direction
- Popups for historical information

3.2.4 Object Placement

OP-FR1: The system must have the capability to store object instances that specify the following details:

- The AR object that was placed.
- The sub-realm(s) in which the object instance will be shared.
- All data required to precisely reproduce the position and orientation of the object in real space.

OP-FR2: The system must provide a way for a user to place AR objects in the space around them, creating object instances.

OP-FR2.1: The system must provide a way for a user to select an object either from their inventory or by generating a new object.

OP-FR2.2: The system must provide a way for the user to select one or more sub-realms of which they are a member.

OP-FR2.3: The system must provide a way for the user to position an AR object in real space.

3.2.5 Realm Interface

RI-FR1: The Realm interface must provide a view of the AR object instances in the area around the user overlaid on their camera feed positioned in real space.

RI-FR1.1: AR object instances presented on the Realm interface must always appear to change perspective correctly with respect to the user's camera position and angle.

RI-FR1.2: AR object instances that overlap or clutter a given area in real space will not be presented simultaneously.

RI-FR1.3: AR object instances reproduced on the Realm interface must be in the correct position and orientation as defined by the object placement.

RI-FR2: The Realm interface must keep track of a selected sub-realm which determines which object instances are displayed.

RI-FR2.1: The Realm interface must provide an indication of what sub-realm is currently selected.

RI-FR2.2: The Realm interface must provide a way for the user to change the selected sub-realm.

RI-FR3: The Realm interface must provide a control to allow the user to enter the object placement workflow.

RI-FR4: The Realm interface must provide a control to allow the user to enter the object scanning workflow.

RI-FR5: When the user is near the starting point of a tour, the Realm interface must provide an indication that there is a tour nearby as well as a link to the tour preview.

3.2.6 Admin Interface

AI-FR1: The admin interface within the app shall only be available to **Admins**.

AI-FR2: The app shall provide an interface for **Admins** to carry-out their two special roles:

AI-FR2.1: The app shall permit **Admins** to act on user reports of an **AR object** by either keeping or removing the **AR object**.

AI-FR2.2: The app shall permit **Admins** to review requests for **Organization users** accounts and either accept or deny the request.

3.3 Use Cases

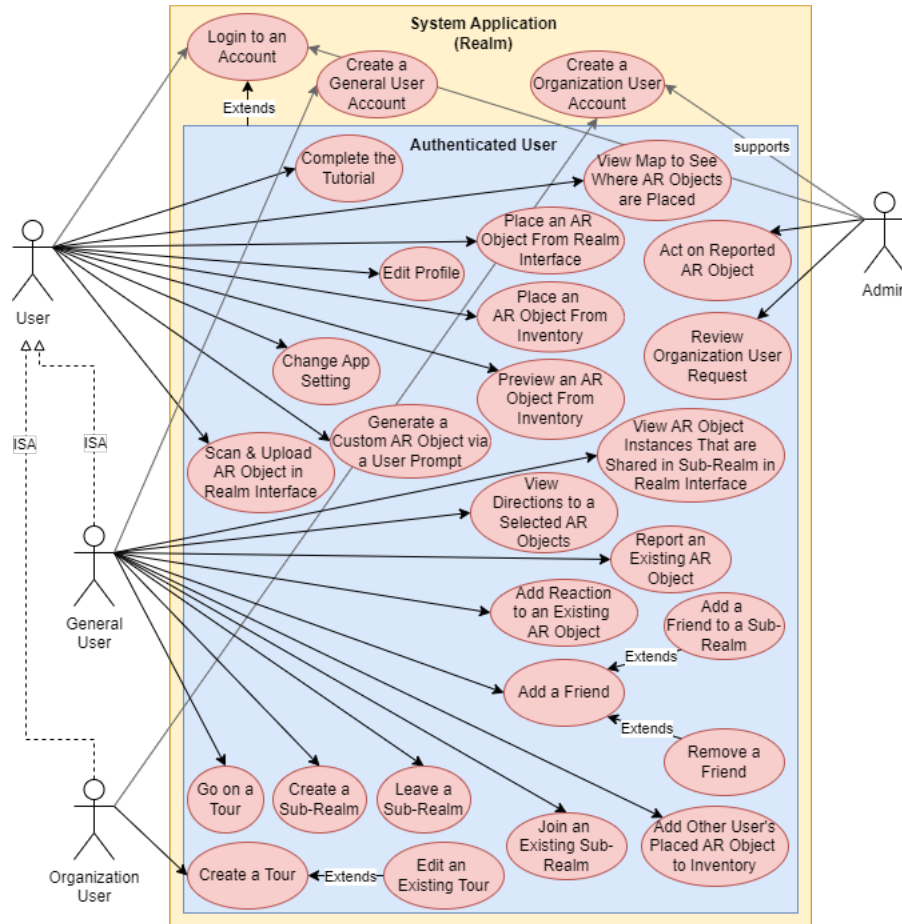


Figure 1: Use Cases

UC1 Complete the Tutorial

Actor: [Users](#)

Pre-condition: User has opened the app

Main Success Scenario:

1. User creates an account (??)
2. System prompts user to complete the tutorial ([TU-FR2](#))
3. User indicates they would like to do the tutorial
4. System opens the **Tutorial Interface**
5. System provides directions to use a feature
6. User tries the feature ([TU-FR5](#))
7. User goes to next feature when satisfied
8. Repeat steps **5-7** for all major app features ([TU-FR1](#))
9. System prompts user to leave and end tutorial
10. User ends tutorial ([TU-FR3](#))
11. System redirects to **Home Interface**

Secondary Scenarios:

1.1: User already has an account

1. User logs into account (??)
2. User navigates to **Help Interface** ([TU-FR4](#))
3. Go to step **2**

10.1: User chooses to stay in sandbox environment

1. System closes prompt and allows user to use sandbox environment
2. User indicates they want to leave tutorial
3. Go to step **11**

Success Postcondition: The user has a general idea of how all the app's major features work.

UC2 Create a Tour

Actor: [Organization users](#)

Pre-condition: User is logged into the app **AND** User has navigated to the **Tour Management Interface**

Main Success Scenario:

1. User selects the option to create a new tour

2. System opens up a form for the user to add the following information about the tour (TM-FR4[1,2,5,6]):
 - i. Name
 - ii. Description
 - iii. Price
 - iv. Relevant web link(s)
3. User fills out the information
4. System identifies no blank fields
5. System gives the option to move on to the route configuration
6. User goes to the route configuration
7. System presents tour route configuration
8. User configures tour area, route, and direction (TM-FR4[3])
9. System gives the option to move on to the inventory setup
10. User goes to the inventory setup (TM-FR4[4])
11. System opens the **Inventory Interface**
12. User scans and uploads a new AR object (??)
13. User repeats step 12 until all desired AR objects are uploaded
14. User indicates they would like to move on to object placement
15. System opens the **Realm Interface** and allows the user to place AR objects and historical information popups in an isolated environment.
16. User places down AR object using the **Realm Interface** (UC5)
17. User repeats step 16 until all desired AR objects are placed
18. User indicates they would like to finish tour creation
19. System gives the user the option create the tour as a draft or directly publish it (TM-FR2/TM-FR3)
20. User selects direct publish option
21. System uploads the tour data
22. System makes the tour available to General users

Secondary Scenarios:

- 4.1: System identifies some blank fields
 1. System validates input to ensure all required fields are not empty
 2. System determines input passes validation
 3. Go to step 5
- 4.1.2: System determines input fails validation
 1. Go to step 3
- 12.1: User does not have any AR objects to scan

1. Go to step 14

20.1: User selects draft option

1. System uploads the tour data but does not make it available to [General users](#)
2. User decides at a later time to publish the draft tour
3. Go to step 22

Success Postcondition: One of the [Organization users](#) has a new tour linked to their organization

UC3 Edit an existing Tour

Actor: [Organization users](#)

Pre-condition: User is logged into the app **AND** User has at least one tour connected to their organization **AND** User has navigated to the **Tour Management Interface**

Main Success Scenario:

1. System provides a list of all tours connected to the user's organization
2. User selects one of the tours
3. System shows the preview of the tour ([TM-FR5](#))
4. System provides the option to edit the tour ([TM-FR6](#))
5. User decides to edit the tour
6. System opens up a form for the user to with the following information about the tour that may or not be populated ([TM-FR4\[1,2,5,6\]](#)):
 - i. Name
 - ii. Description
 - iii. Price
 - iv. Relevant web link(s)
7. User edits information as desired
8. System identifies no blank fields
9. System gives the option to move on to the route configuration
10. User goes to the route configuration
11. System presents tour route configuration
12. User edits configuration of tour area, route, and direction as desired ([TM-FR4\[3\]](#))
13. System gives the option to move on to the inventory setup
14. User goes to the inventory setup ([TM-FR4\[4\]](#))
15. System opens the **Inventory Interface**
16. User scans and uploads a new [AR object](#) (??)

17. User repeats step 12 until all desired AR objects are uploaded
18. User indicates they would like to move on to object placement
19. System opens the **Realm Interface** and allows the user to place/edit AR objects and historical information popups in an isolated environment.
20. User edits AR object using the **Realm Interface** (UC5)
21. User repeats step 16 until all desired AR objects are edited
22. User indicates they would like to finish tour editing
23. System gives the user the option to save the edited tour as a draft or directly publish it (TM-FR2/TM-FR3)
24. User selects direct publish option
25. System uploads the updated tour data
26. System makes the updated tour available to General users

Secondary Scenarios:

- 5.1: User decides not to edit the tour
 1. System goes back to the **Tour Management Interface**
- 8.1: System identifies some blank fields
 1. System validates input to ensure all required fields are not empty
 2. System determines input passes validation
 3. Go to step 9
- 16.1: User decides not to scan and upload any new AR objects
 1. Go to step 18
- 20.1: User places down new AR object using the **Realm Interface**
 1. User repeats step 20.1 until all desired AR objects are placed
- 24.1: User selects draft option
 1. System uploads the updated tour data but does not make it available to General users
 2. User decides at a later time to publish the draft of the updated tour
 3. Go to step 26

Success Postcondition: One of the Organization users has edited an existing tour linked to their organization

UC4 Go on a tour

Actor: General users

Pre-condition: User is logged into the app

Main Success Scenario:

1. User navigates to the **Tour Interface**
2. System displays a list of all tours
3. User selects a tour (TR-FR2.1)
4. System opens up the preview of the selected tour (TR-FR3)
5. System also provides the option to take the tour
6. User reviews details of the tour
7. User decides to go on the tour
8. System opens a modified version of the **Realm Interface** in an isolated environment that does not allow them to place any objects or modify existing objects (TR-FR4.2)
9. System shows the intended direction of travel
10. User walks to the highlighted **AR object** of interest
11. System acknowledges when the user makes it to the **AR object**
12. System provides historical information regarding **AR object**
13. System updates the target to be the next **AR object**
14. Repeat steps 10-13 for every **AR object**
15. System indicates that the tour has been completed
16. System provides metrics for time taken and distance traveled
17. System prompts user to leave tour
18. User leaves tour
19. System goes back to the tour list

Secondary Scenarios:

- 1.1: System pushes a notification to the user when they are in the proximity of a tour (TR-FR2.2)
 1. System opens up the **Tour Interface**
 2. Go to step 4
- 1.2: User scans a tour QR code in camera app (TR-FR2.3)
 1. System opens up the **Tour Interface**
 2. Go to step 4
- 7.1: User decides not to go on the tour
 1. Go to step 19
- 10.1: User indicates they want to open the map
 1. System opens up the **Map** view of the tour (TR-FR4.2)
 2. System shows the user's current location
 3. System shows the indented route, direction and boundaries of the tour

4. System marks the location of **AR objects**
5. User views the map
6. User closes the map
7. Go to step 9
- 14.1: User does not want to go to all **AR objects**
 1. User indicates they wish to exit the tour
 2. Go to step 19
- 18.1: User does not leave tour
 1. System removes target object and allows user to explore **AR objects** in the isolated environment
 2. User eventually decides to exit the tour
 3. Go to step 19

Success Postcondition: One of the **General users** has completed a tour

UC5 Place an AR object from the **Realm Interface (RI-FR1)**

Actor: General User

Pre-condition: User is logged in to the app **AND** has navigated to the **Realm Interface (RI-FR1)**

Main Success Scenario:

1. System presents **Realm interface**.
2. User performs control to initiate object placement.
3. System presents the **Object Selection Menu (OP-FR2.1)**.
4. User selects an existing AR object for placement.
5. System presents the **Sub-Realm Selection Menu (OP-FR2.2)**.
6. User selects the sub-realm(s) in which they wish for their object instance to be shared.
7. System presents the **Object Positioning Interface (OP-FR2.3)**.
8. User positions the AR object using given controls and confirms when done.
9. System returns to Realm screen.

Secondary Scenarios:

- 3.1: User creates new object with a prompt:
 1. Main scenario steps 1-3.
 2. User selects the option to generate an object via prompt.
 3. System presents the **Prompt Object Generation Interface**.
 4. User completes the prompt object generation workflow.
 5. Main scenario steps 5-9.

3.2: User creates a new object with object scanning:

1. Main scenario steps 1-3.
2. User selects the option to generate an object via object scan.
3. System presents the **Object Scanning Interface**.
4. User completes object scanning workflow.
5. Main scenario steps 5-9.

3.3, 5.1, 7.1: User cancels object placement:

1. Main scenario steps 1-3, or 1-5, or 1-7.
2. User selects option to cancel object placement.
3. System returns to the **Realm Interface**.

5.1: User reselects object:

1. Main scenario steps 1-5.
2. User selects option to return to **Object Selection Menu (OP-FR2.1)**.
3. Main scenario resumes from step 3.

7.2: User reselects Sub-realms:

1. Main scenario steps 1-7.
2. User selects option to return to **Sub-Realm Selection Menu (OP-FR2.2)**.
3. Main scenario resumes from step 5.

Success Postcondition: Users that are members of the sub-realm in which the object instance has been shared can see the object instance from the **Realm interface**.

UC6 Place an AR object from the Inventory

Actor: General User

Pre-condition: User is logged in to the app **AND** has navigated to the **Inventory Interface**

Main Success Scenario:

1. System presents Inventory interface.
2. User selects an object they wish to place.
3. System presents object details.
4. User selects option to initiate object placement.
5. System presents the **Sub-Realm Selection Menu (OP-FR2.1)**.
6. User selects the sub-realm(s) in which they wish for their object instance to be shared.
7. System presents the **Object Positioning Interface (OP-FR2.3)**.

8. User positions the AR object using given controls and confirms when done.
9. System returns to Realm screen.

Secondary Scenarios:

5.1, 7.1: User cancels object placement:

1. Main scenario steps 1-5, or 1-7.
2. User selects option to cancel object placement.
3. System returns to the Inventory screen.

7.2: User reselects Sub-realm:

1. Main scenario steps 1-7.
2. User selects option to return to **Sub-Realm Selection (OP-FR2.2)**.
3. Main scenario resumes from step 5.

Success Postcondition: Users that are members of the sub-realm in which the object instance has been shared can see the object instance from the **Realm Interface**.

UC7 View object instances that are shared in a Sub-realm in the **Realm interface**

Actor: General User

Pre-condition: User is logged in to the app **AND** has navigated to the **Realm Interface (RI-FR1)**

Main Success Scenario:

1. System presents the **Realm Interface**.
2. User selects a sub-realm to see objects from.
3. System only presents object instances in the selected sub-realm.
4. User moves their device to see any angle of the object instance.
5. System continuously presents the object instance with the correct perspective relative to the user's camera.

Secondary Scenarios:

10.1: User is not a member of any sub-realms:

1. System presents **Realm Interface (RI-FR1)**.
2. System presents only public object instances.
3. User moves their device to see any angle of the object instance.
4. System continuously presents the object instance with the correct perspective relative to the user's camera.

Success Postcondition: The user can see any AR object instances that are in any sub-realm that they are a part of correctly positioned in real space.

UC8 Report an existing AR object

Actor: General User

Pre-condition: User is logged in to the app **AND** has navigated to the **Realm Interface** (RI-FR1) **AND** there is another user's AR object placed in the vicinity of the user

Main Success Scenario:

1. System presents **Realm Interface** with another user's object instance present on screen.
2. User selects the object they wish to report from the screen.
3. System presents **Object Selection Context Action Menu**.
4. User selects option to report object.
5. System presents **Object Reporting Interface**.
6. User provides required details for report and submits report.
7. System presents message thanking user for report.

Secondary Scenarios:

5.1: User cancels report:

1. User closes **Object Reporting Interface**.
2. System returns to **Realm Interface**.

3.1: User cancels object selection:

1. User closes **Object Selection Context Action Menu**.
2. Main scenario resumes from step 2.

Success Postcondition: The user is informed that their report will be investigated. The system initiates a report review process.

UC9 Add reaction to an existing AR object instance

Actor: General User

Pre-condition: User is logged in to the app **AND** has navigated to the **Realm Interface** (RI-FR1) **AND** there is an AR object placed in the vicinity of the user

Main Success Scenario:

1. System presents **Realm Interface** with an object instance present on screen.
2. User selects the object they wish to react to from the screen.

3. System presents **Object Selection Context Action Menu**.
4. User selects option to react to object instance.
5. System presents **Reaction Selection Menu**.
6. User selects desired reaction.
7. System returns to **Object Selection Context Action Menu** with user reaction displayed.

Secondary Scenarios:

3.1: User cancels object selection:

1. User closes **Object Selection Context Action Menu**.
2. Main scenario resumes from step 2.

Success Postcondition: The user's reaction is visible to any user who selects the AR object instance.

UC10 Add other user's placed AR object to Inventory

Actor: General User

Pre-condition: User is logged in to the app **AND** has navigated to the **Realm Interface** ([RI-FR1](#)) **AND** there are other users' AR objects placed in the vicinity of the user

Main Success Scenario:

1. System presents **Realm Interface** with another user's object instance present on screen.
2. User selects the object they wish to add to their inventory from the screen.
3. System presents **Object Selection Context Action Menu**.
4. User selects option to add object to inventory.
5. System returns to **Realm Interface** with indication that object has been added to Inventory.

Secondary Scenarios:

4.1: User inventory is full:

1. System returns to **Realm Interface** with indication that Inventory is full.

3.1: User cancels object selection:

1. User exits **Object Selection Context Action Menu**.
2. System returns to **Realm Interface**.

Success Postcondition: The user's desired object is available in the user's inventory for preview and placement.

UC11 Act on reported [AR object](#)

Actor: [Admins](#)

Pre-condition: Admin is logged into the app **AND** Admin has navigated to the **Admin Interface**

Main Success Scenario:

1. Admin navigates to the “AR Object Reports” panel ([AI-FR2.1](#))
2. System shows a list of app [AR objects](#) that have been reported
3. Admin selects an object
4. System gives more detailed information of [AR object](#) and allows Admin to open an [AR object](#) inventory preview (IV-FR6)
5. System also provides decision options on whether to keep the object or remove it
6. Admin opens [AR object](#) preview
7. System shows preview of [AR object](#)
8. Admin views [AR object](#) preview
9. Admin closes [AR object](#) preview
10. System once again shows the object detailed information and decision options
11. Admin reviews the information and decides to keep the [AR object](#)
12. System keeps [AR object](#)
13. System removes user report of [AR object](#)
14. System brings admin back to “AR Object Reports” panel

Secondary Scenarios:

- 5.1:** Admin does not open [AR object](#) preview
1. Go to step 11
- 11.1:** Admin reviews the information and decides to remove the object
1. System removes object
 2. Go to step 13

Success Postcondition: An [AR object](#) report is resolved ([AR object](#) is kept or removed)

UC12 Act on reported [AR object](#)

Actor: [Admins](#)

Pre-condition: Admin is logged into the app **AND** Admin has navigated to the **Admin Interface**

Main Success Scenario:

1. Admin navigates to the “Organization User Requests” panel ([AI-FR2.2](#))
2. System shows a list of [Organization users](#) requests
3. Admin selects a request
4. System gives more details about a request
5. System also provides decision options on whether to accept or decline a request
6. Admin reviews information and approves the request
7. System activates the user as one of the [Organization users](#)
8. System notifies the user that their request has been approved
9. System removes organization user request
10. System brings admin back to “Organization User Requests” panel

Secondary Scenarios:

- 6.1:** Admin reviews information and denies the request
1. System notifies the user that their request has been denied and gives a reason why
 2. Go to step 9

Success Postcondition: An [Organization users](#) account request is resolved (approved or denied).

3.4 Quality of Service

3.4.1 Performance

3.4.2 Security

3.4.3 Reliability

3.4.4 Availability

3.5 Compliance

CO1. The project shall comply with the *Personal Information and Electronic Documents Act* (PIPEDA).

Rationale: The Government of Canada requires all companies to follow certain rules regarding the collection, use, and dissemination of personal user information [?].

CO2. The project shall keep records of all in-app purchases and ad revenue for the purposes of yearly tax filing for the period of six years.

Rationale: Corporate taxes must be filed every year and both streams of app income will need to be reported. Businesses must keep records going back six years in the event of an audit [?].

- CO3.** The app shall comply with the *Google Play* developer policy.
Rationale: All apps published through *Google Play* must first be reviewed by Google for compliance with the developer policy published on their website [?].
- CO4.** The app shall comply with *App Store* review guidelines.
Rationale: For an app to be approved for dissemination on the *App Store*, the app must be reviewed and approved by Apple in accordance with the acceptance criteria published on their website [?].

3.6 Design and Implementation

3.6.1 Installation

- DI-I1.** The app shall be installable on *Android* and *iOS* devices from their respective app stores.
Rationale: Users are accustomed to downloading their apps from their device app store and should not be required to navigate to a 3rd party app store. They also should not have to change OS settings in order to download the app.
- DI-I2.** The app shall not require any additional installation steps beyond those required from within the target device app store.
Rationale: Users may be dissuaded from downloading the app if the installation process is too cumbersome compared to other apps.

3.6.2 Distribution

- DI-D1.** The app shall be distributed on any mobile devices running iOS 16.0+ or Android 12+
Rationale: The app should be available to as many people as possible while at the same time making the development easier by not having to keep old operating system versions supported. Versions should be supported for at least a couple years.
- DI-D2.** The app shall be available in Canada and the USA.
Rationale: Due to legal considerations in different countries, the focus for this app should be the country this project is based out of, Canada, and the USA since they have a larger population with similar laws.
- DI-D3.** The app shall have a recommended age requirement of 16+.
Rationale: Users may be exposed to content not suitable for really young kids so an age requirement should be recommended.
- DI-D4.** The system shall store all user data within North America.
Rationale: Data should be located in a jurisdiction close to home and in a reputable country to reduce privacy concerns of foreign state actors viewing user data.

3.6.3 Maintainability

3.6.4 Reusability

3.6.5 Portability

DI-P1. The app shall be developed using a cross-platform mobile platform that can build iOS and Android applications

Rationale: To reach as many potential users as possible, the app should be available on the two major mobile operating systems.

DI-P2. The app shall have a common codebase that only differs in configuration files for different target operating systems.

Rationale: For ease of development on the different mobile operating systems, there should be no extra consideration for the differences between native app implementations that are not handled by the cross-platform framework.

3.6.6 Cost

3.6.7 Deadline

3.6.8 Proof of Concept

4 Verification

5 Appendixes

5.1 Activity Diagrams

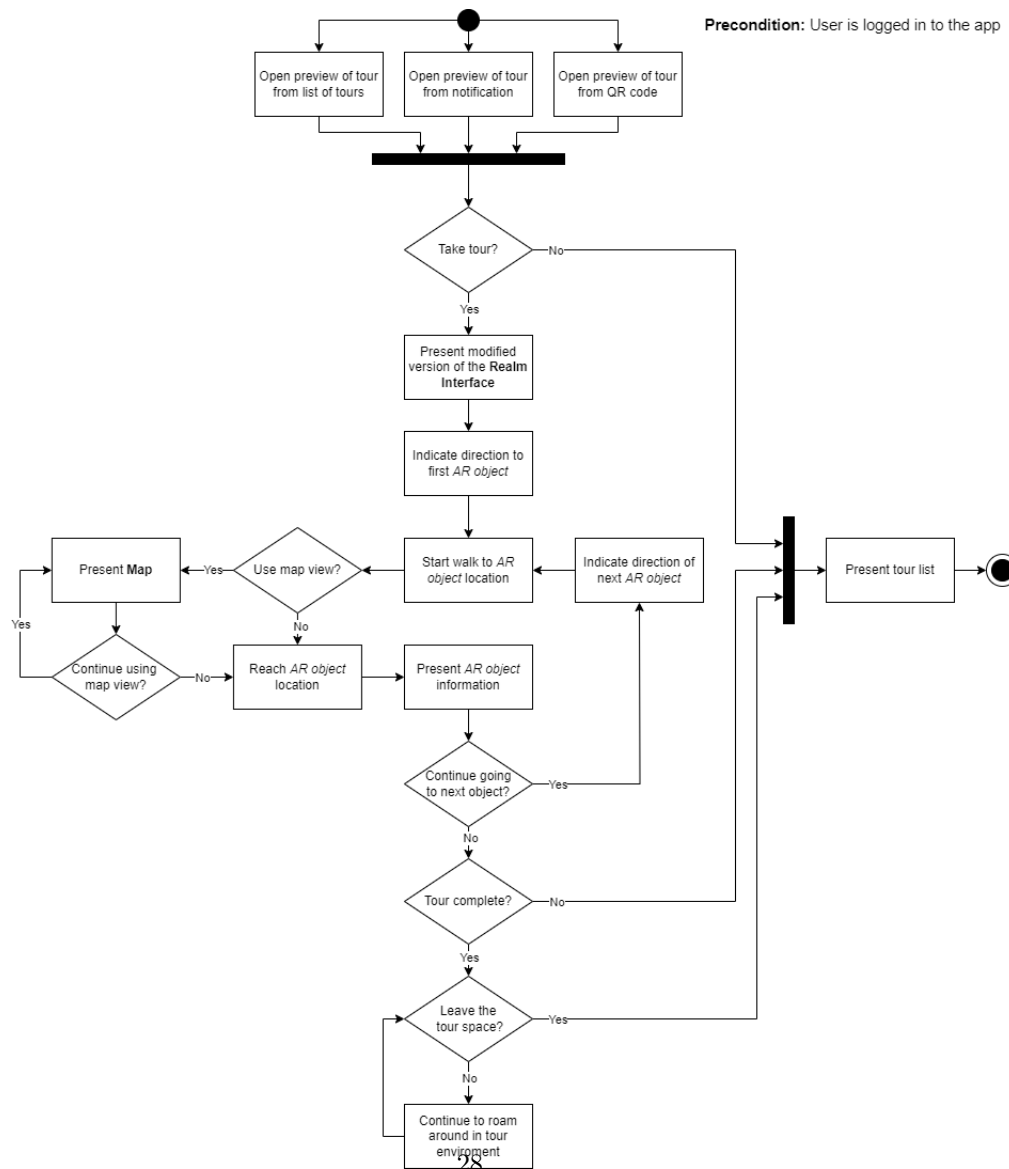


Figure 2: UC4

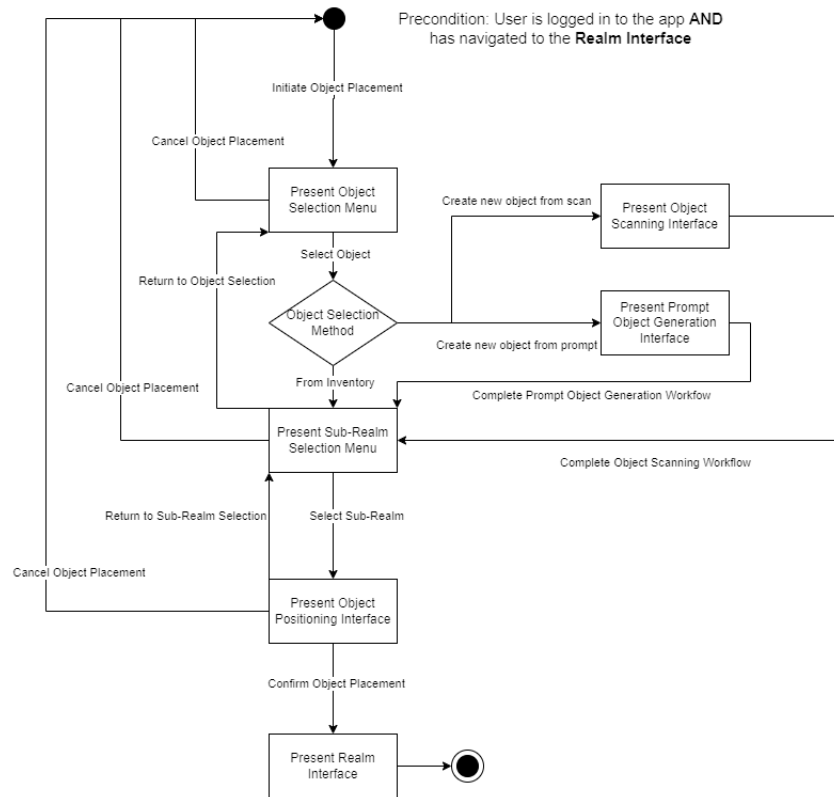


Figure 3: UC7