

# Software Requirements Specification for Software Eng: Mac AR

Team #11, Mac-AR  
Student 1 Matthew Collard  
Student 2 Sam Gorman  
Student 3 Ethan Kannampuzha  
Student 4 Kieran Gara

October 2, 2023

# Contents

<b>1</b>	<b>Project Drivers</b>	<b>1</b>
1.1	The Purpose of the Project . . . . .	1
1.2	The Stakeholders . . . . .	1
1.2.1	The Client . . . . .	1
1.2.2	The Users . . . . .	1
1.3	Mandated Constraints . . . . .	1
1.3.1	Solution Constraints . . . . .	1
1.3.2	Implementation Environment of the Current System . . . . .	2
1.3.3	Partner or Collaborative Applications . . . . .	3
1.3.4	Off-the-Shelf Software . . . . .	3
1.3.5	Anticipated Workplace Environment . . . . .	3
1.3.6	Schedule Constraints . . . . .	3
1.3.7	Budget Constraints . . . . .	3
1.3.8	Enterprise Constraints . . . . .	4
1.4	Naming Conventions and Terminology . . . . .	4
1.5	Relevant Facts and Assumptions . . . . .	4
1.5.1	User Assumptions . . . . .	4
1.5.2	Device Assumptions . . . . .	5
<b>2</b>	<b>Functional Requirements</b>	<b>5</b>
2.1	The Scope of the Work and the Product . . . . .	5
2.1.1	The Current Situation . . . . .	5
2.1.2	Context of the Work . . . . .	5
2.1.3	Individual Product Use Cases . . . . .	7
2.2	Functional Requirements . . . . .	10
2.2.1	Create Game Room . . . . .	10
2.2.2	Join Game Room . . . . .	10
2.2.3	Edit Room Settings . . . . .	11
2.2.4	Exit Room . . . . .	12
2.2.5	Start Game . . . . .	12
2.2.6	Save Game . . . . .	12
2.2.7	Load Game . . . . .	13
2.2.8	Delete Save . . . . .	13
2.2.9	Puzzle Interaction . . . . .	14
2.2.10	Hint Request . . . . .	15
2.2.11	Skip Puzzle . . . . .	15
2.2.12	Send Message . . . . .	15
2.2.13	Read Message . . . . .	16
2.3	Formal Specification . . . . .	16

<b>3</b>	<b>Non-functional Requirements</b>	<b>16</b>
3.1	Look and Feel Requirements . . . . .	16
3.2	Usability and Humanity Requirements . . . . .	16
3.3	Performance Requirements . . . . .	16
3.4	Operational and Environmental Requirements . . . . .	17
3.5	Maintainability and Support Requirements . . . . .	17
3.6	Security Requirements . . . . .	17
3.7	Cultural Requirements . . . . .	17
3.8	Legal Requirements . . . . .	17
3.9	Health and Safety Requirements . . . . .	17
<b>4</b>	<b>Traceability Matrix</b>	<b>18</b>
<b>5</b>	<b>Project Issues</b>	<b>18</b>
5.1	Open Issues . . . . .	18
5.2	Off-the-Shelf Solutions . . . . .	18
5.2.1	Ready-Made Products . . . . .	18
5.2.2	Reusable Components . . . . .	18
5.2.3	Products That Can Be Copied . . . . .	18
5.3	New Problems . . . . .	19
5.3.1	Effects on the Current Environment . . . . .	19
5.3.2	Effects on the Installed Systems . . . . .	19
5.3.3	Potential User Problems . . . . .	19
5.3.4	Limitations in the Anticipated Implementation Environment That May Inhibit the New Product . . . . .	19
5.3.5	Follow-Up Problems . . . . .	19
5.4	Tasks . . . . .	20
5.4.1	Project Planning . . . . .	20
5.4.2	Planning of the Development Phases . . . . .	20
5.5	Migration to the New Product . . . . .	20
5.5.1	Requirements for Migration to the New Product . . . . .	21
5.5.2	Data That Has to Be Modified or Translated for the New System . . . . .	21
5.6	Risks . . . . .	21
5.7	Costs . . . . .	21
5.8	User Documentation and Training . . . . .	21
5.8.1	User Documentation Requirements . . . . .	21
5.8.2	Training Requirements . . . . .	21
5.9	Waiting Room . . . . .	21
5.10	Ideas for Solutions . . . . .	22

<b>6</b>	<b>Appendix</b>	<b>23</b>
6.1	Reflection . . . . .	23
6.2	Symbolic Parameters . . . . .	24

Table 1: Revision History

<b>Date</b>	<b>Developer(s)</b>	<b>Change</b>
September 26, 2023	Matthew and Sam	Initial changes to the project
September 26, 2023	Matthew	Initial NFR changes
September 28, 2023	All	General format editing and addition of the Use Case Diagram
September 29, 2023	Matthew	Addition of Use cases 1-5 and related functional requirements
September 29, 2023	Kieran	Addition of Use cases 5-10 and related functional requirements
September 29, 2023	Ethan	Added Read-Made Products section to Project Issues
September 30, 2023	Ethan	Completed rest of sections in Project Issues
October 1, 2023	Sam	Filled out sections 1, 6

This document describes the requirements for Mac-AR. The template for the Software Requirements Specification (SRS) is a subset of the Volere template ([Robertson and Robertson, 2012](#)). If you make further modifications to the template, you should explicitly state what modifications were made.

# 1 Project Drivers

This section will give a general overview of the motivation behind creating Mac-AR, along with what constraints it needs to be created within.

## 1.1 The Purpose of the Project

In the modern world, it has become increasingly harder to meet new people and form connections with others. This has become especially true in recent years, with the pandemic isolating people even further. Mac-AR aims to solve this by providing a fun, interactive means to cooperate with others and form new connections. As a team, users will be able to work alongside people they already know and strangers around the world to solve interesting puzzles, either outside or from the comfort of their own home.

## 1.2 The Stakeholders

The following groups and individuals are the reason for the existence of Mac-AR.

### 1.2.1 The Client

Dr Irene Yuan is the sole client for the project, requesting that the tool be made. She will act as the point of contact during development and ensure that the final result is in line with what was envisioned.

### 1.2.2 The Users

The users are the ones who will be using the tool and building connections with others. The main user base is intended to be young adults, roughly ages 18-30, who enjoy games and are moderately comfortable using technology.

## 1.3 Mandated Constraints

The project will be developed and released within the following constraints.

### 1.3.1 Solution Constraints

*Description:* The product shall be built as a mobile application.

*Rationale:* The supervisor wants the application to be a mobile application, as to ensure the

largest freedom to the users for where and how the product will be used.

*Fit Criterion:* The product shall be written using Unity engine.

*Description:* The product shall be able to function on a variety of mobile devices, such as iPhones and Androids.

*Rationale:* To provide service to the widest array of users, the most popular phone brands, and by extension their operating systems, need to be supported.

*Fit Criterion:* The product shall be tested to function mobile applications, which includes the following devices:

- iPhone SE
- iPhone XR
- iPhone 12 Pro
- Pixel 5
- Samsung Galaxy S8+
- Samsung Galaxy S20 Ultra
- iPad Air
- iPad Mini
- Surface Pro 7
- Surface Duo
- Galaxy Fold
- Samsung Galaxy A51/71
- Nest Hub
- Nest Hub Max

However, due to timing constraints, testing will only be run on the most popular devices. This would consist of the iPhone, Pixel, and Samsung phones, as well as iPad and Galaxy tablets. For iPhone, which is much more restrictive with backwards compatibility, the most recent IOS versions will be tested, namely 15, 16, and 17.

### **1.3.2 Implementation Environment of the Current System**

As outlined in section [1.3.1](#), the tool will need to be developed for execution on modern mobile devices. There are currently no constraints on the server side of the project, although that may need to be revised as the tool grows in scope.

### 1.3.3 Partner or Collaborative Applications

In the current design of the product, there are no partner or collaborative applications that will work along with the product. Therefore, there are no partner or collaborative constraints.

### 1.3.4 Off-the-Shelf Software

The following off-the-shelf software will be utilized:

- Unity, along with its available libraries for AR app development and communication
- Visual Studio, for development

### 1.3.5 Anticipated Workplace Environment

The anticipated workplace environment will be intentionally broad. The product can be used from anywhere the user has access to a mobile device and internet to run the application, with no limits on being indoors or outdoors.

### 1.3.6 Schedule Constraints

The general document and development milestone constraints are as follows:

Table 2: Schedule Constraints

Date	Deliverable
Sept 25, 2023	Problem Statement
Sept 25, 2023	Development Plan
Oct 5, 2023	Requirements Document
Oct 20, 2023	Hazard Analysis
Nov 3, 2023	Verification and Validation Plan
Nov 13-24, 2023	Proof of Concept Demo
Jan 17, 2024	Design Document
Feb 5-16, 2024	Revision 0 Demo
Mar 6, 2024	Verification and Validation Report
Mar 18-29, 2024	Final Demo (Rev 1)
Apr 4, 2024	Final Documentation

As a general constraint applied on top of all of these, all work should be done a few days in advance to allow for time to review.

### 1.3.7 Budget Constraints

The project has no monetary budget. If there are any necessary purchases for development, the cost shall be paid by the project members and reimbursed by the supervisor. Furthermore, these purchases may not exceed \$750.



### 1.3.8 Enterprise Constraints

The project currently has no enterprise constraints.

## 1.4 Naming Conventions and Terminology

Symbol	description
A	Assumption
AR	Augmented Reality
DD	Data Definition
GD	General Definition
GPS	Global Positioning System
GS	Goal Statement
IM	Instance Model
LC	Likely Change
PS	Physical System Description
R	Requirement
SRS	Software Requirements Specification
RS	Required Skill
Software Eng	Software Engineering
TM	Theoretical Model
VR	Virtual Reality
XR	Extended Reality

## 1.5 Relevant Facts and Assumptions

The product will be developed under the following assumptions.

### 1.5.1 User Assumptions

- A1 Users will have a connection to the internet for the duration of the time that they are actively using the app.
- A2 The user will provide consent for the app to access the camera on the device which is running the app.
- A3 The user will provide the app to access locations services and track where the user is while actively using the app.

### **1.5.2 Device Assumptions**

- A4 Devices will only have a maximum of one instance of the app running at any given time.
- A5 The device that the app is running on will have a functional camera.
- A6 The device that the app is running on will have a functional gyroscope, or equivalent technology to measure device tilt.
- A7 The device that the app is running on will have a functional GPS, or equivalent technology to measure device location.

## **2 Functional Requirements**

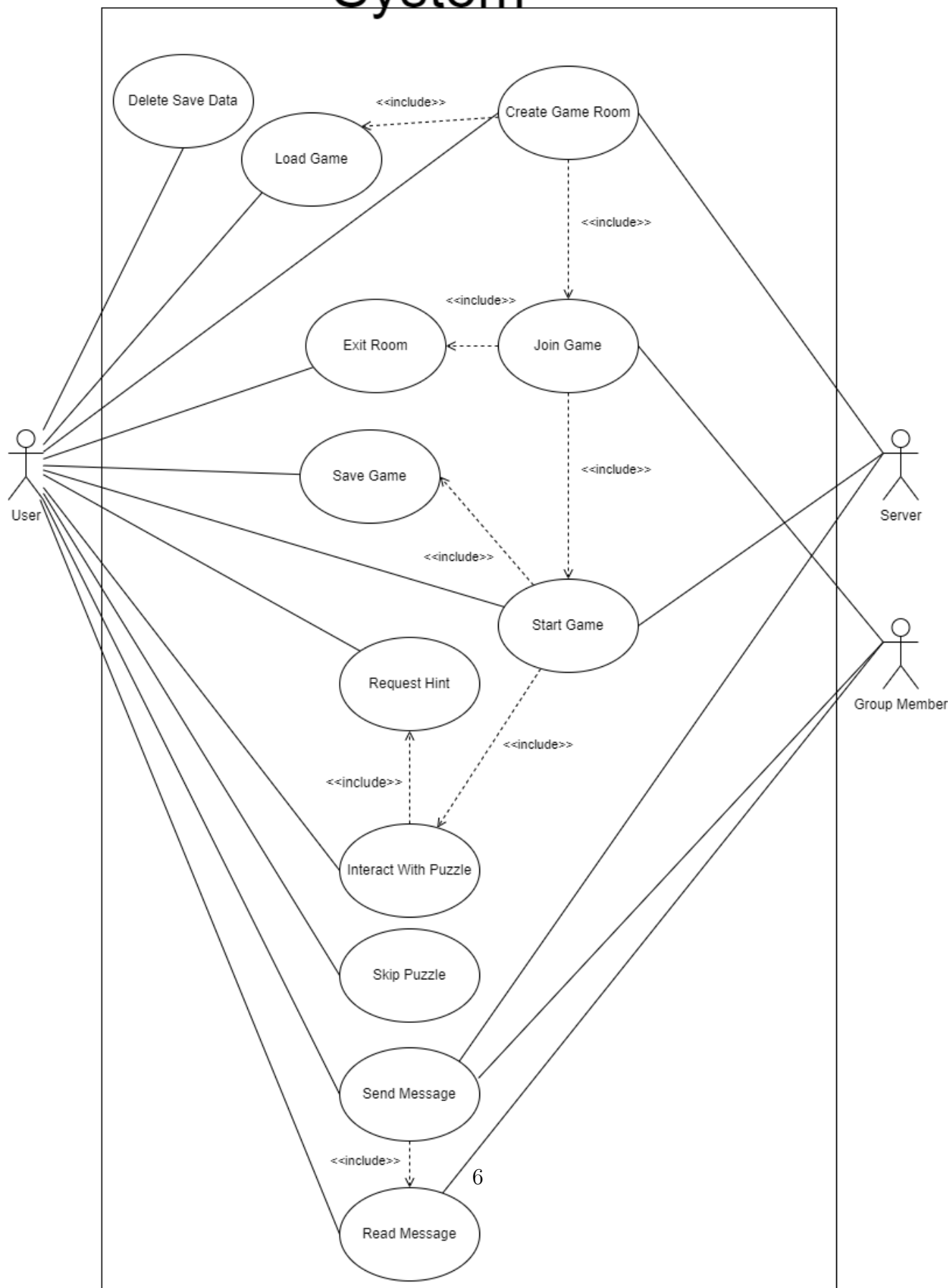
### **2.1 The Scope of the Work and the Product**

#### **2.1.1 The Current Situation**

#### **2.1.2 Context of the Work**

The context diagram depicted below illustrates the interactions of the system with adjacent external systems and services.

# System



### 2.1.3 Individual Product Use Cases

**Use Case #1:** User creates a game room

**Trigger:** The user selects the create game room button

**Pre-condition:** N/A

**Outcome:**

1. The system creates a room for users to join, and automatically makes the user join the room
2. The user can choose the length of the game and the settings
3. The server sends the data of the game room to all the users, to show that the room is available to join, and the capacity of the room, and the current users that have joined the room

**Use Case #2:** User Joins Game

**Trigger:** The user selects a room to join that has available capacity

**Pre-condition:** Another user has created a game room

**Outcome:**

1. The system links the user to the game room, and decreases the available capacity in the room
2. The server retrieves the data from the game room and updates the game room for all the users with the new user's name, and decreases the visible room capacity

**Use Case #3:** User edits the game room settings

**Trigger:** The user of the game room clicks the edit game settings button

**Pre-condition:** The user created the game room

**Outcome:**

1. The system opens the game room settings page
2. The settings are displayed for the user
3. The user chooses the settings they want to change
4. The system updates the game room settings
5. The server informs the other users of the settings changes
6. The user closes the settings page

**Use Case #4:** User exits the game room

**Trigger:** The user clicks the exit game room button

**Pre-condition:** The user is currently in a game room

**Outcome:**

1. The user is removed from the game room
2. The system updates the information of the game room
3. The server sends the updated information to each user in the game room
4. The user is brought back to the main menu

**Use Case #5:** User starts the game

**Trigger:** The users in a game room click the start game button

**Pre-condition:** The game room has enough available players

**Outcome:**

1. The server launches the game and synchronizes all the user data.
2. The user performs a synchronization step to map the room and place the puzzles
3. The system shows the users the first puzzles
4. Each user is given a unique view on the puzzles they have available to them

**Use Case #6:** User saves game

**Trigger:** The user selects the save game button

**Pre-condition:** There is an active game they are playing in

**Outcome:**

1. A local copy of the progress in the game is saved on the user's side
2. All relevant data on the status of the puzzles is saved
3. This data can be used to load the game at a later date, and it is safe for users to leave the session

**Use Case #7:** User Loads game

**Trigger:** The user hits the load game button and selects the game data they want to load

**Pre-condition:** The user has created a game room, and the user has save data to load

**Outcome:**

1. The system reads the user's save data, and when the game is started, the state of the game is loaded
2. The location of the puzzles might change, but the progress of each puzzle and the total progress of the room is the same as when the game state was saved

**Use Case #8:** User deletes save data

**Trigger:** The user hits delete on their save data

**Pre-condition:** The user has save data

**Outcome:**

1. The system deletes the save data from the user's device
2. The system updates the save data list, and removes that save data from it

**Use Case #9:** User interacts with a puzzle

**Trigger:** The user selects a puzzle to attempt

**Pre-condition:** The user has started a puzzle

**Outcome:**

1. The system displays an enlarged and interactable version of the puzzle
2. The user performs an action on the puzzle
3. The system registers the action, performs any necessary puzzle state changes and displays the new puzzle state to the user

**Use Case #10:** User requests a hint

**Trigger:** The user presses the request hint button

**Pre-condition:** The user has started a puzzle

**Outcome:**

1. The system displays the predetermined hint for the associated puzzle
2. The user reads the hint and presses the button to close the hint
3. The system closes the hint display.

**Use Case #11:** User skips a puzzle

**Trigger:** The user hits the skip button

**Pre-condition:** The user is interacting with a puzzle

**Outcome:**

1. The system saves the state of the puzzle and closes the puzzle display, returning to the main scene.

**Use Case #12:** User sends a message

**Trigger:** The user presses a communication button

**Pre-condition:** More than one user are in a game room

**Outcome:**

1. The system queries the server to determine the users in the room
2. The system displays the other users in the room to the user
3. The user selects the desired recipient
4. The system displays the interface for the user to create the message
5. The user creates the message and hits the send button

6. The system uses the server to send the message to the other user's device and to create a message notification on that device. It then notifies the sender that the message has been sent.

**Use Case #13:** User reads a message

**Trigger:** The user clicks on a received message notification

**Pre-condition:** The user has received a message

**Outcome:**

1. The system removes the message notification and displays the message to the user
2. The user reads the message, then closes the message display window
3. The system returns the user to the main game scene.

## 2.2 Functional Requirements

### 2.2.1 Create Game Room

CG1. The system shall allow the user to create a game room

**Rationale:** The user must be able to group together with other users, creating the game room will allow the users to organize the group.

CG2. The system shall allow the user to enter a name for the room

**Rationale:** The users must be able to uniquely identify the game rooms to allow the other users to find the room they want to join in case they have a group they wanted to solve the puzzles with

CG3. The system shall allow the user to set the size of the room

**Rationale:** The user should be allowed to limit the size of the room in case they wanted a smaller game rather than a bigger game.

CG4. The system shall allow the user to put a password on the room

**Rationale:** The user should be able to allow or disallow any other user from joining except the ones they want in the case they are playing with friends.

CG5. The system shall redirect the user to a new screen when in a room

**Rationale:** The users in the room should have a display of the game room and the amount of players in the room etc.

### 2.2.2 Join Game Room

JG1. The system shall allow the user to join an existing game room with available capacity

**Rationale:** The user should be able to join a room to group up with other users

- JG2. The system shall allow the user to enter a password for the room if another user set a password for the room  
**Rationale:** In the case of a game room having a password, the user should be able to have a space to enter the password.
- JG3. The system shall update the game room information to add another member, and decrease the available capacity  
**Rationale:** When a user joins a game room, that information should be updated to let the users see the status of the room to inform them on which game room to join.
- JG4. The system shall redirect the user to a new screen upon successfully connecting to a room  
**Rationale:** When a user joins the room, they should get redirected to a new page where they can see the information on the current game room and wait for the host to start the game.

### 2.2.3 Edit Room Settings

- RS1. The system shall allow the user that created the game room to edit the settings of the game room  
**Rationale:** If a user input the incorrect settings when making a room, the user will have the ability to edit the settings
- RS2. The system shall populate all the changeable settings on the user's screen  
**Rationale:** The user should be able to see all the settings that are able to be changed
- RS3. The system shall allow the user to change the password of the room  
**Rationale:** The user should be able to change the password of the room to ensure security
- RS4. The system shall allow the user to change the capacity of the room  
**Rationale:** If the user was planning on having a set number of users and decides they want more or less users, they are able to change the capacity
- RS5. The system shall allow the user to remove other users from the game room  
**Rationale:** If the host user decides another user should not be in the game room, they can remove them instead of making a new room
- RS6. The system shall display the updated settings to the other users in the game room  
**Rationale:** When a setting changes the other users in the game room should be able to see the changes in case they will affect them



#### 2.2.4 Exit Room

ER1. The system shall allow the user leave the existing game room

**Rationale:** If the user does not wish to stay in a game room, the user should be able to leave

ER2. The system shall update the game room and display the one less user

**Rationale:** The system should update based on the current number of users

ER3. The server shall update all user's screen to show the new amount of users and the updated capacity

**Rationale:** The server should update all the other users on the current capacity and number of users in the game room to inform them of any possible changes

#### 2.2.5 Start Game

ST1. The system shall allow the user to start the game

**Rationale:** The users should be able to start the game to play the puzzles

ST2. The system shall load all the assets used to play the game

**Rationale:** Loading all the assets at the start of the game will stop pauses during gameplay

ST3. The system shall inform the user to complete a calibration step

**Rationale:** AR games require some calibration step to scan the surfaces in the room so objects can be placed with gravity

ST4. The system shall determine an order of the puzzles, and which users can see which parts of the puzzles

**Rationale:** In games with more than one puzzle, there needs to be an order to solve the questions. The system will determine a logical order, and assign which users can see which parts of the puzzles.

ST5. The system shall display a progress bar based on the current progress on the screen

**Rationale:** The users should have some indicator how close they are to finishing the game

ST6. The system shall reveal the available puzzles to the users and commence the game

**Rationale:** The users should be able to view the puzzles they have access to

#### 2.2.6 Save Game

SG1. The system shall allow the user to save the game

**Rationale:** Saving the game will allow users to keep the progress on the game they are playing

- SG2. The system shall save all the relevant data needed for reloading the game  
**Rationale:** If the user wants to load the game in the future they should have all the data required for continuing
- SG3. The system will pause the game during the save for the user saving it, and take a screenshot of the data available upon hitting save  
**Rationale:** All the game data will be saved based on a point in time, so pausing the user's game to save will allow the game state to remain intact
- SG4. The system shall allow other users to act during the saving process  
**Rationale:** One user saving the game should not stop the other users from continuing playing
- SG5. After saving, the system shall resume play for the saving user  
**Rationale:** The user might want to return to playing after a save
- SG6. The user shall be allowed to save as many times as they desire, as frequently as they desire.  
**Rationale:** There should be no limit on saves in case of any external factors that would effect the game
- SG7. The system shall put the saved data on the user's device  
**Rationale:** Storing the data on the user's device will reduce the stress on the database, the save files should be small in size and it likely won't inconvenience the user
- SG8. The system shall encode the data  
**Rationale:** Encoding the data can reduce the size and prevent users from manually changing the save data and potentially creating issues with the data.

### 2.2.7 Load Game

- LG1. The system shall allow the user to load an existing save file  
**Rationale:** Loading the save files is the purpose for having save files in the first place
- LG2. The system shall decode the save file and recall the progress on the saved room  
**Rationale:** Decoding the save file makes the data readable by the system
- LG3. The system shall display the saved data in the game room for all the users to see the progress  
**Rationale:** This information will allow the users in the game room to be caught upto date with the progress on the save file

### 2.2.8 Delete Save

- DS1. The system shall allow to delete the saved game data  
**Rationale:** If the user wishes to never load the data, the user can remove it

- DS2. The system shall remove the saved data from the user's device  
**Rationale:** The user's device will no longer hold onto data it won't use
- DS3. The system shall stop the user from loading the saved game in the future  
**Rationale:** Since the saved game data is gone, it should no longer be able to be loaded

### 2.2.9 Puzzle Interaction

- PI1. The system shall allow users to select a puzzle to attempt from the main scene.  
**Rationale:** The user is shown multiple puzzles on the main scene and must be able to select the one they would like to attempt.
- PI2. The system shall present the user with an enlarged version of the puzzle once they've selected it.  
**Rationale:** For full scale versions of all puzzles to be visible on the main scene, too much of the scene would be obscured. This is resolved by forcing the user to select the puzzle to see the full size puzzle.
- PI3. The system shall allow users to perform all necessary actions on the puzzle to complete the puzzle.  
**Rationale:** The user must be able to complete the puzzles in order to progress in the game.
- PI4. The system shall display the new puzzle state after a user performs an action on a puzzle  
**Rationale:** The user must be able to see the results of their actions in order to decide how to proceed with the puzzle
- PI5. The system shall update the puzzle state of group members cooperating on an updated puzzle  
**Rationale:** When multiple users are cooperating on a puzzle they must be able to see updates caused by the actions of one another.
- PI6. The system shall notify the user when they have completed a puzzle  
**Rationale:** Users should know when they've completed a puzzle so they know they can move on. A notification at the end of a puzzle can also make completing it more satisfying for the user.
- PI7. The system shall update the overall room progress when a puzzle is completed  
**Rationale:** The overall room progress must be continuously updated as puzzles are completed so the game can detect when all the puzzles have been completed and the users have accomplished their goal.

### 2.2.10 Hint Request

HR1. The system shall have at least one hint for each puzzle.

**Rationale:** If a user gets stuck on a puzzle they should have an alternative to giving up on the puzzle.

HR2. The system shall have a means of requesting a hint for the puzzle to be displayed.

**Rationale:** The player must be able to request the hint for the puzzle.

HR3. The system shall allow the user to close the hint display

**Rationale:** The user should not have to keep the hint on the screen, as it may obstruct other functions.

### 2.2.11 Skip Puzzle

SP1. The system shall allow the user to skip a puzzle and return to the main scene.

**Rationale:** The user should not have to complete a puzzle to attempt a different one. They should be able to leave a puzzle and return to it later.

SP2. The system shall save the users progress on a skipped puzzle.

**Rationale:** If a user skips a puzzle they should be able to return to it without having to start over.

SP3. The system shall notify any cooperators on the skipped puzzle that it has been skipped

**Rationale:** One users work on a puzzle will typically depend on another users work on the puzzle. They must be notified the puzzle has been skipped so they are not waiting on an action from the other user.

### 2.2.12 Send Message

SM1. The system shall allow the user to view all the users in the current game room.

**Rationale:** For the communication system the user must be able to see all the users in the game room to choose who to contact.

SM2. The system shall contain an interface that allows the user to compose a message

**Rationale:** The users must be able to create their own custom messages to allow for optimal communication in puzzles and to create the most realistic social experience.

SM3. The system shall allow the user to send a message to any other user in the game room

**Rationale:** The primary goal of the game is to allow users to have social interaction from a remote setting. All users in the game must be able to interact with one another to complete this goal.

SM4. The system shall allow the user to cancel a message before sending it.

**Rationale:** If a user decides they do not want to send a message they must be able to return to the game without having to send something.

SM5. The system shall notify the user once the message has been sent.

**Rationale:** It is important that the user receive feedback that the message has been sent so there is no uncertainty or confusion for the user.

### 2.2.13 Read Message

RM1. The system shall notify the user when a new message is received.

**Rationale:** Communication is core to the game. If a user doesn't know when they receive a message good communication will not be maintained.

RM2. The system shall allow the user to view messages.

**Rationale:** Users must be able to see what other users are trying to communicate to them for social purposes and to complete puzzles.

RM3. The system shall allow users to close a message.

**Rationale:** Users must be able to return to the mains scene after viewing a message to continue game play.

## 2.3 Formal Specification

$(\exists a \in \text{users} \ \& \ \exists r \in \text{room} \ \& \ c = r.\text{capacity} \mid \sum a \in r \leq c) \Rightarrow \text{true}$

The users that exist in a room shall be within the capacity of the room

# 3 Non-functional Requirements

## 3.1 Look and Feel Requirements

LF1. The system shall adjust and scale to fit the physical screen size

LF2. The system shall have fonts and colours that will allow users to easily read the text

## 3.2 Usability and Humanity Requirements

UH1. The system shall be accessible by any supported mobile device

UH2. The system shall notify the user if there is no network, or they get disconnected

UH3. The system shall be able to be navigated by any user

## 3.3 Performance Requirements

PR1. The system shall respond to user interaction within 100ms

PR2. The system shall be available for users at any time

### **3.4 Operational and Environmental Requirements**

OE1. The system shall be available on all android and iOS platforms

### **3.5 Maintainability and Support Requirements**

MS1. The system shall be well documented and kept up to date, and have updated design documents.

MS2. The source code shall be well commented for future developers, including author, edit dates documented.

### **3.6 Security Requirements**

SR1. The system shall keep user data private

SR2. The users will only be allowed to see limited data. Unnecessary data will not be displayed to the user.

### **3.7 Cultural Requirements**

CR1 The system shall not use any offensive images, text, or sound that could offend any religious or culture groups.

CR2 Any user that leaves an offensive message will have their account banned from the system

CR3 The system will use Canadian English

### **3.8 Legal Requirements**

LR1 The system shall not infringe on the rights of any person(s), and shall not use any assets that infringe on copyright claims. When using open source resources, the developers shall give appropriate credit.

### **3.9 Health and Safety Requirements**

HS1 The system shall give a warning to the user to be aware of their surroundings while using the system, and to not bump into any objects or obstacles in their path

## 4 Traceability Matrix

## 5 Project Issues

The following section will go over the current issues for the project. It will describe open issues, current off-the-shelf solutions, new problems, and tasks that need to be worked on.

### 5.1 Open Issues

There are no open issues for this project.

### 5.2 Off-the-Shelf Solutions

#### 5.2.1 Ready-Made Products

There are several already made products that have similar aspects to the project however they are missing some key aspects. Examples include ARctic Escape and We Were Here. Similarly to this project, ARctic Escape is an AR game that encourages social interaction. Users must work together to solve puzzles and complete the game. The main difference between this project and ARctic Escape is that in ARctic Escape, users must be in the same area/room to play the game, however, Mac-AR is based around users being able to interact and play the game remotely. Continuing on, We Were Here is a two player game in which users must work together to solve difficult puzzles. The game can be played remotely between the two players and there is an in-game communication system that allows the users to talk to each other. We Were Here, however, does not have any AR component to it, which is one of the main aspects of this project.

#### 5.2.2 Reusable Components

Plenty of libraries and other frameworks that already exist can be used in the project. AR Foundation is one such framework that can be used through Unity. It supports features such as device tracking and 2D image tracking which can be used for tracking the movement of users phones while playing the game. Additionally, there are the Zed-Rig plugins that Unity supports. These can help provide functionality for AR object placement as well as spatial mapping. Continuing on, ARctic Escape which was mentioned in the Ready-Made Products, was able to have spatial mapping through a Snapchat Lens. This allowed for users to record their surroundings and have the room they were in mapped and able to be used in the AR environment.

#### 5.2.3 Products That Can Be Copied

Currently, the only known product that is similar to this project and can be copied without copyright issues is ARctic Escape, however, this game is implemented much differently than

the plan for this project. ARctic Escape was implemented as a Snapchat Lens, however, the goal for this project is to create a standalone application through the Unity framework.

## **5.3 New Problems**

The following section will describe potential problems of the application's creation. It will go over the effects on the current environment and installed system, as well as user problems and other follow-up problems.

### **5.3.1 Effects on the Current Environment**

(not sure)

### **5.3.2 Effects on the Installed Systems**

The application will not have any effect on the mobile devices it is installed on. It will act as a standalone application on the device and should not affect any other parts of the device.

### **5.3.3 Potential User Problems**

A potential user problem may be the difficulty in solving in-game puzzles/tasks. If tasks are too difficult to complete, it may result in users wanting to quit the game, and therefore resulting in a decreased level of social interaction. Additionally, if the tasks are too easy to solve, this may also reduce the level of social interaction as users will not need to communicate as much. This may also result in users not enjoying the gameplay.

### **5.3.4 Limitations in the Anticipated Implementation Environment That May Inhibit the New Product**

It is assumed that users will be able to use the application in an area with internet connection, however, it is not unusual that people may have Internet issues, which may cause them to lose connection to the Internet. This results in a problem as without Internet connection users will disconnect from the host server, causing them to not be able to play the game or interact with their partner.

### **5.3.5 Follow-Up Problems**

Other problems include if libraries used in the creation of the application get updated while the project is ongoing. This may result in certain in-game functionality not behaving as intended. Another problem that may arise is if users are able to find ways to solve tasks by themselves, without the need for interaction with others. The main purpose of the application is to have social interaction between users, and this would result in users not needing to communicate with each other.



## 5.4 Tasks

The following section will go over the planning of the project.

### 5.4.1 Project Planning

The project schedule will follow the deadline for the deliverables outlined in the SFWRENG 4G06 course outline.

Revision #	Deliverable	Due Date
Revision 0	Hazard Analysis	October 20, 2023
	V&V Plan	November 3, 2023
	PoC Demo	November 13-24, 2023
	Design Document	January 17, 2024
	Revision 0 Demo	February 5-16, 2024
	V&V Demo	March 6, 2024
Revision 1	Final Demo	March 18-29, 2024
	EXPO Demo	April, 2024
	Final Documentation	April 4, 2024

### 5.4.2 Planning of the Development Phases

The development of the project will be split into three phases:

- Creation of a proof of concept and documentation
- Initial development of the application with consideration to PoC
- Refinement of application and documentation after user feedback

The creation of the proof of concept demonstration will aid in overcoming the main challenges of the system. It will demonstrate the ability for multiple users to interact in AR, and provide the documentation for the future application to be developed.

The initial development of the application will take the design of the PoC and improve on the concepts. All the components of the application will be working in this version, and the stakeholders will be able to see the implementation and provide feedback.

The refinement will take all the user feedback and improve on the initial design. There is no expectation for major features to be developed at this stage, just improvements on the existing system and the addition of potential stretch goals.

## 5.5 Migration to the New Product

The following section will go everything required for migration to the application.

### **5.5.1 Requirements for Migration to the New Product**

There are no requirements for migration for the application.

### **5.5.2 Data That Has to Be Modified or Translated for the New System**

No data needs to be modified or translated for the application.

## **5.6 Risks**

The main risks of the project are:

- Users losing awareness of their surroundings and potentially getting injured
- Server overload could cause lag in gameplay, resulting in an unpleasant user experience
- Failure to meet deadlines will cause setbacks for the project. In the event of this, some requirements might not be met, and documents will have to be modified to reflect this.

## **5.7 Costs**

There is no financial costs associated with the development of the application, and there is no cost for users downloading and using the application. Furthermore, the time to development time is estimated to be six months, with work being divided amongst four developers.

## **5.8 User Documentation and Training**

The following section will go over any training and documentation that is needed for users, as well as waiting room ideas and ideas for implementation

### **5.8.1 User Documentation Requirements**

The users will have a user help document available to them within the application in the form of a gameplay menu. It will contain information on how to use the application, as well as any instructions regarding interacting with puzzles/tasks.

### **5.8.2 Training Requirements**

The users will have a "tutorial" when they first start the game that will teach them how to use the application. It will show the users how to use AR technology, and give them a simple puzzle to solve.

## **5.9 Waiting Room**

There are currently no requirements or functionalities that are not being included in the initial release. Future ones may be added later on.

## 5.10 Ideas for Solutions

During the requirements and idea generation phases, additional ideas were generated on how to implement the solution.

- Form
  - Since the application will be developed using Unity framework, the idea of using Unity based libraries and plugins for AR framework such as AR Foundation and Zed-Rig.

## 6 Appendix

This section will be used to capture miscellaneous information that couldn't easily be fit into the rest of the document.

### 6.1 Reflection

A new and unique set of skills will be required for the completion of this project. It is hoped that the entire team will have an exposure to everything listed below, but the team members in charge of fostering the skill and development of the sections that require it are also listed.

**RS1 Mobile App Development:** Developing a mobile application presents unique challenges that aren't present when developing a standard computer application. Dealing with limited memory and processing power, navigating the licensing and system requirements for different devices, and fully utilizing the tools embedded in phones are all skills that will need to be developed over the course of the project.

– **Main Team Members:** Sam, Matthew

**RS2 Augmented Reality:** Integrating AR into an application requires an understanding of how to get a piece of software to integrate with the real world. Team members will need to learn how to make the tool process the readings from various video and location instruments, calculate how to place objects into a 3D space, and give the user the tools to manipulate the world through their augmented view.

– **Main Team Members:** Kieran, Ethan, Sam, Matthew

**RS3 Communication Services:** Allowing different devices, running on different types of software to communicate with each other over long distances is a vital part of capturing the collaborative aspect of the project. The team will need to investigate how to accomplish this networking.

– **Main Team Members:** Kieran, Matthew

**RS4 Location Services:** Tracking the location of the user and integrating nearby landmarks in real time adds additional depth to the tool. A vital skill will be getting the tool to interface with existing location services.

– **Main Team Members:** Ethan, Sam

**RS5 Game Design:** Getting a large and diverse user base is essential for the collaborative aspect of the tool. Researching what makes a game fun to play and striking a balance between the difficulty and complexity of the puzzles is what will drive this engagement.

– **Main Team Members:** Kieran, Ethan

## 6.2 Symbolic Parameters

There are currently no symbolic parameters that need to be recorded for this document. Any future symbols will be recorded in the following table:

Symbol	Value	Description

## References

James Robertson and Suzanne Robertson. *Volere Requirements Specification Template*. Atlantic Systems Guild Limited, 16 edition, 2012.