

# Software requirements specification for Gamification of Evacuation Drills

Nov 28, 2022

## Preface

This document outlines the requirements of the project for both the project owner and the development team.

Document version number	Changes and rationale
1	-
2	Added user requirement 10 and system requirements 2.1 and 10.1
3	Added activity diagram

# Introduction

The system will be a phone app which will use AR to track the player as they move through an interior space in a simulated evacuation drill. The player would have to avoid obstacles on the screen and respond to other changes in the AR environment such as smoke. The goal of the player is to get to a safe spot, at which they will be notified and the game will end. All location data for the player's game session will be stored on a server for later analysis.

The purpose of this app is to be able to map flow patterns in real time during egress. The app would enable experiments to be made to validate numerical egress models with high precision. The app would also provide feedback to the player, such as providing information on smoke spread, making the drill more realistic and resulting in better egress flow statistics.

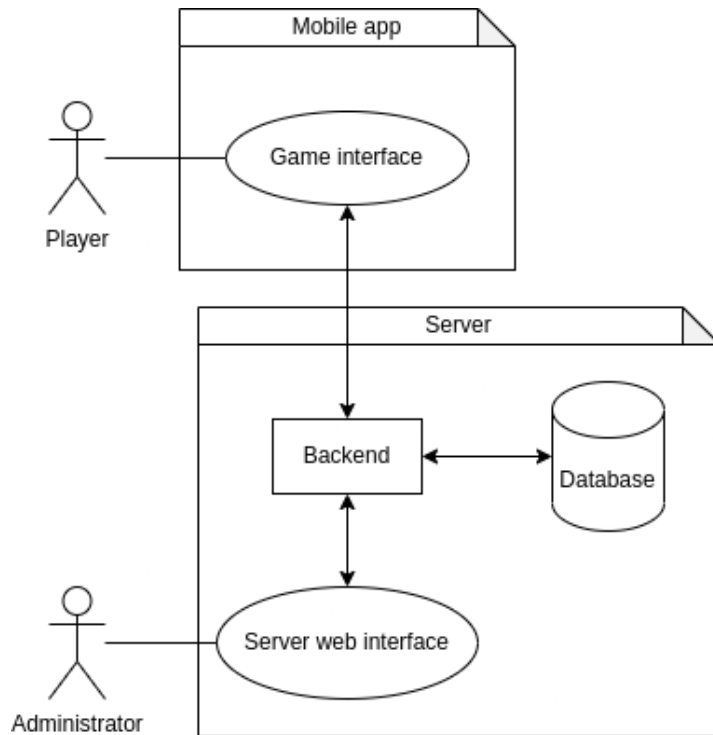
## Glossary

Augmented reality (AR)	Virtual elements superimposed on a view of the real world.
Player	The user of the phone app.
Administrator	The person managing the server side of the system.
Server	Remote computer used to coordinate data between users.
Session	The place in cyberspace where a single run of the game takes place, from when the first player joins the game to when the last player completes the game.

## User requirements

1. The application should be a phone app which will use AR to track the player as they move through an interior space.
2. This location data should be available later on, and be timestamped.
3. The app will be developed using the Unity game engine.
4. The application only needs to consider one floor.
5. There should be an alarm sound, becoming louder closer to the alarm source.
6. The application will display obstacles overlaid in AR, which the player will have to avoid.
7. The application should track when the user arrives at the safe zone.
  - a. The app should inform the player when this happens.
  - b. The player will win the game when this happens.
8. No map of the building should be included during the drill.
  - a. Additionally, no overlay of the environment model is needed.
9. The application does not need to display fires in AR, just the alarm and smoke.
10. Dynamic spawn for the user, so they can start from anywhere in the building.

# System architecture



The database component will use a pre-existing system, and the web interface and backend will be developed using existing web frameworks.

## System requirements

**1.1** The application will have 3 main scenes:

- a. A start menu screen with button options allowing the player to change options, view a help message, and proceed to the game (b).
- b. The game screen. The game will start based on parameters set by an administrator. This screen may have additional overlays to control settings or to end the session.
- c. The win screen, see section 7 for more details.

**1.2** GitHub will be used for version control and prototype distribution.

**2.1** The location data will be stored using the third-party site JSONBin.io during development and testing, and they will be stored on our backend later when it is ready for production.

**3.1** The location data will be sent to a server using a web request containing a unique ID for the player, their location, a time stamp, and a session password. This data may be batched to be sent at a later time to limit the amount of requests sent to the server. It will be developed in a separate GitHub repository.

**3.2** The database server is developed as a separate application from the main game client, and may use pre-existing frameworks.

3.3 The server may also allow the administrator to change game parameters for all players connecting to the session.

3.4 The server may allow administrators to visualize the collected data.

3.5 The server will be packaged in such a way as to make set up easy for the client.

3.7 The administrator will be able to set a password for the session. The player can enter this password in the game app to send its location data to the server.

4.1 No implementation of height detection will be made, but we will use 3D vectors for location data with the height component set to 0 so as not to limit future developers.

5.1 The sound coming from the alarm will have its volume attenuated by a falloff function based on the distance, making sure that the max and minimum volumes possible are taken into consideration.

7.1 The phone may vibrate and display a “You win!” message to the player, and end the location recording session.

7.2 Additional information such as points may be displayed.

7.3 The player will have a button option to go back to the main menu.

9.1 The smoke will come from a predetermined point specified in the game parameters set by the administrator on the server.

10.1 Before starting the drill the user gets to use a minimap to add their starting position and looking direction, so the obstacles can be placed correctly in relation to the environment.

## System evolution

The system is developed using a few fundamental assumptions:

- The system is assumed to remain small-scale and as such not need a scalable database/web app distribution platform. One session is meant to run on one server only.
- Height/floor detection will not be implemented, but the datatypes used will support future implementations.

As AR is developed and used more and more, people may not use their phones for AR content anymore and instead use more convenient technologies such as glasses. This is far enough in the future though that it is not considered a development concern for this system.

# Appendices

How the data is laid out in the database:

Database		
Session 1		
Game Parameters	Players	
<div><div>Session password</div><div>Game environment</div><div>Object spawn points</div></div>	Player 1	Player 2
	Timestamp: Location	Timestamp: Location
	Timestamp: Location	Timestamp: Location
	Timestamp: Location	Timestamp: Location
	Timestamp: Location	Timestamp: Location
Session 2		
Game Parameters	Players	
<div><div>Session password</div><div>Game environment</div><div>Object spawn points</div></div>	Player 1	Player 2
	Timestamp: Location	Timestamp: Location
	Timestamp: Location	Timestamp: Location
	Timestamp: Location	Timestamp: Location
	Timestamp: Location	Timestamp: Location

# Activity diagram

