

# VR DEVOPS GAME

## FUNCTIONAL SPECIFICATION



Adam Strahan  
C00216499

## Contents

Table of Figures .....	2
Table of Tables .....	2
Introduction .....	3
Project Description.....	4
Users .....	4
Use Case Diagram .....	5
Use Cases .....	6
FURPS+ .....	10
Functionality .....	10
Core Functionality.....	10
Secondary Functionality.....	10
Usability .....	10
Reliability.....	10
Performance .....	11
Supportability.....	11
'+' .....	11
Metrics .....	11
Testing.....	11
Appendix .....	12
Plagiarism Declaration .....	12

## Table of Figures

Figure 1. Player Use Case Diagram .....	5
---	---

## Table of Tables

Table 1. Create Account Use Case .....	6
Table 2. Update Account Details Use Case .....	6
Table 3. Delete Account Use Case .....	7
Table 4. Log In Use Case.....	7
Table 5. Log Out Use Case.....	7
Table 6. Start Game Use Case .....	8
Table 7. Join Game Use Case .....	8
Table 8. Pick Up Block Use Case.....	8
Table 9. Place Block Use Case .....	9
Table 10. Give Feedback Use Case.....	9

## Introduction

For this project I am creating a virtual reality game that teaches users about DevOps. This functional specification will outline the more technical aspects of the project, including:

- A description of the project.
- Details about the users.
- A use case diagram.
- Use cases.
- FURPS+.
- Metrics by which we can judge the success of the project.
- Testing for the project.

## Project Description

To play the game a player first has to create an account in Engage and log in. They will enter their name, password and whether they are a developer or in operations. Once a player is logged in, they can choose to start a new game, join a game, or fill in a survey to give feedback on the game.

If a player chooses to start a new game, they will receive a 6-digit code for the game which other players must enter in order to join the game. Once all the players are in the lobby they are split into 2 teams: Developers and Operations. Operations is then sub-divided into Pre-production and Production.

The game consists of 4 sprints for each team, with a maximum time limit of 3 minutes per sprint. During the first sprint, the developer team must build a tower using blocks. The dev team then have to deploy the tower to pre-production within their 3 minutes. The ops team must refuse any non-documented tower from this sprint. When it is the operations teams turn for the sprint, they have to rebuild the dev teams tower in production.

For the second sprint, the teams cannot communicate with each other. The dev team just have to document their tower delivery like the first sprint but with no new objectives. The operations team have to start with a T-shaped base though, meaning a horizontal block balancing on a vertical one. This will make the dev tower almost impossible to reproduce, showing the importance of communication between teams.

For the third sprint, the teams can see each other's towers and communicate with each other. The dev team take back the last tower and rebuild it with any new features and the ops team have to rebuild that tower in pre-production and production.

The final sprint runs very similar to the third. It gives the teams a chance to continue communicating and speed up their delivery time. Each team is then awarded points based on objectives given by the client. The goal of the game is to get as many points as possible.

## Users

The users of the game are referred to in this document as players. The game is to be implemented into the Engage VR platform from Immersive VR. It is then used by third level institutions to teach students DevOps. This means our players are third level students. The players will have to make an account which contains their name, email, password and whether they are a part of the developer team or operations team. Once they have created an account and logged in, they have access to the game. They also have the option of deleting their account or editing their account details. The game is a simulation of a DevOps work environment and the player should come away from the game with a deeper understanding of DevOps. As the game is for third level students as opposed to professional developers, the level of understanding of DevOps after playing the game is not expected to be of a professional standard, just to the standard of a college course in DevOps.

## Use Case Diagram

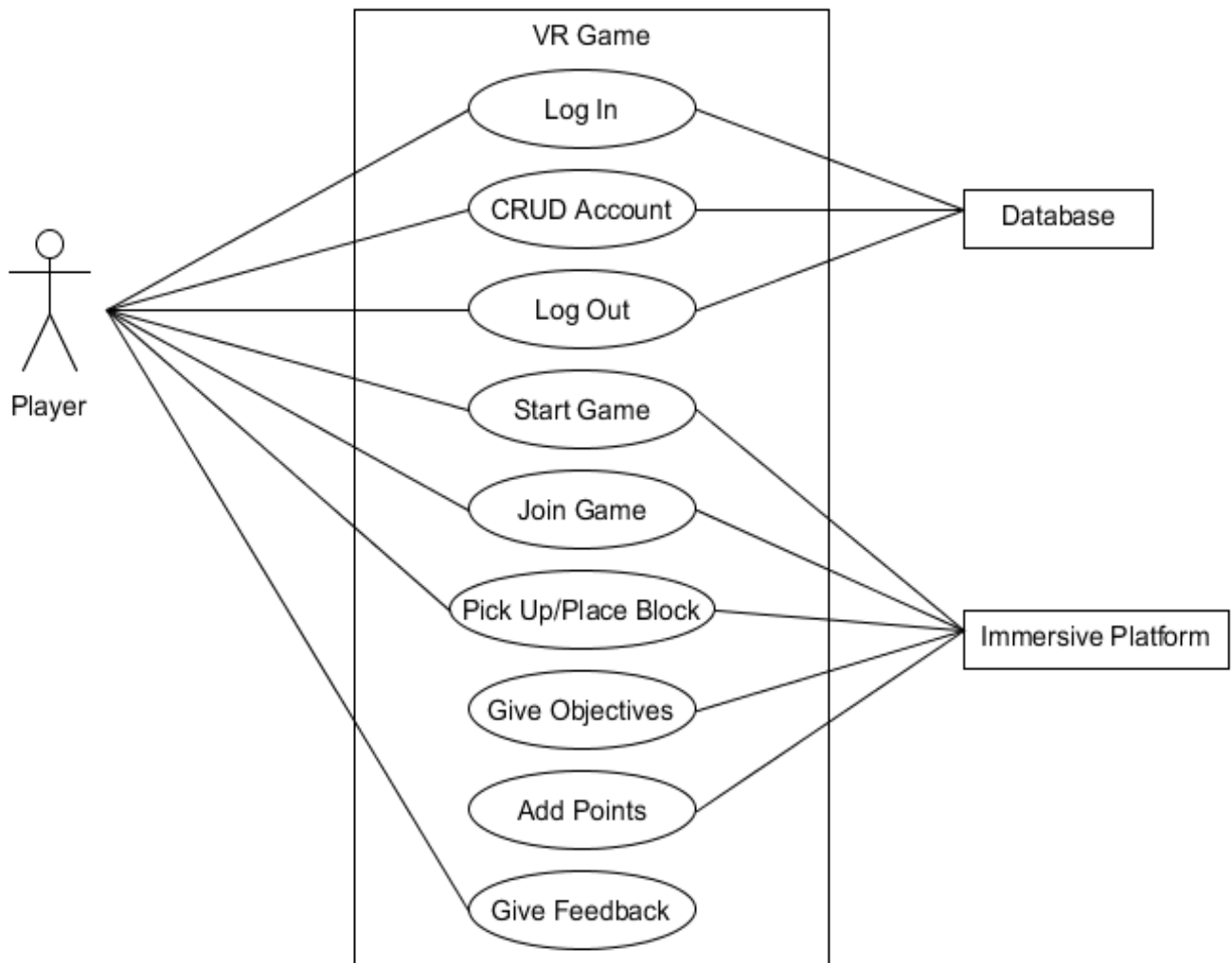


Figure 1. Player Use Case Diagram

## Use Cases

Name	Create Account
Actors	Player, Database
Preconditions	Player has downloaded and opened the game.
Activity	<ol style="list-style-type: none"> <li>1) Player chooses 'Create Account' when prompted to log in or create account.</li> <li>2) Player enters a username.</li> <li>3) Player enters and confirms a password.</li> <li>4) Player enters an email address.</li> <li>5) Player chooses whether they are a developer or operations.</li> <li>6) Player confirms their details.</li> <li>7) Account details are saved to the database.</li> </ol>
Postconditions	Player has created an account and it is saved in the database.
Alternatives	<ol style="list-style-type: none"> <li>1) Player enters an invalid email address               <ol style="list-style-type: none"> <li>1a) Player is prompted to enter a valid email</li> </ol> </li> <li>2) The passwords entered do not match               <ol style="list-style-type: none"> <li>2a) Player is prompted to enter matching passwords</li> </ol> </li> </ol>

Table 1. Create Account Use Case

Name	Update Account Details
Actors	Player, Database
Preconditions	Player is logged in.
Activity	<ol style="list-style-type: none"> <li>1) Player clicks 'Update Account Details'.</li> <li>2) Player enters their old password and confirms.</li> <li>3) Player enters a new username.</li> <li>4) Player enters a new email.</li> <li>5) Player enters and confirms a new password.</li> <li>6) Player chooses whether they are a developer or operations.</li> <li>7) Player confirms the new details.</li> <li>8) New details replace the old details in the database.</li> </ol>
Postconditions	Player has updated their account details and saved them to the database.
Alternatives	<ol style="list-style-type: none"> <li>1) The entered password is incorrect               <ol style="list-style-type: none"> <li>1a) Player is prompted that the password is incorrect and asked to re-enter it.</li> </ol> </li> </ol>

Table 2. Update Account Details Use Case

Name	Delete Account
Actors	Player, Database
Preconditions	Player is logged in.
Activity	<ol style="list-style-type: none"> <li>1) Player clicks 'Delete Account'.</li> <li>2) Player enters their password.</li> <li>3) Player confirms they want to delete their account.</li> <li>4) Account is removed from the database.</li> </ol>
Postconditions	Player is logged out and the account is deleted.
Alternatives	<ol style="list-style-type: none"> <li>1) The password is incorrect. <ol style="list-style-type: none"> <li>1a) Player is prompted that the password is incorrect and asked to re-enter it.</li> </ol> </li> </ol>

Table 3. Delete Account Use Case

Name	Log In
Actors	Player, Database
Preconditions	Player has the game open.
Activity	<ol style="list-style-type: none"> <li>1) Player chooses 'Log In' when prompted to log in or create and account.</li> <li>2) Player enters their username.</li> <li>3) Player enters their password.</li> <li>4) Player confirms their details.</li> </ol>
Postconditions	Player is logged in.
Alternatives	<ol style="list-style-type: none"> <li>1) Player enters an invalid username. <ol style="list-style-type: none"> <li>1a) Player is prompted to enter a valid username.</li> </ol> </li> <li>2) The password entered is incorrect. <ol style="list-style-type: none"> <li>2a) Player is prompted that the password is incorrect.</li> </ol> </li> </ol>

Table 4. Log In Use Case

Name	Log Out
Actors	Player, Database
Preconditions	Player is logged in.
Activity	<ol style="list-style-type: none"> <li>1) Player clicks 'Log Out'.</li> <li>2) Player is prompted to confirm log out.</li> <li>3) Player confirms.</li> </ol>
Postconditions	Player is logged out.
Alternatives	<ol style="list-style-type: none"> <li>1) Player cancels the log out. <ol style="list-style-type: none"> <li>1a) Player clicks 'Cancel' when prompted to confirm log out.</li> </ol> </li> </ol>

Table 5. Log Out Use Case



Name	Start Game
Actors	Player, Immersive Platform
Preconditions	Player is logged in.
Activity	<ol style="list-style-type: none"> <li>1) Player clicks 'Start Game'.</li> <li>2) Player receives a 6-digit code for the game to be used by others to join the game.</li> <li>3) Player clicks 'Start'.</li> </ol>
Postconditions	Player has set up a game.
Alternatives	<ol style="list-style-type: none"> <li>1) Player closes the game. <ol style="list-style-type: none"> <li>1a) Player clicks 'Leave Game'.</li> </ol> </li> </ol>

Table 6. Start Game Use Case

Name	Join Game
Actors	Player, Immersive Platform
Preconditions	Player is logged in.
Activity	<ol style="list-style-type: none"> <li>1) Player clicks 'Join Game'.</li> <li>2) Player enters a 6-digit code and confirms.</li> </ol>
Postconditions	Player has joined a game.
Alternatives	<ol style="list-style-type: none"> <li>1) Player enters an invalid code. <ol style="list-style-type: none"> <li>1a) Player is prompted to enter another code.</li> </ol> </li> </ol>

Table 7. Join Game Use Case

Name	Pick Up Block
Actors	Player, Immersive Platform
Preconditions	Player is in a game.
Activity	<ol style="list-style-type: none"> <li>1) Player moves their VR controller over a block.</li> <li>2) Player clicks the interact button</li> </ol>
Postconditions	Player has picked up a block.
Alternatives	<ol style="list-style-type: none"> <li>1) Player has not moved their controller over a block. <ol style="list-style-type: none"> <li>1a) Player does not pick up anything when the interact button is pressed.</li> </ol> </li> </ol>

Table 8. Pick Up Block Use Case

Name	Place Block
Actors	Player, Immersive Platform
Preconditions	Player has picked up a block.
Activity	<ol style="list-style-type: none"> <li>1) Player moves their VR controller over a space.</li> <li>2) Player clicks the interact button.</li> </ol>
Postconditions	Player has placed the block.
Alternatives	<ol style="list-style-type: none"> <li>1) The controller is out of the play area. <ol style="list-style-type: none"> <li>1a) The block is not placed when the interact button is pressed.</li> </ol> </li> </ol>

Table 9. Place Block Use Case

Name	Give Feedback
Actors	Player
Preconditions	Player is logged in.
Activity	<ol style="list-style-type: none"> <li>1) Player clicks 'Give Feedback'.</li> <li>2) Player answers multiple choice questions about DevOps.</li> <li>3) Player gives a rating of their experience.</li> <li>4) Player types in comments about the game.</li> <li>5) Player submits the form.</li> </ol>
Postconditions	Player has submitted feedback.
Alternatives	<ol style="list-style-type: none"> <li>1) Player backs out of giving feedback. <ol style="list-style-type: none"> <li>1a) Player clicks 'Cancel' while in the feedback form.</li> </ol> </li> </ol>

Table 10. Give Feedback Use Case

## FURPS+

FURPS is an acronym for Functionality, Usability, Reliability, Performance, Supportability and is used to outline the functional requirements of a project. The '+' was added to outline any additional considerations for non-functional requirements such as design, implementation, interface, or physical constraints. FURPS+ is used widely in the Software Development Industry as it may be used easily to outline many of the requirements quickly.

### Functionality

#### Core Functionality

The VR DevOps game has the following core functionality:

##### *Start a game*

A player must be able to create a game in order for the game to function.

##### *Pick Up/Place a block*

Picking up and placing blocks is what the player must do in order to progress through the game, therefore it is core functionality and must be implemented correctly.

##### *Join a game*

For the multiplayer aspect of the game to function, players need to be able to join a game. If the game isn't multiplayer, then it can't effectively teach DevOps methodologies.

##### *Create/Delete/Edit an account*

Players need to have accounts to differentiate themselves in a lobby.

#### Secondary Functionality

The VR DevOps game has the following secondary functionality:

##### *Give Feedback*

This is secondary functionality as it isn't essential for the game to function, but it would help to maintain and improve it over time.

### Usability

As virtual reality is still a new technology to most people, it is vital to get the user experience and usability right. Many people suffer from motion sickness while using VR when they haven't gotten accustomed to it. To minimise the risk of motion sickness, the player should have a range of movement options available to move around the environment. Teleporting and snap turning (turning the field of view a set number of degrees when a button is pressed) should be available for players to use. Free movement should be available for more experienced players who don't suffer from motion sickness.

For ease of use, the player should be able to log in within 10 seconds, 95% of the time. Logging out should take less than 5 seconds, 99% of the time. Setting up a game should be possible within 30 seconds, 90% of the time.

### Reliability

As the game will be available within Immersive VR's Engage platform, the game should be up and running as long as Engage is up, 95% of the time. This allows for downtime for updates or unforeseen issues.

## Performance

By using a cloud-based database and MySQL, the game should be able to retrieve and send data quickly. It should take no more than 20 seconds to send data to the database or retrieve data from the database 90% of the time. This gives us some leeway should there be any server issues.

## Supportability

With the Engage VR platform, the game will be supported on all available VR headsets that support Engage.

'+

## Security

As the game has personal accounts, the game needs to be secure enough to comply with GDPR guidelines. The accounts are password protected and the passwords should be encrypted before being stored in the database in case of a cyber-attack.

## Metrics

Metrics in the context of this project describes how the success of the application may be measured. The success of the game can be measured by the following:

- It should run on VR headsets (Oculus Quest 2).
- The player should be able to create an account.
- The player should be able to log in and out of their account.
- The game should be able to work with the VR controllers to allow players to interact with the blocks.
- The player should be able to submit feedback about the game.
- The game should be implemented into the Engage VR platform.
- Multiple players should be able to connect to the same game.

## Testing

The testing for the game will be done as the game is being built. I will be testing the functionality as I build it. There are also other developers working on separate functionality for a larger simulation within Engage that will be testing it. I would also like to get neutral play testers in to test the game as it gets closer to completion in order to get a few unbiased views on the software.

## Appendix

### Plagiarism Declaration

#### Projects & Plagiarism

All projects are Individual projects. The project is expected to be all your own work. Under certain circumstances, you may be able to incorporate material from other sources. This "incorporation" needs to be discussed with, and agreed upon by, lab supervisor and all such material must be referenced appropriately.

- Plagiarism is defined as " ... *presenting someone else's work as if it were your own, whether you mean to or not. 'Someone else's work' means anything that is not your own idea, even if it is presented in your own style. It includes material from books, journals or any other printed source, the work of other students or staff, information from the Internet, software programs and other electronic material, designs and ideas. It also includes the organization or structuring of any such material ...* " [taken from: Victoria University of Wellington, New Zealand, <http://www.victoria.ac.nz/home/glossary.aspx#p>].
- What this means is that it is wrong to copy another's work or download material for incorporation into your project from the Internet (without the express permission of your supervisor). Even if you take someone else's algorithm/code and rewrite it, this must be documented in your program to state where you got the original algorithm from. Failure to do so is considered plagiarism. If you are in any doubt, always ask your lab supervisor.
- Plagiarism is a **serious academic offence** (equivalent to cheating on an exam) and the consequences of being found guilty of plagiarism are severe.
- Students can use the internet to research information on their projects, they can ask lab supervisors for help with their project and they can also discuss problems with their fellow class mates.
- Students should be able to explain fully their code and may be asked to do so by one of your lab supervisors.

Work submitted for assessment which does not include this declaration will not be assessed.

#### DECLARATION

\*I declare that all material in this submission e.g. thesis/essay/project/assignment is entirely my/our own work except where duly acknowledged.

\*I have cited the sources of all quotations, paraphrases, summaries of information, tables, diagrams or other material; including software and other electronic media in which intellectual property rights may reside.

\*I have provided a complete bibliography of all works and sources used in the preparation of this submission.

\*I understand that failure to comply with the Institute's regulations governing plagiarism constitutes a serious offence.

Student Name: (Printed) Adam Strahan

Student Number(s): C00216499

Signature(s): Adam Strahan

Date: ~~21/11/20~~ 12/11/20

-----

#### Please note:

The Institute regulations on plagiarism are set out in Section 10 of Examination and Assessment Regulations published each year in the Student Handbook.