

1. Home .....	2
1.1 Requirements .....	4
1.1.1 Background .....	5
1.1.2 Motivational modelling .....	6
1.1.3 Personas .....	7
1.1.4 User Stories & Acceptance Criteria .....	11
1.2 General Process Documentation .....	14
1.2.1 Communication Plan .....	15
1.2.2 Roles and Responsibilities .....	16
1.3 Research .....	17
1.3.1 Back-End Tools .....	18
1.3.2 Database .....	19
1.3.3 Dev-Ops Tools .....	20
1.3.4 Front-End Tools .....	22
1.3.5 Game Formats .....	23
1.3.6 Platform .....	24
1.3.7 Proposed Gameplay Designs .....	25
1.3.8 Related Games .....	29
1.3.9 Testing Frameworks .....	30
1.4 Game Design .....	31
1.4.1 Gameplay .....	32
1.4.2 Characters .....	33
1.4.2.1 Aeronautical engineer .....	34
1.4.2.2 Airline pilot .....	35
1.4.2.3 Boeing executive .....	36
1.4.2.4 Federal Aviation Administration (FAA) official .....	37
1.4.2.5 Software developer .....	38
1.4.3 Plot .....	39
1.4.4 Decision Tree .....	40
1.4.4.1 Decision Tree (version 1) .....	41
1.4.4.2 Decision Tree (version 2) .....	46
1.4.4.3 Decision Tree (version 3) .....	47
1.4.4.4 Decision Tree (version 4) .....	48
1.5 Prototypes .....	49
1.5.1 Digital prototype .....	50
1.5.2 Paper prototype .....	51
1.6 Architecture .....	54
1.6.1 Activity Diagram .....	55
1.6.2 API Specification .....	60
1.6.3 Architecture Design .....	66
1.6.4 Class Diagram .....	68
1.6.5 Component Diagram .....	70
1.6.6 Deployment Diagram .....	73
1.6.7 Domain Model .....	74
1.6.8 ER Diagram (Database) .....	75
1.6.9 High Level Diagram .....	78
1.6.10 Sequence Diagram .....	79
1.6.11 Technology .....	82
1.7 Project Plans .....	83
1.7.1 Sprint1 Report(Planning) .....	84
1.7.2 Sprint2 Report .....	87
1.7.3 Sprint3 Report .....	92
1.8 Quality Assurance .....	95
1.8.1 Coding Standards .....	96
1.8.2 Review process .....	97
1.9 Version Control .....	99
1.9.1 Git Workflow .....	100
1.10 Testing .....	101
1.10.1 Back-End Unit Testing/Integration Testing .....	102
1.10.2 Front-End Unit Testing .....	104
1.10.3 Acceptance Testing .....	107
1.10.4 End-to-End Testing .....	108
1.10.5 Bug Report .....	109
1.10.6 Testing Approach .....	110

# Home

## Ethics Role Play

### COMP90082 Software Project (2022)

**Client:** Dr Simon Coghlan <[simon.coghlan@unimelb.edu.au](mailto:simon.coghlan@unimelb.edu.au)>

**Supervisor:** Mauro Mello Jr <[mauro.mellojr@unimelb.edu.au](mailto:mauro.mellojr@unimelb.edu.au)>

## Repositories

**Frontend:** <https://github.com/COMP90082-2022-SM2/GE-RedBack-FrontEnd>

**Backend:** <https://github.com/COMP90082-2022-SM2/GE-RedBack-BackEnd>

(Please find pdf version of confluence in <https://github.com/COMP90082-2022-SM2/GE-RedBack-FrontEnd/tree/master/docs>)

## Trello

Sprint1: <https://trello.com/b/8kFLu01k/ge-sprint1>

Sprint2: <https://trello.com/b/yI38PtXy/gesprint2>

Sprint3: <https://trello.com/b/2cVEIE2B/gesprint3>

## Deployments

Production	
Frontend (Website)	<a href="https://game-of-ethics-2022.herokuapp.com/">https://game-of-ethics-2022.herokuapp.com/</a>
Backend (API)	<a href="https://ge-backend.herokuapp.com/">https://ge-backend.herokuapp.com/</a>

## Team RedBack

Roles		Responsibilities	Name
Scrum Master		Specify the development approach of the development team. Resolve difficulties encountered by development members in a timely manner. Ensure close cooperation between development members and protect the development team from outside interference.	Decheng Xu
Product Owner		Confirm product features, release dates and content. Maintain close contact with stakeholders and provide timely feedback on team progress and stakeholder needs. Set initial feature priorities and update them with each iteration.	Jiazhe Hou
Dev Team Members	Front end	1. Implement front-end ui and interaction development requirements to ensure excellent user experience on different platforms and devices. 2. Continuous optimisation of front-end experience and page response speed, as well as compatibility and execution efficiency.	Yujie Bai Jiazhe Hou Eryi Zhou
	Back end	1. Responsible for web front-end modification and optimization upgrade, development and program modification according to the website business needs. 2. Responsible for the technical maintenance of the daily operation of the website, and actively and effectively deal with any problems that arise.	Eryi Zhou Haiyu Hao Jiazhe Hou
	Tester	1. Analyse test points, define test scope, and determine test scope based on requirements documents and design documents. 2. Independently responsible for project testing, controlling project progress, reporting project status, reducing project risks, ensuring project quality.	Yujie Bai Decheng Xu

Subject matter experts	Answer questions about project requirements. Helping to confirm the direction of development at the beginning of each sprint. Help review project content and provide feedback at the end of each sprint.	supervisor
------------------------	---	------------

## Recent space activity

**Haiyu Hao**

Boeing executive 17 •  
Airline pilot 17 •  
Aeronautical engineer 18 •  
Software developer 18 •  
Federal Aviation Administration (FAA) official 19 •

## Space contributors

- [Haiyu Hao \(17\)](#)
- [Jiazhe Hou \(1\)](#)
- [Yujie Bai \(1\)](#)
- [Decheng Xu \(14\)](#)
- [Eryi Zhou \(46\)](#)
- ...

# Requirements

## Introduction

The Requirements page contains all the material for the requirements analysis phase, from the background of the project to personas, from motivational models(do-be-feel lists) to user stories. There are four subsections:

### Background

This page displays the project overview, game background, and client goals.

### Motivational modelling

This page presents the motivational model of the project, which includes the do-be-feel list and goal model.

### Personas

This page designs the personas for the project based on the possible users of the game.

### User Stories & Acceptance Criteria

This page shows all possible user stories based on the improvement of this project.

# Background

## Change log

Version date	Editor	Comment
2022-08-19	Decheng Xu	Initial Version
2022-08-20	Decheng Xu	Version2
2022-09-16	Jiazhe Hou	Version3

## Product overview

The game provides a virtual platform to simulate the design process of aircrafts with an educational meaning. It helps students to focus on the contradictory information and make decisions, which will improve their critical thinking and problem solving skills. The prototype that was handed over by the previous team has a very complex flow and there is room to improve the user interface. The goal of this project is to shorten the flow of the game, improve the user experience, and make the game more interesting, so that it can be used as a teaching method by teachers in law school or a wider range. The client would like to add features to the game such as a timer that were mentioned in the previous development but not implemented. Currently all characters have exactly the same options when faced with an unexpected event, and it is necessary to highlight the characteristics and features of different characters so that they can have different levels of influence when making decisions. This product can be used as a tool when teaching ethics courses to help students better understand responsibility and ethics.

## Game background

This game is inspired by the two Boeing 737 two accidents. In this game, players will be put in a scenario of building a model of airplane flight simulation. Five players will take on the roles of the five members in the project. When playing, players will have a chat and try to finish a successful project. The game has five roles, which are

- Aeronautical engineer
- Airline pilot
- Boeing executive
- Federal Aviation Administration (FAA) official
- Software developer

Different people have different perspectives. Each player need to keep thinking and make a moral decisions based on their responsibility facing incident from their own standpoint. Finally ultimately guide the plane through the disaster.

## Client goals

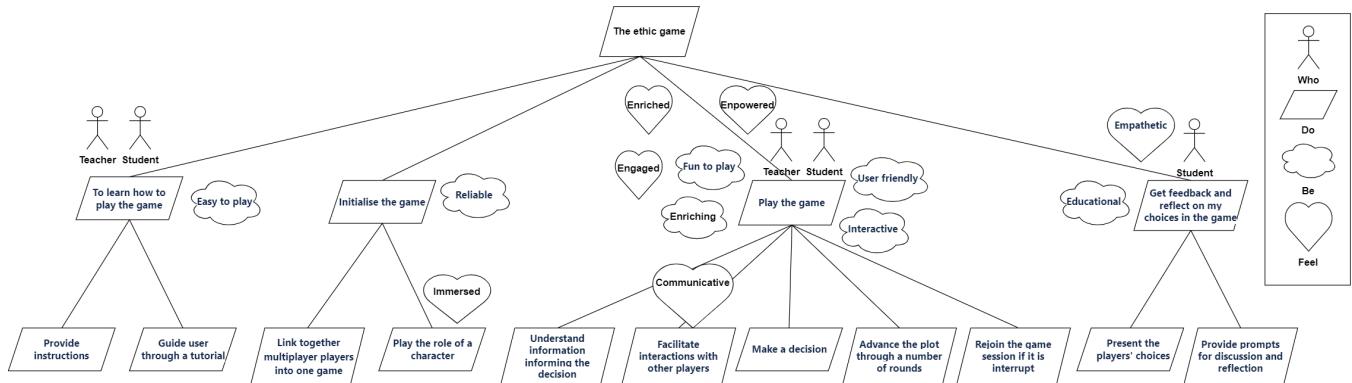
The basic goal was the need to successfully deploy the game to the cloud, fix situations where the game would get stuck halfway through and other bug fixes. Shorten the whole game play process by reducing the number of in-game issues. Improve the user's gaming experience. Test the entire game process to make the gameplay richer while highlighting the educational aspect of responsibility and ethics.

# Motivational modelling

## Do-Be-Feel List

Who	Do	Be	Feel
Student	To learn how to play the game	Provide instructions	Enriching
Teacher		Guide user through a tutorial	Easy to play
	Initialise the game	Link together multiplayer players into one game	Fun to play
		Play the role of a character	User friendly
	Play the game	Understand information informing the decision	Interactive
		Facilitate interactions with other players	Reliable
		Make a decision in limited discussion time	Educational
		Advance the plot through a number of rounds	
		Rejoin the game session if it is interrupted	
	Get feedback and reflect on my choices in the game	Present the players' choices	
		Provide prompts for discussion and reflection	

## Goal Model



# Personas

## Change log

Version date	Editor	Comment
2022-08-19	Haiyu Hao	Initial Version
2022-08-20	Haiyu Hao	Version2
2022-09-01	Haiyu Hao	Version3

## Design ideas

There are three personas for this project.

### 1. Student

- study ethics
- study engineering

### 2. Teacher

- ethics professor

We design a student and a teacher who need to study ethics based on the target user model. We also designed a persona with potential users. This persona needs to learn extra ethics for some reason.

# Anne Hathaway



"The ultimate value of life lies in the ability to wake up and think, not just to survive."

Age: 19-22

Work: Philosophy Student

Family: Sound family, one brother and sister

Location: Melbourne, VIC, Australia

## Goals

- Deeper understanding of ethics through games
- Self-reflection and summarising the knowledge of ethics by the game
- Use interesting methods to understand boring ethics
- Observe the behavior of others in the game

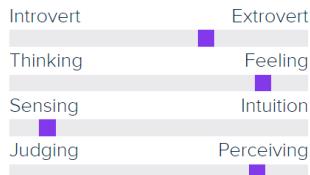
## Frustrations

- Not clear about the process and rules of the game.
- Not able apply ethics principle to real-life case
- Lack of opportunities to observe people's reactions in related to moral good.

## Motivation

- Able to achieve excellent grades in ethics subjects.
- Apply ethics principle to real-life case.
- Understand the theoretical knowledge of ethics from different perspectives.

## Personality



## Skills



## Bio

Anne Hathaway is an undergraduate student at the University of Melbourne. Her major is philosophy, and this is her second academic year. Anne has a lot to learn about ethics this school year. She is now very confused about the learning content of this part, because she cannot verify the learning content through practice and has no way to apply what she has learned in practice.

Annie hopes to validate what she has learned in some practical examples. Anne is also interested in trying new ways to learn about ethics. At the same time, observing different characters' different answers to questions can better help her understand some more profound ethics. However, as a non-technical person, she worries about the game's overly complicated operation. Anne wanted a simplified and fun tool so she could focus more on learning to understand ethics.

## Calvin Ferdinand



*"The Principle of Least Astonishment  
Make a user interface as consistent  
and as predictable as possible."*

**Age: 23-26**

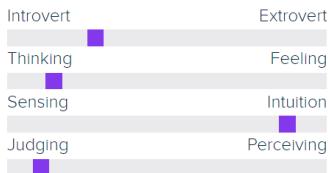
**Work: Engineering student**

**Family: One-parent family**

**Location: Melbourne, VIC,**

**Australia**

### Personality



### Goals

- Achieving good grades in courses in artificial intelligence ethics
- Learn more about ethics
- Help students understand the relevant knowledge of ethics

### Frustrations

- It is still very difficult for him to learn the theoretical knowledge of ethics.
- Calvin rarely communicates with people, and multiplayer games are a challenge for him.

### Motivation

- Relieve the stress of your studies and your family.
- Increase extracurricular knowledge and enrich his thinking

### Skills



### Bio

Calvin is a graduate student at the University of Melbourne. His major is information technology. He spends most of his time learning programming languages and algorithms. In a recent course Calvin learned about the ethics of artificial intelligence. Calvin is very interested in this part.

Calvin is happy to study ethics in his spare time, which can reduce the stress of programming and improve his own personality. Calvin hopes to use his knowledge to help students understand the relevant knowledge of ethics.

## Graeme Heman



*"The greatest happiness of teachers is to send a group of children to the ideal shore."*

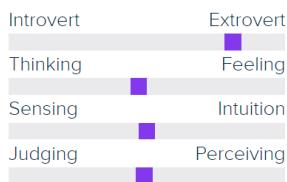
Age: 45-60

Work: Professor of Philosophy

Family: Married, 2 kids.

Location: Melbourne, VIC, Australia

### Personality



### Goals

- Teaching students ethics and wanting students to be engaged.
- Hoping students can learn and reflect on real-life cases by playing the game.
- Try new ways of teaching.

### Frustrations

- Different levels of ethics understanding compared to students.
- Graeme has never played any video games.

### Motivation

- Find out what students are thinking through games.
- Hope that students can gain more through ethics courses.

### Skill



### Bio

Graeme is a Professor of Philosophy at the University of Melbourne. He has extensive teaching experience. But recently he found it difficult to interact with students in class. Though philosophical studies are beneficial for people to improve their problem-solving, persuasion, and writing skills. How do make students really feel the topic of morality? How do ethical topics relate to students' future lives? These questions have always been a conundrum in the teaching career of professors.

The professor knew that this knowledge was boring, so he wanted to try a new way of teaching to change that. The professor hopes that the theoretical knowledge about ethics can be described in the form of games, and the problems in the game can make students reflect on real-life cases

# User Stories & Acceptance Criteria

## Analysis of requirements

### What is in-scope?

The overarching objective of this project is to provide an online game of ethic available for ethics education, allowing users especially students who learns ethics to simulate ethical issues they have to face in an industry setting and learn how to make better decisions as a result. The following requirements are listed as a product backlog in order of relative importance, as estimated from the provided case study.

#### Functional Requirements

- Show an overall report in the end of the game
- Show the option of players after every question
- Add a timer in the discussion session
- Efficiently allow players to play the game in a user friendly manner

#### Non-functional Requirements

- Testing and debugging the previous version of the game
- Simplifying the game to make it more fun and motivational
- Ensure a high level of security for the system's data
- The system must be available for 24/7
- The system must be delivered and maintained at a low cost

### What is out-of-scope?

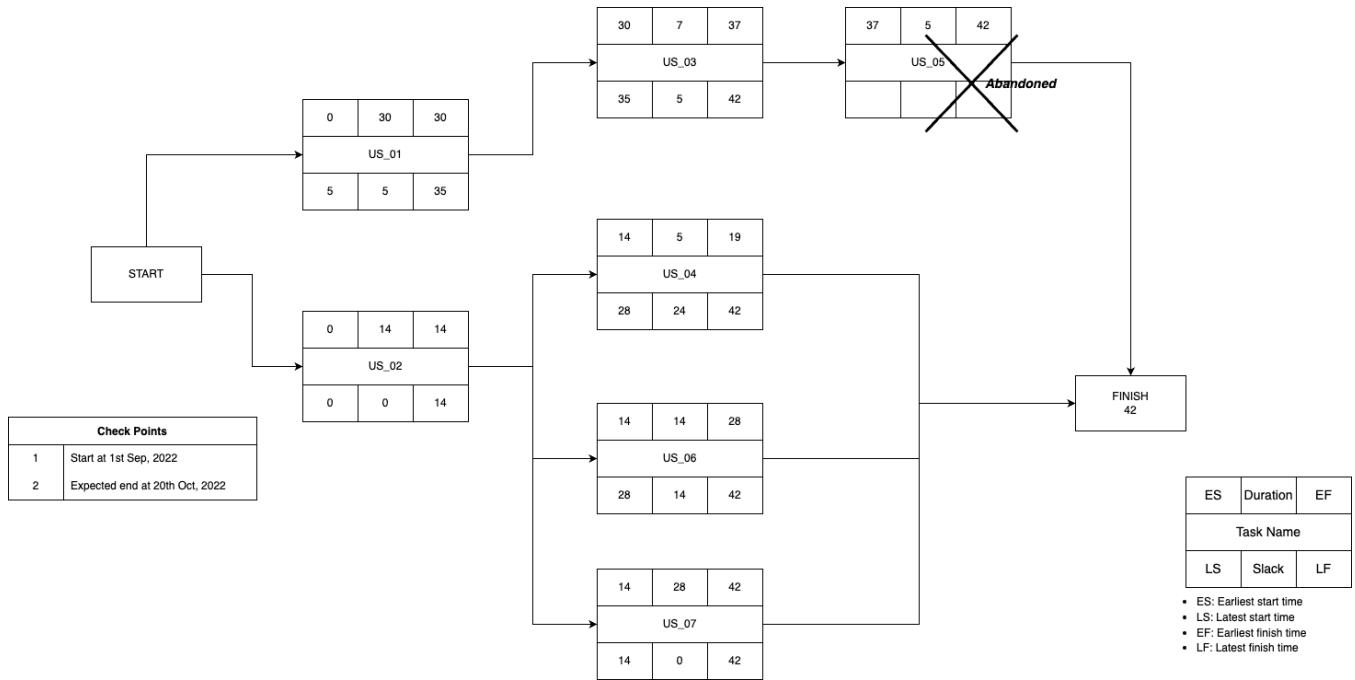
As probable expansions of the system, the following requirements should not be taken into consideration.

- Platform migration of the game
- The ability to login as an administrator for teachers
- Play the game with different language

## User stories

User story ID	As a ...	I want ...	so ...	Priority	Spent	Estimated Workload (Person days)	Achieved
US_01	teacher	For the decision-making process to be relatively simple	the game is fast-paced, which should improve student engagement and outcomes.	Must have	2	14	✓
US_02	student	To submit my option successfully	I can play the game without getting stuck.	Must have	2	14	✓
US_03	student	To get a overall report that shows decisions made by other users for each questions throughout the game	I can clearly see whether or not I achieved my character goals or not, and also reflect upon whether personal success, if attained, came at any cost to plot outcomes.	Must have	3	7	✓
US_04	student /teacher	To be able to rejoin the game session	I can continue the game if it is interrupted.	Better have	3	5	✓
US_05	student	To be able to see all chosen decisions made for all of my personal and group decisions.	I can get some insight into what other group members decided, and recall any forgotten decisions made earlier in the game.	Better have	3	5	Deleted
US_06	teacher	To add a timer to limit the discussion time for each decision in game	I can make students put more focus on the contradictive information and force them to make decisions, which will improve their critical thinking of whether their decisions are right to do.	Better have	3	14	✓
US_07	student /teacher	To have an uncluttered and appealing user interface	I can have a better experience.	Better have	3	14	✓

## PERT Chart



## Detailed test table and results

User Story ID	User story	Acceptance Criteria ID	Given	When	Then	Acceptance Test ID	Test*	Expected Result	Test Pass /Fail
US_01	As a teacher I want to be relatively simple for the decision-making process so the game is fast-paced, which should improve student engagement and outcomes.	AC_01	The teacher control the game time.	The student plays the game.	The game is fast-paced, which should improve student engagement and outcomes.	AT_01	Play the whole game and test the gaming time.	The game process is in 20 minutes.	✓
US_02	As a student I want to submit my option successfully so I can play the game without getting stuck.	AC_02	The student submit the option.	Submit the option of each questions.	The game continues to next question fluently without stuck.	AT_02	All Users click the submit bottom in question page.	Next question page is displayed	✓
US_03	As a student I want to get a overall report that shows decisions made by other users for each questions throughout the game so I can clearly see whether or not I achieved my character goals or not, and also reflect upon whether personal success, if attained, came at any cost to plot outcomes.	AC_03	The student submit the option.	Submit the option of each questions.	The student can see overall report of other players submitted.	AT_03	All Users click the submit bottom in question page.	Display whether game characters goals achieved and the overall report of other users decision for the question.	✓
US_04	As a student/teacher I want to be able to rejoin the game session so I can continue the game if it is interrupted.	AC_04	The student want to rejoin the game.	The game is interrupted and return to the index page.	The game can continue and locate the latest process.	AT_04	Quit the game and click the rejoin bottom to rejoin the game in the home page.	Rejoin the game and transform to the latest game process.	✓
US_05	As a student I want to be able to see all chosen decisions made for all of my personal and group decisions. so I can get some insight into what other group members decided, and recall any forgotten decisions made earlier in the game.	AC_05	The student sees all the options chosen.	The student reaches the outcome page.	The student can get some insight into what other group members decided, and recall any forgotten decisions made earlier in the game.	AT_05	Submit the final question and reach the outcome page	All chosen decisions are shown in the outcome page	Deleted

US_06	As a teacher I want to add a timer to limit the discussion time for each decision in game so I can make students put more focus on the contradictive information and force them to make decisions, which will improve their critical thinking of whether their decisions are right to do.	AC_06	The student checks how much time they left to answer the questions.	The student answers the questions.	The student puts more focus on the contradictive information and forced to make decisions in limited time.	AT_06	Enter the question's pages	A timer is working and shows how many times left	<input checked="" type="checkbox"/>
US_07	As a student/teacher I want to have an uncluttered and appealing user interface, so I can have a better experience.	AC_07	The student creates a game.	The student plays the game.	The student can see different animations or changes when click button or choose an option.	AT_07	The animations and background are rendered successfully.	The webpage is user-friendly and attractive.	<input checked="" type="checkbox"/>

# General Process Documentation

General Process Documentation contains:

- Roles and Responsibilities
- Communication Plan

# Communication Plan

Stakeholder	Objective	Format	Contingency ways	Frequency	Owner	Importance
Eduardo Araujo Oliveira	Report progress updates, recording Sally's comments on current projects.	Sprint review-zoom	Emile report	Weekly	Product Owner	High
	Answer project team questions about requirements.	Sprint retrospective-zoom	Emile report	Weekly	Product Owner	High
	Identify the current resource needs of the development team.	Emile	SMS	Weekly	Product Owner	High
Project Team	Specify the work that needs to be done and discuss how it will be done before it can be done.	Sprint planning meet-zoom	Live Chat Tools	At the begin of each sprint	Scrum master	High
	Showing currently completed functions.	Sprint review-zoom	Live Chat Tools	Weekly	Scrum master	High
	Confirmation of what will be developed in the next sprint and the expected outcomes.	Sprint retrospective-zoom	Live Chat Tools	Weekly	Scrum master	High
	Keeping track of team progress & task delegation.	Kanban(Trello)	Report	Daily	Scrum master	Medium
	Demonstrate the difficulties currently encountered in the sprint (not addressed at the meeting).	Stands-up zoom	Live Chat Tools	Daily	Scrum master	Medium
Supervisor	Report the technological problems encountered during the process and seek help.	Tutorial-zoom	Emile	weekly	Scrum master	Medium

# Roles and Responsibilities

Roles	Responsibilities		Name
Scrum Master	Specify the development approach of the development team. Resolve difficulties encountered by development members in a timely manner. Ensure close cooperation between development members and protect the development team from outside interference.		Decheng Xu
Product Owner	Confirm product features, release dates and content. Maintain close contact with stakeholders and provide timely feedback on team progress and stakeholder needs. Set initial feature priorities and update them with each iteration.		Jiazhe Hou
Dev Team Members	Front end	1. Implement front-end ui and interaction development requirements to ensure excellent user experience on different platforms and devices.  2. Continuous optimisation of front-end experience and page response speed, as well as compatibility and execution efficiency.	Yujie Bai  Jiazhe Hou
	Back end	1. Responsible for web front-end modification and optimization upgrade, development and program modification according to the website business needs.  2. Responsible for the technical maintenance of the daily operation of the website, and actively and effectively deal with any problems that arise.	Eryi Zhou  Haiyu Hao  Decheng Xu
	Tester	1. Analyse test points, define test scope, and determine test scope based on requirements documents and design documents.  2. Independently responsible for project testing, controlling project progress, reporting project status, reducing project risks, ensuring project quality.	Yujie Bai  Jiazhe Hou
Subject matter experts	Answer questions about project requirements. Helping to confirm the direction of development at the beginning of each sprint. Help review project content and provide feedback at the end of each sprint.		Supervisor

# Research

## Introduction:

The Research page collects game information related to the project, presents all possible game ideas from the developers, and prepares for the final form of the game. There are four subsections:

### Game Formats

This page discusses some possible formats for the project, as well as formats' descriptions, examples, cons, and pros.

### Platform

This page describes the potential techniques the development team might use in the game, including the game engine and the game framework.

### Proposed gameplay design

This page describes the three proposals the development team designed for the game, with a detailed description of each proposal, and highlights the final one discussed with the client at the beginning of the page.

### Dev-Ops Tools

This page discusses tools that will be utilised by the deployment team to achieve tasks such as:

- Building a consistent development environment across developers
- Continuous Integration (CI)
- Continuous Deployment (CD)

To learn more information about the subsections, please follow the links below:

[Game Formats](#)

[Platform](#)

[Proposed gameplay designs](#)

[Dev-Ops Tools](#)

# Back-End Tools

For back-end technology, we narrowed our search space to two main contenders, Java and Node.js.

Technology	Description	Advantages	Disadvantages
Java	Java is an Object-Oriented, general-purpose programming language and class-based. Developers can use the principle – “write once, run anywhere” with Java. Java is portable, means a program written for any platform must run similarly on a combination of hardware and operating system.	<ul style="list-style-type: none"> <li>Extensive concurrency, networking and GUI support</li> <li>As a low-level programming language, it is extremely flexible and allows us to customise in almost any conceivable way</li> <li>Extensive libraries and documentation for building back-end systems with Java exist</li> </ul>	<ul style="list-style-type: none"> <li>Using this means that there will be a disconnect between front-end language and back-end language, which complicates integration</li> <li>More difficult to build systems, since lower-level languages lack the constructs necessary to speed development</li> </ul>
Node.js	Node JS is a runtime library and environment which is cross-platform and used for creating running Javascript applications outside the browser. It is a free and open source and utilized for creating server-side JS applications. Node JS allows developers to execute their code on the server-side. It provides a faster way to write scripts that are scalable and light.	<ul style="list-style-type: none"> <li>Can be utilised to write BOTH front-end and back-end code, it allows for use of the same Javascript code for both ends</li> <li>Server-side capabilities are extensively provided, Node.js was originally meant for back-end development</li> <li>An event-based model to address scalability, and rich libraries to simplify coding</li> <li>Much faster and much more scalable than Java</li> </ul>	<ul style="list-style-type: none"> <li>Less flexibility, we can only work within the capabilities Node.js provides</li> <li>Heavy reliance on third-party libraries leads to reliability risk, what if a library is bugged?</li> <li>Weaker concurrency constructs than Java</li> </ul>

# Database

Below lists a few alternative database options that we have considered, featuring a description, some advantages and disadvantages, and a price point.

Database Type	Database	Description	Advantages	Disadvantages
SQL (RDBMS)	MySQL	MySQL stores its data in tables and uses the structured query language (SQL) to access the data. MySQL uses schemas to define the database structure, requiring that all rows within a table have the same structure with values being represented by a specific data type.	<ul style="list-style-type: none"> <li>Supports features like Master-Slave Replication, Scale-Out</li> <li>Query Cache for repeatedly used statements</li> <li>You can easily learn and troubleshoot MySQL from different sources like blogs, white papers, and books.</li> </ul>	<ul style="list-style-type: none"> <li>Transactions related to system catalog are not ACID compliant</li> <li>Sometimes a server crash can corrupt the system catalog</li> <li>Stored procedures are not cacheable</li> <li>MySQL tables which is used for the procedure or trigger are most pre-locked.</li> </ul>
	SQL Azure	Microsoft Azure SQL Database is a managed cloud database provided as part of Microsoft Azure.	<ul style="list-style-type: none"> <li>High Availability</li> <li>Data Security</li> <li>Scalability</li> </ul>	<ul style="list-style-type: none"> <li>Requires Management</li> <li>Requires Platform Expertise</li> </ul>
NoSQL	MongoDb	In MongoDB, data is stored in JSON-like documents that can have varied structures. To improve query speed, MongoDB can store related data together, which is accessed using the MongoDB query language. MongoDB is schema-free, allowing you to create documents without having to define the structure of the document first. These documents can be easily changed by adding or deleting fields.	<ul style="list-style-type: none"> <li>No schema</li> <li>Fast</li> </ul>	<ul style="list-style-type: none"> <li>Transactions using MongoDB are complex</li> <li>In MongoDB, there is no provision for Stored Procedure or functions, so you can't implement any business logic in the database level, which you can do in any RDBMS systems.</li> </ul>

# Dev-Ops Tools

Ensuring that developers are able to use a consistent environment.

In order to reduce issues in taking code from a development to a staging or production environment, we can simulate these environments.

Tool	Description	Advantages	Disadvantages
No Tool	Instead of using a tool, we provide instructions to set up the development environment	<ul style="list-style-type: none"> <li>Developers don't need to learn a new tool</li> </ul>	<ul style="list-style-type: none"> <li>Ability to have a consistent environment across all developers is limited even with documentation</li> </ul>
Docker	Use Containers (essentially a lightweight virtual machine) during development, developers use the containerised services	<ul style="list-style-type: none"> <li>Developers will need to learn how to set up the software and run containers</li> </ul>	<ul style="list-style-type: none"> <li>Allows for fine-grained control of the development environment (version of tools)</li> </ul>
Virtual Machine	A virtual machine is an emulation of a computer system.	<ul style="list-style-type: none"> <li>Full control over the environment (more so than docker)</li> </ul>	<ul style="list-style-type: none"> <li>Higher performance overhead</li> <li>More complex to set up and use than other solutions</li> </ul>

## Continuous Integration / Continuous Deployment

**Continuous Integration (CI):** The practice of merging developers code into a shared mainline (shared branch) on a constant basis. This helps us to avoid integration issues when developers need to merge all their features into one product.

**Continuous Deployment (CD):** A software release process whereby new code is tested in an automated fashion to ensure that the changes made are suitable for a production environment. This allows us to ensure that at any time we have a stable product that can be deployed.

Upon Researching It was found that there are a plethora of CI/CD tools, To limit the scope we looked at tools that are:

1. Well known
2. Have a large community backing

Tool	Description	Advantages	Disadvantages
Bamboo	Atlassian Tool designed for CI/CD	<ul style="list-style-type: none"> <li>Strong Integration with Bitbucket (since they are developed by the same company)</li> <li>Most likely freely available for the project (will have to check if Melbourne Uni provides us with the tool)</li> </ul>	<ul style="list-style-type: none"> <li>Unsure if unimelb provides hosted bamboo (the service appears to have no recent activity)</li> </ul>
Jenkins	Open source automation server designed for CI/CD processes	<ul style="list-style-type: none"> <li>Has a large community backing           <ul style="list-style-type: none"> <li>Many plugins</li> <li>Well Documented</li> </ul> </li> </ul>	<ul style="list-style-type: none"> <li>Requires us to set up our own infrastructure for compiling and testing builds</li> </ul>
CircleCi	CI/CD tool, Closed source.	<ul style="list-style-type: none"> <li>Has "orbs" (Preconfigured templates for integrating with many services)</li> </ul>	<ul style="list-style-type: none"> <li>Limited feature set on the free tier           <ul style="list-style-type: none"> <li>Can only run 1 job at a time</li> <li>Has a credits system (limited number of builds per month)</li> </ul> </li> </ul>
BuildKite	A platform for running fast, secure, and scalable continuous integration pipelines on your own infrastructure.	<ul style="list-style-type: none"> <li>Unlimited number of builds and number of concurrent jobs on the free tier</li> </ul>	<ul style="list-style-type: none"> <li>Requires us to set up our own infrastructure for compiling and testing builds</li> </ul>
Travis CI	Travis CI is a hosted continuous integration service used to build and test software projects hosted at GitHub and Bitbucket.	<ul style="list-style-type: none"> <li>Easy setup (compared to other tools)</li> </ul>	<ul style="list-style-type: none"> <li>Free only for public repositories, closed repositories have a 1-month free trial</li> </ul>

Gitlab	GitLab is a web-based DevOps lifecycle tool that provides a Git-repository manager providing wiki, issue-tracking and continuous integration/continuous deployment pipeline features	<ul style="list-style-type: none"> <li>Has an integration with Bitbucket</li> <li>Offers a free tier cloud-hosted solution</li> </ul>	• limited to 2,000 CI pipeline minutes per month
--------	--	---	--

## Cloud Hosting

Service	Description	Ease of use	Flexibility	Cost/Limitations of free tier
AWS	Amazon cloud hosting services.	<p>Offers GitHub integration through a tool called "Cloud Deploy".</p> <p>We would need to further investigate how we deploy the application (docker image on ec2 instance)</p>	Extremely Flexible. AWS offers services for many technologies	AWS offers both always free and 12 month free tiers for their services. The Limitations depend on the service. For example, EC2 instances offer 750 hours free per month for 12 months. For info <a href="#">here</a>
Heroku	Platform as a service. Built on top of AWS.	<p>Yes, Has GitHub integration built-in.</p> <p>Heroku will automatically install dependencies and run the application.</p>	Low flexibility. Heroku is designed to 'do one thing and do it well' as opposed to 'doing everything'. For example, there doesn't seem to be an easy way to add load balancing.	Heroku offers free 'dynos' however only on personal accounts. Therefore we need to create 1 account and share the credentials
Digital Ocean	Cloud infrastructure provider	No. Requires a third-party CI/CD tool	Very Flexible. Offers many cloud services. More info <a href="#">here</a> .	Offer \$50 in free credit which lasts for 12 months.
Google cloud	Cloud infrastructure provider	Yes, GCloud has a tool called "Cloud Build to achieve this".	Very Flexible. It offers many cloud services. More info <a href="#">here</a>	\$300 credit for 12 months. In addition, they offer free tiers for their services. More info <a href="#">here</a> . Based on personal use I believe this is enough for the purposes of the project.
Mongo Atlas	MongoDB service provider	Yes, setup of the cluster is done through UI. No code required.	No. Only offers MongoDB service.	Free MongoDB cluster with limitation of 500MB.

## Decisions

### Development environment

If team members need help setting up their development environments we will use **Docker** to achieve this.

## CI/CD

For CI/CD we will use **Gitlab** since it has a free tier cloud-hosted solution with limitations that should not impact our project.

## Cloud Hosting

For Staging, we will use **Heroku** and **Mongo Atlas** due to ease of use.

For Production, we may use **AWS** since it adds flexibility. This will depend on the production infrastructure we will need.

# Front-End Tools

## Introduction

Having decided our project is going to be a web application, on this page we will compare different options for front-end technology to use, and document the decision made and why.

## Template engine vs framework

The first decision was whether a simpler template engine would suffice, or if a more powerful front-end framework would be needed.

Templating engines (e.g. Pug, Handlebars) would allow us to use static HTML-like templates with the content filled in based on data coming from the back-end. It would be a viable option.

However a framework (e.g. React, Angular) is capable of running the application on a single page, sending and receiving data to the back-end and updating accordingly - it would enable us to provide a much more fluid and interactive experience. Using a framework may be more challenging (especially as there is not much direct experience on the team) but would also be a good learning opportunity.

On this basis, we have decided at this stage that we will go with a framework for the front-end.

## Front-end framework comparison

We looked at the more frequently used frameworks, and selected Vue and React to compare and contrast. These are two widely-used frameworks that are somewhat similar in terms of their capabilities.

Tool	Language	Description	Pros	Cons
Vue.js	Javascript	Vue.js is an open-source Model–view–viewmodel JavaScript framework for building user interfaces and single-page applications.	<ul style="list-style-type: none"> <li>• Easy to learn and use for beginners</li> <li>• Tiny size</li> <li>• Virtual DOM rendering and performance</li> <li>• Reactive two-way data binding</li> <li>• Single file component and readability</li> <li>• Concise documentation</li> </ul>	<ul style="list-style-type: none"> <li>• Lack of support for large-scale projects</li> <li>• Reactivity complexity</li> </ul>
React	Javascript	React is a JavaScript library for building user interfaces. React can be used as a base in the development of single-page or mobile applications.	<ul style="list-style-type: none"> <li>• Powerful framework</li> <li>• Virtual DOM</li> <li>• Open source facebook library</li> <li>• Widely-used, in-demand skill</li> </ul>	<ul style="list-style-type: none"> <li>• Difficult to set-up and use</li> <li>• Documentation is only adequate</li> </ul>

In general, Vue would be easier to use while still capable of doing what we need for the project, given that our front-end needs are not that complex. React on the other hand is powerful and widely-used, and a good in-demand tool to learn. But it is hard to justify using React if it is much harder to use for little benefit for the project.

## Conclusion

On this basis, we have decided to proceed with Vue.js as a front-end framework for our project.

# Game Formats

Format Name	Description	Advantages	Disadvantages	Examples
Card Based Game	User is presented with a choice (card or cards) and must either select a card (in the case of multiple cards) or swipe a card (in the case of one card) to make a decision.	<ul style="list-style-type: none"> <li>Allows us to focus on a deep story line as development is simplified</li> </ul>	<ul style="list-style-type: none"> <li>May potentially make decisions seem less important</li> <li>Format doesn't hide that the game is decision based</li> </ul>	<ul style="list-style-type: none"> <li>• Reigns</li> </ul>
First Person Perspective	The player will chose a role in the game and promote the episods by making actions or choices. Each episode focuses on a particular character, and each choice will influence all episodes.	<ul style="list-style-type: none"> <li>Combine the role play and strategy</li> <li>The player has a strong sense of substitution</li> <li>Based on the player's view</li> <li>The player has overall image of the game structure</li> </ul>	<ul style="list-style-type: none"> <li>Too much flexibility may add complexity</li> </ul>	<ul style="list-style-type: none"> <li>• Detroit: Become human</li> </ul>
Interactive /Adventure	User makes decisions by clicking among choices. Choices will then change the environment	<ul style="list-style-type: none"> <li>A dynamic environment can make choices feel more meaningful</li> <li>Multiple ends</li> <li>More close to the real world</li> </ul>	<ul style="list-style-type: none"> <li>Can potentially limit depth of story line as development time will need to be dedicated to visuals</li> <li>Heavily depend on the story script</li> </ul>	<ul style="list-style-type: none"> <li>• Talk To strangers</li> <li>• Replica</li> <li>• Beholder</li> <li>• Detroit: Become Human</li> </ul>
SIM (Simulation Game) building or managing	The player will play a role in an aspect of reality and control the development.	<ul style="list-style-type: none"> <li>Combine the role play and strategy</li> <li>Engage players more</li> </ul>	<ul style="list-style-type: none"> <li>No specific ending and rewards</li> </ul>	<ul style="list-style-type: none"> <li>• SimCity</li> </ul>
Turn-Based Strategy	Players take turns playing (Making choices). Strategy games tend to have more complicated metrics systems	<ul style="list-style-type: none"> <li>Users / Teachers may be able to better quantify the 'results' after a game is played due to this format's tendency to have more metrics</li> </ul>	<ul style="list-style-type: none"> <li>Learning curve for users to play game compared to other formats</li> <li>May take a long time for each round</li> </ul>	<ul style="list-style-type: none"> <li>• Democracy 3</li> <li>• Sid Meier's Civilization VI</li> <li>• Chess</li> <li>• Werewolf</li> </ul>

# Platform

## Introduction:

This page aims to discuss the related technique, such as language for backend and frontend, for game design and their advantages and disadvantages.

### 1. Use Engines: Game engines can help us to render so that we can concentrate on the logic and design parts.

Engines	Language Skill	Price	Pros	Cons	Libraries	Example	Link
Construct 3	No language skill needed; operate with GUI	free	<ul style="list-style-type: none"> <li>Powerful event-based system</li> <li>Easy to operate</li> </ul>				
GameMaker Studio 2	Dragging (no language skill needed) OR C	free for development paid for publishing	<ul style="list-style-type: none"> <li>Extensive integration toolkit</li> </ul>	<ul style="list-style-type: none"> <li>paid for publishing</li> </ul>			
Unity	C#	free	<ul style="list-style-type: none"> <li>Full-featured</li> <li>Much information and tutorials</li> <li>Extensible (Customizable)</li> </ul>	<ul style="list-style-type: none"> <li>Unfriendly to a beginner</li> <li>More resource needed (CPU, GPU...)</li> </ul>			
Godot	GDScript (similar to python) OR C++ (can be imported as GDScript)	free	<ul style="list-style-type: none"> <li>3D and 2D, better in 2D</li> <li>Multi-platform</li> <li>Visual editor</li> <li>Lightweight</li> <li>Easier animation and scene system</li> </ul>	<ul style="list-style-type: none"> <li>less tutorial/demo plugin template</li> </ul>	asset lib ( <a href="#">Godot Asset Library</a> )	<ul style="list-style-type: none"> <li>Spooky Ghost</li> <li>Dotsots Dot Com (Steam)</li> <li>Grimante (Steam)</li> </ul>	
Unreal Engine 4 (UE4)	C++ (Encapsulated via Epic, less difficult than C++)	free for studying purpose	<ul style="list-style-type: none"> <li>3D 2D</li> <li>better for multithreading</li> <li>better effects</li> </ul>	<ul style="list-style-type: none"> <li>Hard to start</li> <li>More complex logic needed</li> </ul>			
cocos2d	C++, Lua or JavaScript	free	<ul style="list-style-type: none"> <li>Lightweight and efficient cross-platform</li> <li>Develop for multiple platforms: OSAndroidHTML5PC</li> <li>Open-source</li> <li>2D</li> </ul>	<ul style="list-style-type: none"> <li>Long compilation time</li> <li>Need time to learn</li> </ul>			
Phaser	JavaScript	free	<ul style="list-style-type: none"> <li>Friendly for beginners</li> <li>Open source codabase</li> <li>Keep together WebGL and Canvas components</li> </ul>				<a href="https://phaser.io">https://phaser.io</a>

### 2. Develop with pure code: Suitable for a small-scale project, it is efficient to develop.

- Including front-end, which uses HTML5css3 and JavaScript as the main language and server-end, which we can use java, python, C, C++.
- Pros:
  - There is no need to download and install the engine.
  - Totally free
  - The game can be published or played on multiple kinds of devices and OSs. e.g. App Store, integrated with a webpage, Google play
  - No compilation process when developing. Save time and lower computer configuration required
- Cons:
  - Excellent skill of front-language needed.
  - Callback function nesting
  - A huge amount of code
  - Need to consider rendering process
- Some GUI development tools can be used, for example, QT.

# Proposed Gameplay Designs

## Introduction:

There are three proposals for the project, first is Individual+group-based decision making, second is Round-based group decision making and the last one is Episodic game. The final proposal is the second one, Round-based group decision making game format.

## The details of three proposals:

### 1. Individual + group-based decision making

- Live multi-player game, each player accessing the game with their laptop, phone or tablet, while also sitting around a table face-to-face
- Players are assigned one of the five roles at the start of the game. Each role is assigned a particular set of goals that he/she is to fulfill by game's end.
- Player are presented with the next part of the plot, and is faced with a set of individual decisions. They also learn certain information relevant to an upcoming group decision.
- At certain times, the group is tasked with making a group decision - they are to discuss the decision face-to-face as a group, and then enter the decision on device(s). Depending on the nature of the decision, sometimes the players might all get an equal vote for the decision. Other times, some roles may have more power, or certain roles may even be exclusively given the final say (creating power asymmetry).
- Once the decision is made, the plot advances accordingly depending on the decision, and a new round starts/

This is similar to the round-based group decision making, however, players are tasked with making individual decisions on their device (and possibly even interacting with particular players one-on-one) in addition to the group-based decisions discussed face-to-face.

This would provide a very rich, realistic experience, however may be too complex to achieve as the plot would have to be able to handle many different combinations of decisions.

### 2. Round-based group decision making

This model is the same as Possibility #1 but simplified by removing individual decisions affecting the plot - instead the game is driven by rounds of group-based decisions.

In this proposal, the game is a live, multi-player game. Each player access the game using their own laptop/phone/tablet, while also sitting around a table face-to-face.

Players are assigned one of the five roles at the start of the game.

The game takes place in rounds. At the start of each round, each player is presented with the next part of the plot, and they also learn certain information relevant to an upcoming decision. The information given to each player depends on their role (creating information asymmetry between players; potential conflict!).

Then, the group is tasked with making **one** decision for the round - they are to discuss the decision face-to-face as a group, and then enter the decision on device(s). Depending on the nature of the decision, sometimes the players might all get an equal vote for the decision. Other times, some roles may have more power, or certain roles may even be exclusively given the final say (creating power asymmetry).

Once the decision is made, the plot advances accordingly depending on the decision, and a new round starts.

Although this idea is a bit more of an abstraction compared to real life, it is comparable to how a board game simplifies real life for the purpose of gameplay.

This would simplify the complexity of the game, compared to our original model where players making individual decisions, as it results in fewer combinations of decisions in the plot (compared to having 5 different roles making different decisions) - while still retaining the element of interactivity and group dynamics.

## Example round:

- **Context:** (to be incorporated into each player's plot and information):
  - Shared context: Everyone is working hard to drive the development of the 737 MAX, in time to fulfill the first order from Lion Air, which will be a huge win for everyone. The work looks to be on track.
  - **Boeing executive:** Lion Air's order will be a huge financial gain. Huge pressure to get everyone working hard and meet the deadline; otherwise huge \$ loss and face the Board
  - **Aeronautical Engineer:** also desperately needs to meet deadline for their own KPIs. But has identified a potentially critical issue regarding the reliance on one single AoA sensor.
  - **Software Developer:** wants to maintain a good pace on the software development; doesn't want some issue to disrupt their progress and cause everyone to pull long hours
  - **Pilot:** has info to share on what an AoA sensor does (+ it isn't normally catastrophic if it fails)
  - **FAA:** also has some info on why an AoA sensor might be important, past cases of it failing
- **Group discussion + decision:**
  - Should the Boeing Executive halt development in order to investigate the AoA sensor further?
  - For this decision: all players are instructed to vote, however the **Boeing Executive** has the ultimate say (while taking input from the team). However the development can also be stopped, if both **Aeronautical Engineer** and **Software Developer** agree to veto the Executive's decision

## 3. Episodic game

- Game is played on either a single or multiple devices (depending on number of players)
- Players are not assigned to characters, instead they are assigned goals (increase company revenue, increase safety, etc )
- All players vote on all decisions
- Game is structured in an Episodic way. We include particular key events leading up to the Boeing 737 maxi incidents. Consequences from past episodes will affect future ones
- Focus on one character at a time and each character will have an ending (not necessarily at the same time).

Example Episodes:

- **Boeing Executive** finding out about new generations of planes being developed by a competitor
- **Aeronautical Engineer** designing plane
- **Software Developer** developing flight automation software
- **Airline Pilot** goes to a flight simulator to learn about the new plane
- **Airline Pilot** flying the new plane
- **FAA Official** investigating crash: (this would be an example of an episode that is dependent on what occurs in previous episodes)

## Example

Assumptions:



1. Built with the structure of the game "Talk to strangers"

This means the storyline is first person and conversational based between one or many NPCs and the player. Of course this can be changed.

### Conventions

#### Characters

NPC (Non playable characters): represented by *italicised text*

Player: Represented by **bold text**

- An example of the player dialog is a situation where the player has 1 choice like this situation



## Types of choices

**Impactful:** represented by **bold underlined**

- leads to unlocking of future choices, influences ending

unimpactful: represented by underlined text

- may change some dialog but doesn't change future choices or ending
- These essentially give the player a sense of complexity without actually making the story complex

An Impactful choice will be denoted as follows

**Choice <name of choice>:** "Choice 1"(choice\_value) OR "Choice 2" (choice\_value) OR ...

Example an Impactful choice

**Choice** (make\_storyline\_for\_game): "Yeah sure, how bad can it be" (TRUE) OR "I'm a software developer, why am I doing this" (FALSE)

To simplify I believe we should stick to binary (two) choices, however we can allow for many choices with this convention by using values other than TRUE or FALSE

## Conditional Statement

Used to represent changes based on previous choices

**IF** (condition1 == value OR condition2 == value2) AND (condition3 == value3):

THEN

<Something here>

ELSE

<Something here>

**Note:** ELSE statement is optional

Example:

IF software\_students\_make\_storyline\_for\_game == TRUE:

THEN

**client:** Why did I enlist software engineering students to write a story

ELSE

**client:** This is a good story

## Endings

- In the episodes we will reference Endings by ID << Ending\_ID >>
- We can also concatenate endings like this << Ending\_ID1 >>, << Ending\_ID2 >>, ...

- **Example: Pilot dies, software engineer goes to jail, CEO is fired**
- We will then refer to the IDs with a table
- Endings can be placed at the **end of the episode OR in a conditional statement**

## Episode: Training day

*You are a pilot for <name of airline> and are learning about the new mk8 jumbo jet.*

*Instructor:* Hi, <name of pilot> I trust you read the training briefing for the this new mk8 jet

**Choice** admit\_to\_not\_reading\_manual: "I haven't had the time" (TRUE) OR "Yes I have" (FALSE)

**IF** admit\_to\_not\_reading\_manual == TRUE:

THEN

*Instructor:* No worries, this plane is very similar to the mk7, with a couple of tweaks to make the pilot's life easier.

ELSE

*Instructor:* That's great, since I have many pilots to train today I will skim over the details.

*Instructor:* <more generic dialog with unimpactful choices>

**IF** company\_adds\_autopilot\_changes\_documentation\* == TRUE

THEN

*Instructor:* Here are some important changes I need to mention. Due to changes in the plane's aerodynamics systems a new autopilot system has been installed. This system makes the system behave as the MK7 jet so there should be no noticeable changes when flying. However, if you need to you can disable the feature by pressing this button (points to button)

*Walking out of the session you bump into an old work colleague you have lost contact with.*

<Some dialogue with unimpactful choices>

*Work colleague:* How about we go grab a drink for old time's sake?

**Pilot:** Thanks but I better not, I have to bring those things tomorrow

*Work colleague:* ah, you know how it is, these things practically fly themselves. Don't worry it'll still be 8 hours from bottle to throttle.

**Choice** go\_for\_drinks: "go-to the bar" (TRUE) OR "go home" (FALSE)

\***note:** the **company\_adds\_autopilot\_changes\_documentation** would be a choice made in an earlier episode whether or not to inform pilots of changes to the autopilot system.

## Related Games

# Testing Frameworks

JavaScript testing frameworks	Introduction	Advantage
<b>MochaJS</b>	<p>Mocha is a feature-rich JavaScript test framework running on Node.js and in the browser, making asynchronous testing simple and fun. Mocha tests run serially, allowing for flexible and accurate reporting, while mapping uncaught exceptions to the correct test cases. Hosted on GitHub.</p>	<ul style="list-style-type: none"> <li>• Provides compatibility for both frontend and backend testing</li> <li>• NodeJS debugger is supported which makes error tracing easier</li> <li>• Accurate reporting</li> <li>• Provides support for all browsers including the headless Chrome library</li> <li>• Very convenient framework for the developers to write test cases</li> </ul>
<b>JEST</b>	<p>Jest is a delightful JavaScript Testing Framework with a focus on simplicity. It works with projects using: Babel, TypeScript, Node, React, Angular, Vue and more.</p>	<ul style="list-style-type: none"> <li>• Compatible with NodeJS, React, Angular, VueJS and other Babel based projects</li> <li>• Standard syntax with documentation support</li> <li>• Very fast and highly performant</li> <li>• Managing tests with larger objects is possible using Live Snapshots</li> </ul>
<b>Jasmine</b>	<p>Jasmine is a behavior-driven development framework for testing JavaScript code. It does not depend on any other JavaScript frameworks. It does not require a DOM. And it has a clean, obvious syntax so that you can easily write tests.</p>	<ul style="list-style-type: none"> <li>• Provides small, clean and straightforward syntax for easy testing</li> <li>• Does not require any Document Object Model (DOM)</li> <li>• Provides support for both frontend and backend tests</li> <li>• Ease in coding as the syntax used is very similar to a natural language</li> <li>• Strong documentation and community support</li> </ul>

# Game Design

## Introduction:

The Game design page aims to focus on designing the game format, crucial contents and details related to background and client's requirements. There are three subsections:

### Gameplay Page

This page discusses the main components and some of the preliminary design of the game by the development team.

### Plot Page

This page discusses one of the main game elements, the plot. It lists some related components of the plot, characters, decisions, outcomes, and so on.

### Characters Page

This page discusses the design of the game's characters. Furthermore, it describes details and the information about each character.

### Decision Tree

This contains the decision tree, comprising the final version of the plot and game content

# Gameplay

## Game Principles

- A 5-player game, where each player assumes the role of a key 'player' at Boeing during the 737 MAX project. It is likely that the company, and project, will be re-appropriated to conceal any linkage to the 737 MAX, since much about the consequences of this project is relatively well-known among the public
- To maximise complexity of decisions and reduce the flow of the game, each 'player' will be given a set of personal goals that they are to fulfill before game end, throughout in intervals, players will be given vague updates on these goals to let them know whether they are going on the right track with their decision-making, these goals will necessarily conflict with the goals of others
- A primary focus is to generate conflict between players, where clashes of incentive cause players to rally behind different goals. Each player must ensure that the project runs smoothly, but should also strive for higher personal rewards. This conflict should serve to create a concept of 'diplomacy' where players feel support or dislike toward others based on their decisions
- Players are only to communicate with one another when prompted explicitly by the game.(Team game sessions)

## Two Approaches

- Round-based: The whole story is played out as a series of 'rounds', where in each round each player in turn makes a certain decision, possibly influenced by decisions made before, each of these decisions ultimately impact how the story plays out and thus the decisions that are presented in the next round.
- Diversity: The whole story is divided into individual and team parts. In the individual story, each player makes choices that only affect their character and the outcome of the overall story. In a team story, all players must make choices after careful discussion. Because these team stories often have a greater impact on the outcome of the story, compared to individual stories
- Fluid: There is little structure to the story, characters are weaved in and out as the story sees fit, such a method allows for maximum versatility of the decision-tree and thus complexity, but this is harder to design well

## Game Objective

- To teach players about the value of ethics in decision-making in industry, where many factors and complexities are at play

## Major Decision/Plot Points

### Game Start

- Each player is randomly assigned a role, and is run through the context specific to each of their roles
- Each player is then run through a tutorial of the game, followed by being presented with a list of goals that the player must achieve in order to personally win the game

...

...

### Game End

- The story plays out to completion, resulting in either a team victory (on-time and cost-effective deployment of a superior plane), or team defeat (crashes, too slow, plane not superior etc.)
- The player receives their personal result, which will likely, but not always, be failure with a team defeat, the idea is that some players will win and some players will lose unless decisions are near-flawless in terms of taking ethics into account in an appropriate and realistic manner

## Roles in the game, Personas and Personal Goals

- Boeing Executive **BE**
- Software Developer **SD**
- Airline Pilot **AP**
- Aeronautical Engineer **AE**
- FAA Official **FAA**

# Characters

## Introduction:

This page mainly describes the relevant character characteristics of the five important characters related to the game being developed.

## Character personas

To learn more information about each characters, please follow the links below:

- Aeronautical engineer
- Airline pilot
- Boeing executive
- Federal Aviation Administration (FAA) official
- Software developer

## Summary

Roles of the game	Boeing executives	Software developer	Airport pilot	Aeronautical Engineer	Federal Aviation Administration
Responsibilities	<ul style="list-style-type: none"> <li>• Maintain the company's reputation</li> <li>• Ensure the interest of the general board</li> </ul>	<ul style="list-style-type: none"> <li>• Develop a pilot-assistant software that can improve the safetyself accomplishment?</li> <li>• Support the familymoney</li> </ul>	<ul style="list-style-type: none"> <li>• Make sure the airplane landed safely</li> </ul>	<ul style="list-style-type: none"> <li>• Make sure the design of airplane is safe enough</li> <li>• Support the family</li> </ul>	Administration work <ul style="list-style-type: none"> <li>• Company functions as normal</li> <li>• Report looks nice</li> </ul>
Reward	<ul style="list-style-type: none"> <li>• Company reputation</li> <li>• More <b>money</b></li> <li>• More <b>market share</b></li> <li>• <b>Stock</b> price</li> </ul>	<ul style="list-style-type: none"> <li>• More <b>salary</b></li> <li>• <b>Promotion</b> (to manager level)</li> </ul>	<ul style="list-style-type: none"> <li>• More <b>salary</b></li> <li>• <b>Promotion</b></li> <li>• Self <b>safety</b> and passengers safety</li> </ul>	<ul style="list-style-type: none"> <li>• More <b>salary</b></li> <li>• <b>Promotion</b></li> </ul>	<ul style="list-style-type: none"> <li>• More <b>salary</b></li> <li>• <b>Promotion</b></li> </ul>
Punishment*	<ul style="list-style-type: none"> <li>• Imprisonment</li> <li>• Lose of the executive position</li> <li>• Loss of company's reputation</li> <li>• Loss in stock price</li> </ul>	<ul style="list-style-type: none"> <li>• Imprisonment</li> <li>• Loss of job</li> <li>• Loss of reputation in the software industry</li> <li>• Loss of money</li> </ul>	<ul style="list-style-type: none"> <li>• Loss of life</li> <li>• Imprisonment</li> <li>• Grounding of aircraft</li> <li>• Loss of job</li> <li>• Loss of money</li> </ul>	<ul style="list-style-type: none"> <li>• Imprisonment</li> <li>• Loss of job</li> <li>• Loss of reputation in the aircraft industry</li> <li>• Loss of money</li> </ul>	<ul style="list-style-type: none"> <li>• Loss of reputation</li> <li>• Imprisonment (if inspector is found to be negligent)</li> <li>• Loss of job</li> <li>• Loss of money</li> </ul>
Interact with others**					

\*perhaps as a twist to the story-line one of these people can lose someone close to them as a result of a plane crash. (or this situation could be a reflection question after the game)

You can also put pictures if it helps.

\*\* who can the role interact with and how

\*\*\*bold part in reward may be regarded as the "attributes"

# Aeronautical engineer

## Name

Tristram Buddy

## Role definition

Gives guidance on the feasibility of designs. Liaises with executive and with workers on the planes. Engineers have their own ethical code of practice. Consider the possibility of whistleblowing.

## Background

Though still relatively inexperienced, the **AE** has worked with Boeing for 3 years now, and has for the most part found him/herself floundering with relatively insignificance maintenance tasks and trivial calculations. As a sole breadwinner for the family, with a spouse, and an infant child to take care of, his/her current income as a Boeing employee is insufficient. Whilst it is enough to survive, the **AE** hopes to work his/her way up the engineering ladder to a more senior position and salary grade; as the **AE** fears the guilt that may be faced in the event he/she is unable to support the family.

With the advent of the 737 MAX project, the **AE** has been given a managerial role in the technical/engineering sector of development, and will work closely with many types of engineers (mechatronics, mechanical, in particular software with the **SD**). The **AE** knows that this is the chance he/she has been seeking, and is eager to seize upon it.

## Responsibilities

The **AE** acts as the project lead for the 737 MAX project and is responsible for the continuous and stable operation of the 737 MAX in the future. For this reason, the **AE** will frequently report to the **BE** about their progress, obstacles and other such intrigues. The **AE** is also in a position to suggest the demotion or promotion of other members in the team (**SD, AP**) based on his/her perception of their alignment to the project.

In addition to this, the **AE** will spearhead the hardware design and implementation of the 737 MAX and take responsibility for any consequences of any damages caused by the project. The **AE** is also to gather reports from the **SD** and **AP** about plane software and risk management, respectively and use these to drive decision-making.

## Outcomes

- As a result of your choices, there are serious hardware flaws in the design of the aircraft, both in the cockpit of the aircraft and in the autopilot. Even parts of the software design were known to other companies. The company decided not to hire you.
- Thanks to your choices, the aircraft can operate appropriately for some time, despite some flaws in the aircraft's design. However, at some point in the future, the aircraft may be involved in an accident due to unknown factors, so the company will still observe your choice afterward
- Thanks to your choice, the aircraft can still be kept in good flying condition with regular maintenance, despite certain defects. At the same time, you have completed the design by the company's requirements for time and cost. The company decides to offer you a higher position.
- Thanks to your choice, the aircraft can be kept in excellent condition for a long time. However, the design cycle and cost exceed the company's affordability limit, so the company will still observe your subsequent choice.

## Goals

- On-time completion of the project, with a superior plane as the output
- Successfully bid for, or obtain naturally, a salary increase
- Be granted a leadership role for the subsequent development project

## Pressures

- The **AE**, in failing to achieve business objectives, is liable to be demoted or dismissed from the project team, leading to differing pay cuts
- The **AE** will frequently face pressure from the **BE** to speed up development, and face opposite pressure from the **SD** and **AP** to minimise risk and maximise quality
- In the event of a crash, if Boeing holds the **AE** liable, the **AE** will lose the game

# Airline pilot

## Name

Montgomery Wincent

## Role definition

Peter is a pilot who flew one of the 737's that did not crash. He has flown for more than 22 years and now his work is more about testing. He was to work closely with the SD and AE and will work under the AE, as the SD will.

## Background

The **AP** is a pilot who has worked for Boeing for the last 22 years, immensely loyal to Boeing and what it stands for, the pilot was disturbed by the news of a new **BE** being appointed, worried that the upheaval of the company might change it for the worse. Having retired from commercial flight in 2008, the **AP** has stepped into a testing role as it reduced the stress of travel and seemed a natural transition into retirement, which was expected to be in around 8 years from then. A relatively uninspiring job, the **AP** was delighted to hear that he/she would work on the 737 MAX project, where the **AP** would get the opportunity to test new and innovative plane designs; and leverage his/her commercial flying experience to comment on designs and software models from a practical perspective.

The **AP** was to work closely with the **SD** and **AE**, and will work under the **AE**, as the **SD** will.

Because Boeing recently held an emergency board of directors and decided to use a group of newcomers to develop Boeing 737MAX series models and compete with Airbus A320 models for market share, Peter was worried that the results of this project could not reach the goal, because Peter did not understand these newcomers, so he wants to get involved as a pilot for many years to evaluate and test aircraft.

## Responsibilities

The **AP**'s primary role will be to test and evaluate the aircraft model designs generated throughout the 737MAX project in the perspective of reality (more practical). As subordinate to the **AE**, the **AP** will frequently report his findings to him, which will primarily focus on risk management as well as some practical suggestions for airplane improvement (cockpit design, software functionality etc.). As the only experienced member of the team, the **BE** expects the **AP** to play an influential role in decision-making and where necessary challenge the **AE**'s leadership.

## Outcomes

- Due to your choice, the cockpit of the aircraft, as well as the autopilot, still has defects that make it unbearable for the pilot. The aircraft is not even capable of being flown by the pilot for long periods, so the company has decided not to hire you.
- Thanks to your choices, the aircraft can operate appropriately for some time, although there are still some drawbacks to automated flight. However, at some point in the future, the aircraft may break free from the pilot's operation, and an accident may occur, so the company will still observe your choice after the fact.
- Thanks to your choices, the aircraft is kept in excellent flying condition through regular maintenance, despite certain defects. At the same time, you have completed the quality inspection of the aircraft by the company's requirements for time and cost. The company has decided to offer you a higher position.
- Thanks to your choice, the aircraft can be kept in excellent condition for a long time. However, the aircraft quality inspection cycle and costs exceed the company's tolerance limit, so the company will still observe your subsequent choice.

## Goals

- Retain current position in the company
- Improve reputation with the company
- See the Boeing 737 MAX outcompete the Airbus A380

Whilst finances aren't as big a concern for the **AP** as for others, the **AP** as a loyal Boeing employee does seek recognition from the company of some form, and wishes to see the company succeed.

## Pressures

- The **BE** has placed high pressure on the **AP** to be an influential member of the team, failing to take a leadership role at critical points will open up risks of demotion.
- The **AP** desires to become a critical part of Boeing's success, failure to be looked up favourably by Boeing leadership (**AE**, **BE**), or failing to produce good outcomes for Boeing will be seen as a failure.

Peter's pressure comes from Boeing's trust and recognition for himself for many years. If he can't complete the task in the 737MAX project or cause incalculable losses in the future, Peter will lose the company's trust in him. In serious cases, he may be investigated with legal liability.

# Boeing executive

## Name

Vespasiano Ilu

## Role Definition

Overseeing the plan to catch up to Airbus. Motivated, in part, by financial and reputational considerations. What are the duties of company executives? Who do they owe duties to? Shareholders? Corporation? Public? Environment? Etc.

## Background

The game sets itself at the beginning of 2011. A month earlier, Airbus, the main competitor of Boeing launched the Airbus A320 family that delivered fuel burn and operating efficiency outcomes far superior to existing Boeing models in the market (that is, it uses less oil to run longer distances), the resultant orders for these new planes from many established Boeing partners such as American Airlines has led to substantial shareholder pressure and reduction in value of Boeing stock. The board of directors, dissatisfied with these results has taken the drastic action of firing the existing executive in hopes of finding a new one who can turn their dire situation around.

The new **BE** is introduced, with the assignment of leading a new development effort for a new airplane family, the Boeing 737 MAX. The board of directors, shareholders, and existing management have made it very clear what the **BE** is there to accomplish, a rapid deployment of a superior plane that would reclaim lost market share from Airbus.

## Responsibilities

The **BE** is in charge of the whole 737 MAX project, as well as the other operations occurring within the Boeing industry. One of the primary challenges as **Bi** will be to allocate focus appropriately, to ensure other existing plane models do not experience problems that interfere with, or place undue pressure on the 737 MAX. The **BE** will also receive reports from the **AE** about the progress of the project (these may or may not be falsified), which will influence executive decisions that will influence the project from above.

The **BE** will also be responsible for appropriate public relations (PR), handling any interviews or press releases.

## Outcomes

- Due to your previous decision, all the employees of the company, have expressed their dissatisfaction with you, and the environmentalists have been putting pressure on the company all the time. Under these circumstances, the company has decided to dismiss you.
- As a result of your previous decision, a large portion of the company's workforce is still unhappy with you, and the company has taken considerable risks in R&D. However, your choice has kept the development schedule and costs within reasonable limits, so the company has decided to observe your subsequent decision
- Due to your previous decision, a small number of employees in the company are still unhappy with you. However, your choice has kept the R&D schedule and costs within reasonable limits. Therefore, the company has decided to offer you a higher position.
- Thanks to your previous decisions, all company employees are loyal to you, and environmentalists have reported your deeds to the media. However, your decision has significantly increased the company's R&D progress and costs, and the company has decided to observe your subsequent decision.

## Goals

- Improve reputation with the board of directors
- Improve reputation with corporate employees (**AE, AP, SD**)
- Improve overall corporate reputation with public

The main motivation for BE Pat comes from the trust of Boeing's board meeting and greater recognition of its potential for improvement. Because Pat was elected through the board meeting. Secondly, the motivation of BE Pat also comes from the return on value, that is, to obtain a higher salary.

## Pressures

- The board of directors expects: on-time delivery of a superior plane, strong market share outcomes upon plane release, good financial management during the project
- The employees expect: fair treatment, a feeling of empowerment, a minimisation of pressure placed on the project from above
- The public expect: good ethical practice, a focus on safety and quality of the plane, good PR

A sufficiently low reputation with the public or board of directors will lead to being fired (game over), a sufficiently low reputation with the employees merely suggests dissatisfaction with executive decisions, but will almost certainly lead to eventual failure with the other two parties.

# Federal Aviation Administration (FAA) official

## Name

FAA Team

## Role Definition

Federal Aviation Administration (FAA) official: In charge of liaising with Boeing. Responsible for the safety of planes in the USA.

## Background

The Federal Aviation Administration always pays attention to the safety of civil aviation, and has enjoyed a high reputation domestically and internationally for many years. At the same time, he has also witnessed the development of a number of airlines, such as Airbus, American Airlines and Boeing. In recent years, with the development of aerospace companies and the introduction of new models by various companies' products, civil aviation aircraft crashes that have focused too much on economic benefits and neglected flight safety have occurred one after another. The department decided to conduct a close investigation of the crashed airliner and to conduct more detailed management in the field of civil aviation.

## Responsibilities

1. In charge of liaising with Boeing
2. Standardize the distribution of air navigation facilities and aviation safety inspection standardsIf the aircraft crashes in the future, it is necessary to re-investigate the safety of the Boeing 737MAX model and take decisions to remove the aircraft or maintain flight.
3. The issue, suspend or revoke a flight licenseIf the aircraft crashes in the future and is found to be due to improper pilot operations or other related responsibilities, perform this step.

## Outcomes

- Due to your choices, there were many errors in the aircraft that went undetected, even some obvious ones. This is something that has never happened in your company. Therefore, the company decided not to hire you anymore.
- As a result of your selection, the aircraft may operate appropriately for some time, although the aircraft still has undetected defects. However, at some point in the future, a hidden defect, or even some irreparable defect, may occur in the aircraft, so the company will still be watching your selection after the fact.
- Thanks to your choice, all defects are checked before takeoff, and you are assured that the aircraft will fly in good flying condition for a long time. At the same time, you completed a quality inspection of the aircraft by the company's requirements for time and cost. The company decides to offer you a higher position.
- As a result of your selection, the aircraft is inspected for all defects and even some hidden defects. However, the period and cost of aircraft quality inspection exceed the company's tolerance limit, so the company will still observe your subsequent selection.

## Goals

Ensure the stable order of civil aviation, reduce the possibility of aircraft crashes, and promote the development of civil aviation.

## Motivations

Because the FAA team has a significant historical responsibility, the FAA team's motivation is the strong social responsibility and the passion for serving the people. At the same time, if the FAA team can guarantee the safety of people's lives and the well-being of aviation, the Federal Aviation Administration will also win the trust of more people.

## Pressures

If FAA team cannot properly handle the civil aviation incident, it will lose people's trust in the Federal Aviation Administration and cause considerable public opinion pressure and also have a bad reputation (in severe cases, internal processing may be conducted)

# Software developer

## Name

Tomasz Bartel

## Role Definition

Develop the MCAS software under directions from engineers and executive. Consider Codes of practice and ethics for computer programmers etc.

## Background

The **SD** is a recent university graduate of the class of 2010, having completed an graduate degree in Software Engineering at a prestigious Ivy League university in the United States. For a long time, the **SD** has been very passionate about the complexities of air travel and how software plays a role. For the **SD**, Boeing is the dream job. In 2009, the **SD** completed a summer internship at the local Boeing office working relatively menial software tasks and group assignments, and in late 2010 the **SD** survived a rigorous and highly selective application process for a graduate role at Boeing headquarters in Chicago.

Having recently arrived at Chicago, with little to name, the **SD** was informed by a manager that he/she was to work with a team of other, relatively senior developers on a new, highly confidential Boeing 737 MAX project; an extremely lucrative project that placed large responsibility on the **SD**, which in turn gave a once-in-a-lifetime opportunity for a rapid rise through the company ranks.

## Responsibilities

The **SD** is responsible for the creation of software that handles the following processes: functionality of the engine, responses to certain situations the plane finds itself in, auto-pilot etc. As subordinate to the **AE**, the **SD** is to report to the **AE** on behalf of the software team and accurately reflect the sentiments of the team. These reports will feature progress reports on developed software, any testing requirements, as well as discussing the segregation of priorities (documentation, coding, design etc.).

The **SD** will take responsibility for the quality of software on the 737 MAX, and is expected to frequently liaise with the **AP** to ensure software is of the highest standard, which will likely lead to delays due to the **AP's** demands.

## Outcomes

- Due to your choice, there are many software errors in the design of the aircraft, and this error will even change the flight path without authorization or affect the pilot's operation when the aircraft is in flight. So, the company decided not to hire you anymore.
- As a result of your choices, the aircraft may operate properly for some time, even though there are still software errors in the aircraft's design. However, at some point in the future, the pilot may make an operational error that causes the aircraft to crash because the UI interface does not operate as it was conventionally designed, so the company will still be concerned about your choice after the fact
- Thanks to your choice, the aircraft has an excellent operating system and a UI interface, and you can be sure that the aircraft will fly in good flying conditions for a long time. At the same time, you have completed the software design of the aircraft by the company's requirements for time and cost. The company has decided to offer you a higher position.
- Thanks to your choice, the aircraft has an excellent operating system, and even in the future, many companies will borrow such an operating system and the UI interface. However, the design cycle and cost of the aircraft software exceed the company's tolerance limit, so the company will still observe your subsequent choice.

## Goals

- Earn a promotion
- Improve reputation with the **AE**
- On-time delivery of software

The main motivation of SD Michelle is the promotion of personal positions and experience in participating in major development projects. Based on the perfect completion of the development project and improving the company's ranking, it also improved its own position competitiveness.

## Pressures

- The **SD** is under pressure from the **AE** to reach time goals, so that software and hardware development operates in sync, failure to do so will incur negative consequences
- The **SD** will need to balance the wishes of the software team with those of the **AE**, **AP**, failure to do so will lead to loss of motivation of the software team, or loss of cohesion within the team, both likely leading to project failure

# Plot

This page mainly describes the key plots of game process.

## Roles in the game, Personas and Personal Goals

- Boeing Executive **BE**
- Software Developer **SD**
- Airline Pilot **AP**
- Aeronautical Engineer **AE**
- FAA Official **FAA**

## Key decision points and roles involved

Decision ID	Decision	Roles involved
1	Develop a new model or modify current model	<b>BE AE</b>
2	Using a larger but dangerous engine or a small safer engine	<b>BE AE</b>
3	Consider pilots' responses to MCAS function or not	<b>BE AE SD AP</b>
4	Ground new model aircrafts after the first crash	<b>BE AE SD AP FAA</b>
5	Investigate the cause & Disclose investigation result to the public	<b>BE AE SD AP FAA</b>
6	Using highly trained test pilots or different experience level pilots for new model aircraft training	<b>BE AP FAA</b>

## Key Factors that impact outcomes

- Available money
- Quality of Project
- Time
- Possibility of Crash (risk)
- Individual score for each character

## Notes

- The 2nd crash is only related to the risk of crashing, however, the 1st crash will always happen no matter how low the possibility of crashing is.
- The investigation results have no impact on the 2nd crash.
- Some choices cannot be made if the available money or the time is 0.

## Outcomes for Boeing

- **Outcome 1:** This is a key winning result. Boeing built a successful model that outperformed Airbus' model in terms of payload and fuel efficiency, and launched it before Airbus released its model. This brought Boeing a large number of orders for this model. All members of the project team received significant promotions and accolades from the company.
- **Outcome 2:** Boeing built a successful model that outperformed the Airbus model in terms of payload and fuel efficiency. The aircraft was relatively successful and competitive in the marketplace, so the program was called a success by the executive team. Unfortunately its release was later than Airbus' release of its new model, which reduced Boeing's stock value and reduced orders for the new model. Overall, however, the aircraft's performance offset these problems to a satisfactory degree. Well done!
- **Outcome 3:** The aircraft was relatively successful and competitive in the marketplace, and as a result, the program was described as a small success by the executive team. While some employees may have lost their jobs due to their own personal failures, the program did maintain a competitive edge, despite problems with time or budget that prevented the aircraft from dominating more of the market.
- **Outcome 4:** Regardless of the success or otherwise of this aircraft, unfortunately, the development costs were too high, which allowed competitors to establish an advantage outside the scope of the program. Boeing decided to immediately stop manufacturing all aircraft of this type. The decision needed to be made with the need to deliver the program within budget and within a reasonable timeframe.
- **Outcome 5:** While the aircraft was initially a success, a number of undetected problems caused some disastrous results. Three months after the plane's first commercial flight, a crash occurred in which everyone on board died. Two months later, due to the company's inaction, a second crash occurred, again resulting in the deaths of everyone on board. The legal investigation was thorough and found that the company had been extremely negligent in designing the aircraft, often cutting corners and prioritizing profit and efficiency over the well-being and safety of its customers. So ultimately the project was a complete failure.

# Decision Tree

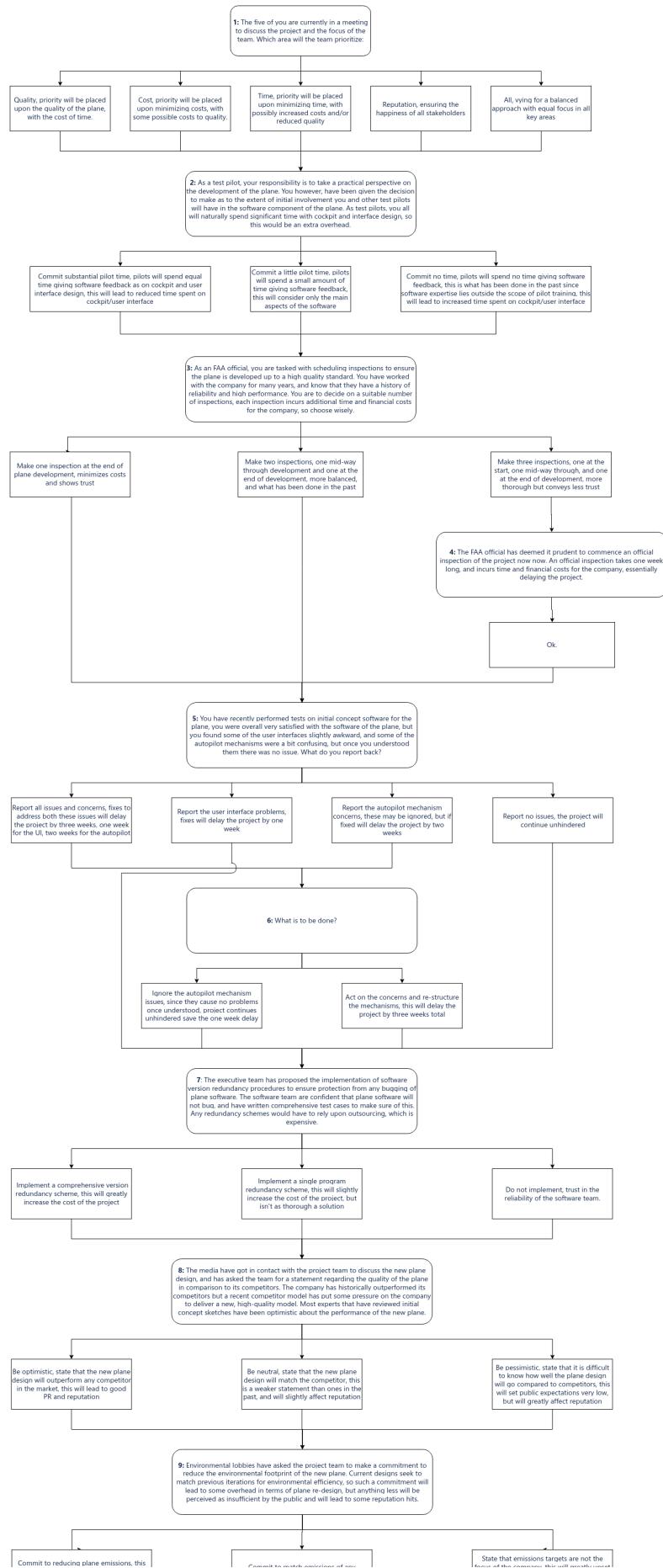
The [initial decision tree](#) consists of 27 questions. The game would not be end until every player answeres all questions. Moreover, there is no limit to the discussion time of questions, which lead to the game is extremely long with over 30 minutes per round.

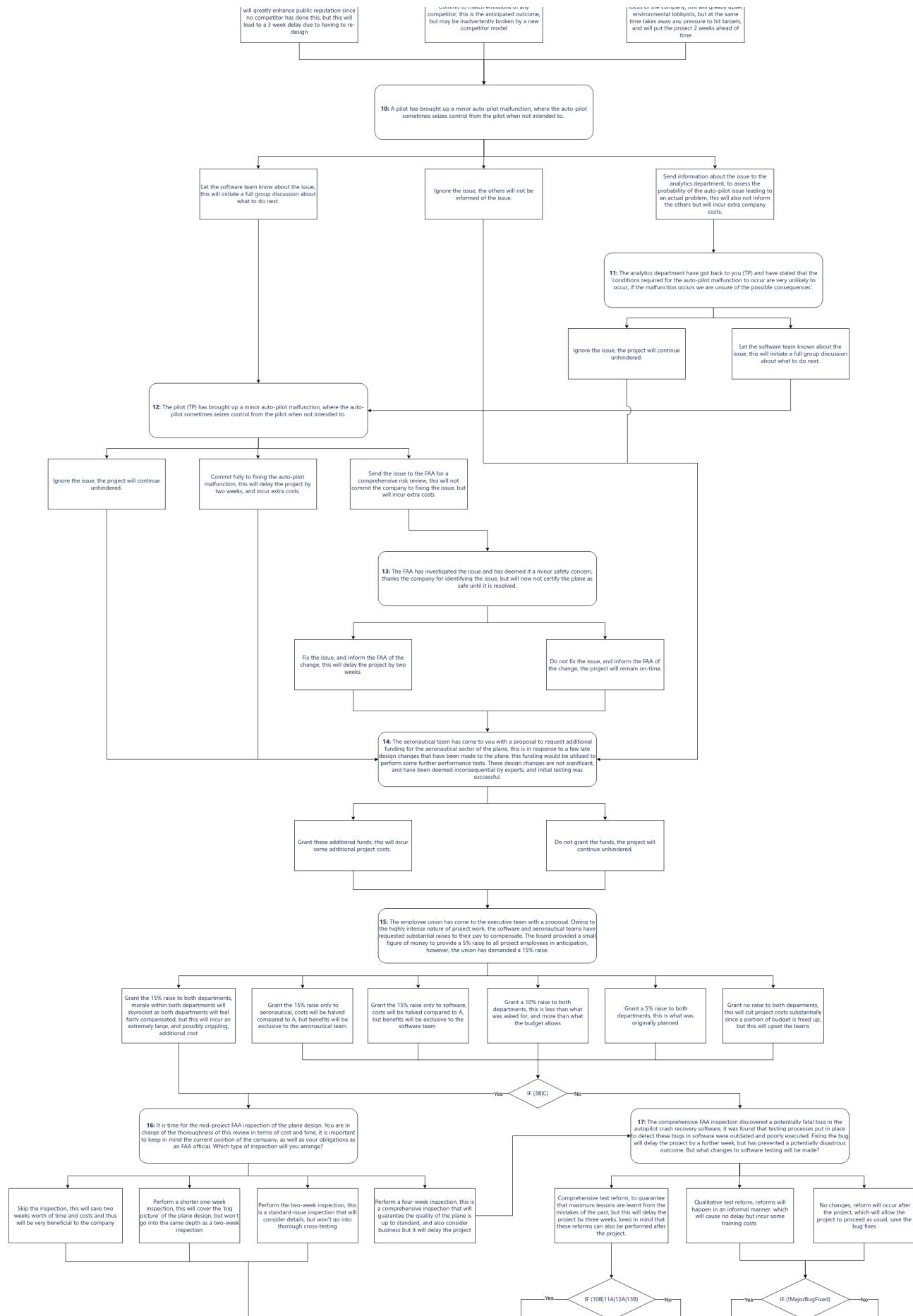
The simplification process of the decision tree is shown below:

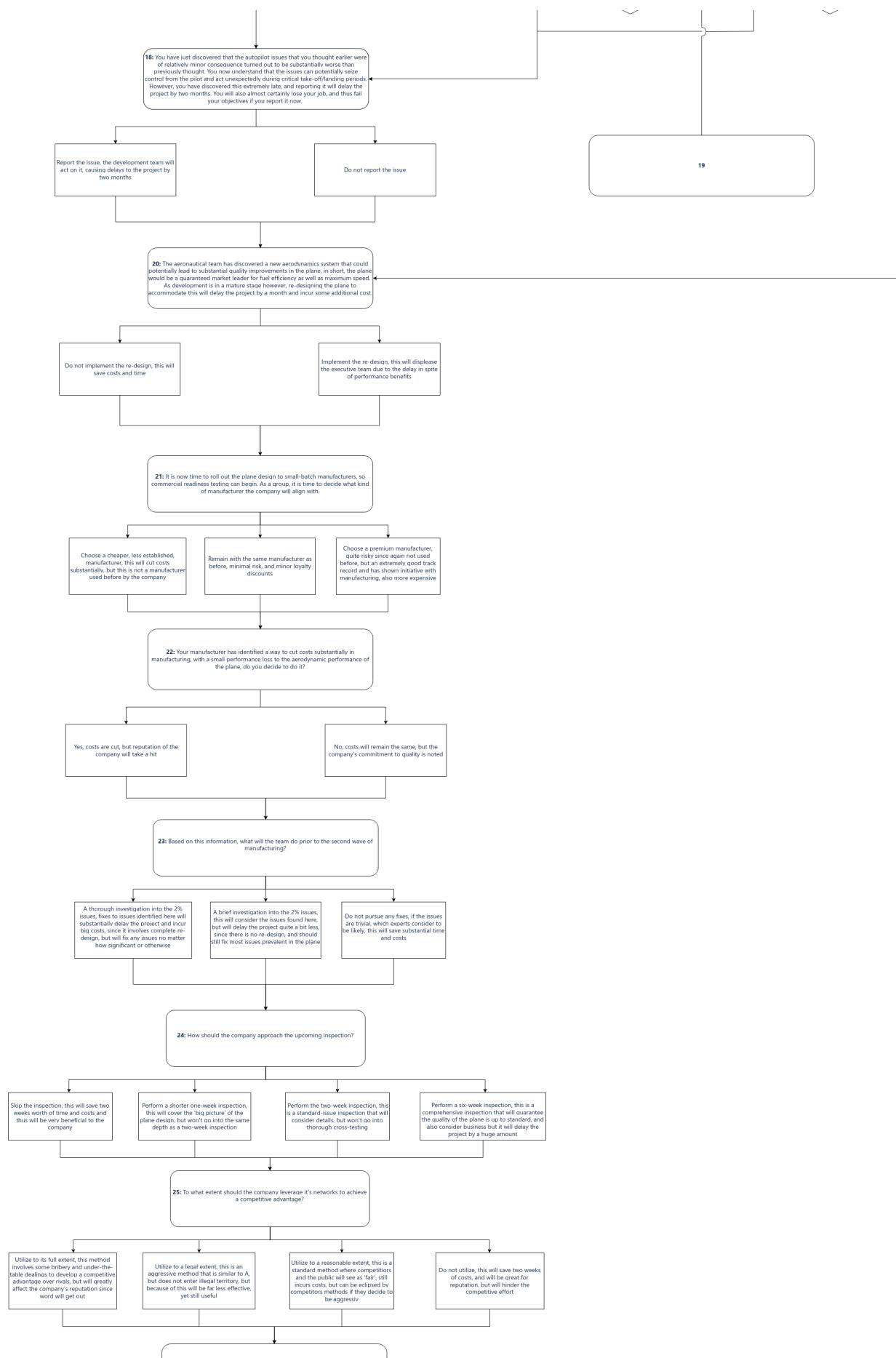
Simplification methods	Question Number	Reasons
Delete questions	1, 4	Question 1 is about a tip of the game to help bring players into the situation, which is not essential to the result of the game.  Question 4 only has one choice "OK", which has no effect on game progression.
Personal answered questions  (1 minute per question)	Boeing Executive: 8, 9, 14, 15  Aeronautical Engineer: 2, 5-6-7  Software Developer: 2, 5-6-7, 13  FAA Official: 3, 5, 13, 16, 17  Boeing Pilot: 2, 10-11-12, 18	The questions that should be answered by each player depend on how it is related to the role. In the context of a question, only someone who is the responsible for this event would answer this question.  Question "5-6-7" and "10-11-12" means these questions are related. For example, different answer of question 5 would lead to question 6 or question 7. While no matter which choice is chosen, the question would finally come to question 7.  Each question must be answeres within 30 seconds.
Group discussion questions  (2 minutes per question)	20, 21, 22, 23, 24, 25, 26, 27	These general questions need to be answeres after group discussion.  Each question must be answeres within 2 minutes.

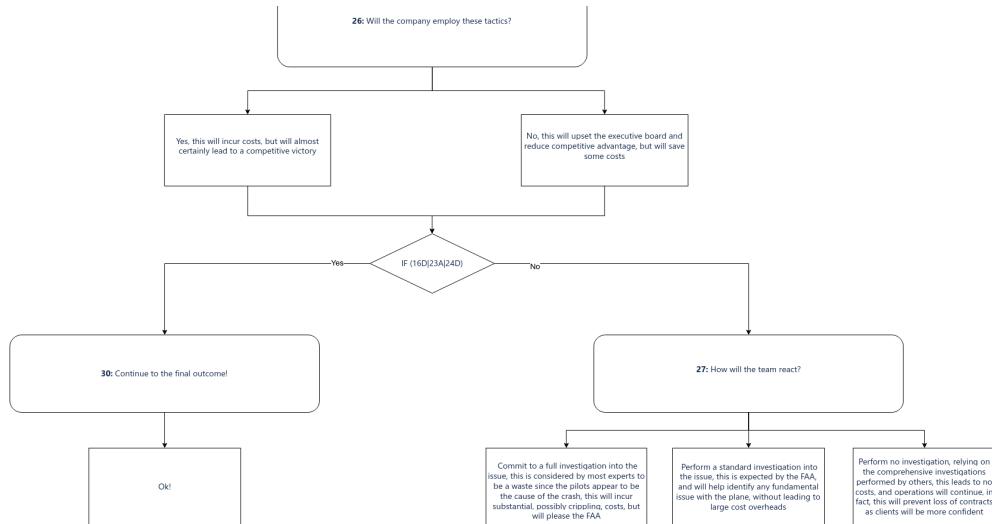
[Current decision tree](#) consists of 25 questions. Each players are requested to answer 4 to 5 personal questions and 8 group discussion questions. Play time is limited to 20 minutes.

## Decision Tree (version 1)



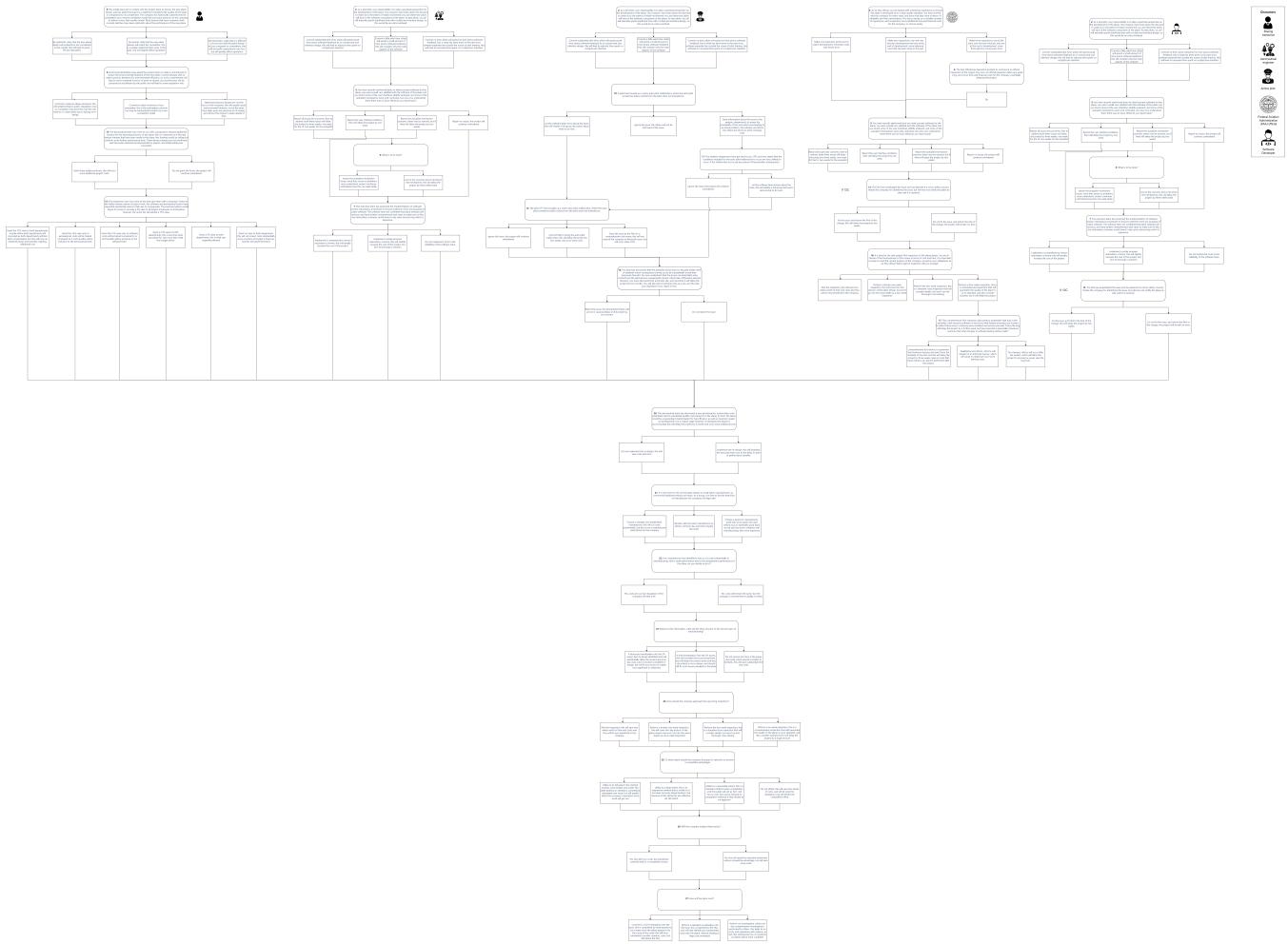






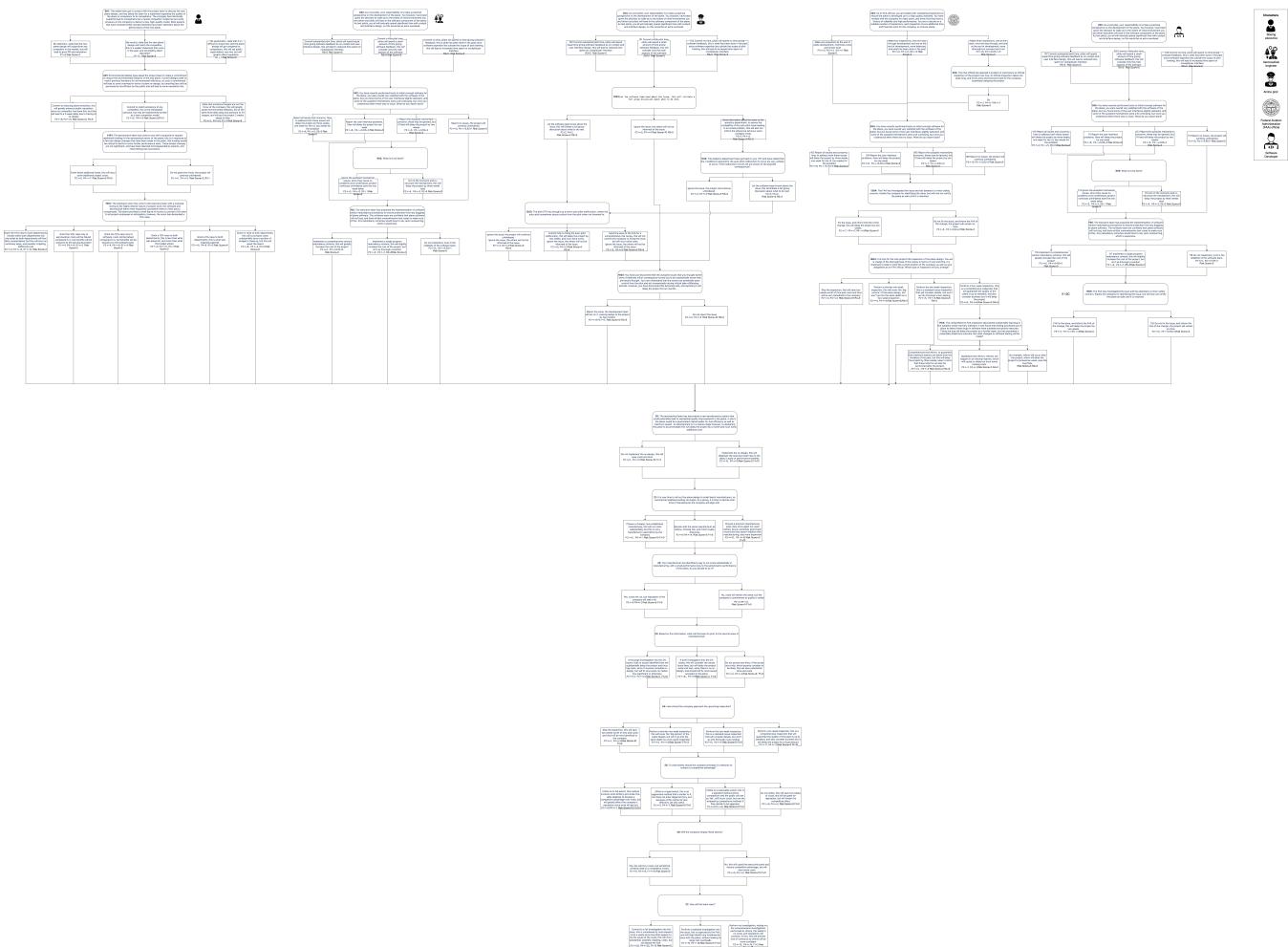
# Decision Tree (version 2)

Separate the questions into role questions and general questions to reduce the total playing time.



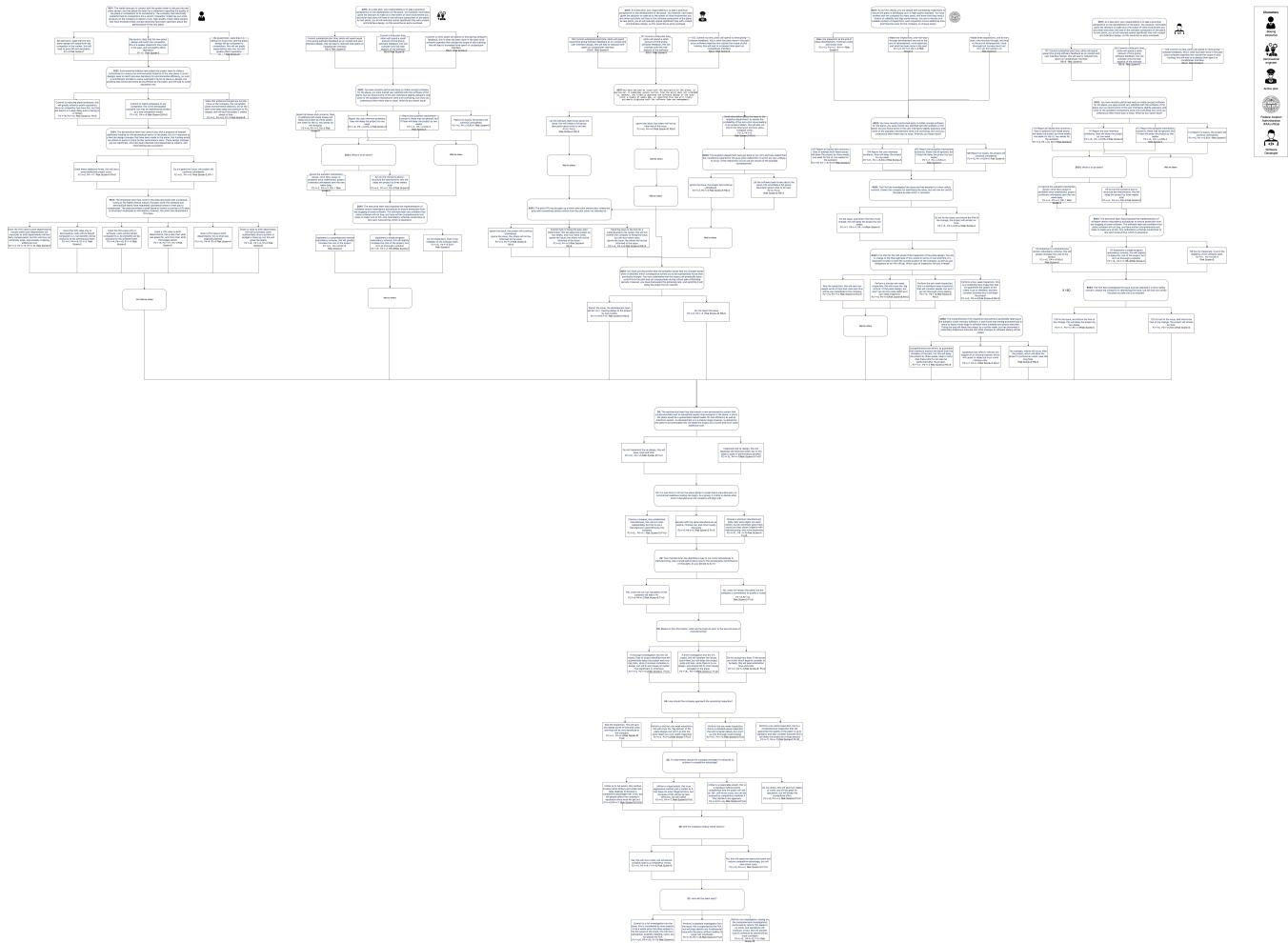
# Decision Tree (version 3)

Refine the decision tree and change the question id.



# Decision Tree (version 4)

Add a page to wait for other characters to answer so that each character has the same total number of questions to answer.



# Prototypes

## Introduction:

The Prototypes page displays the design of the paper prototype and digital prototype based on preliminary requirements for further development. There are two subsections:

### Paper Prototype

This page presents the paper prototype, which includes how the game executes and what contents it remains.

### Digital Prototype

This page display the static prototype only contains the simple click operations to show the essential functions of the game.

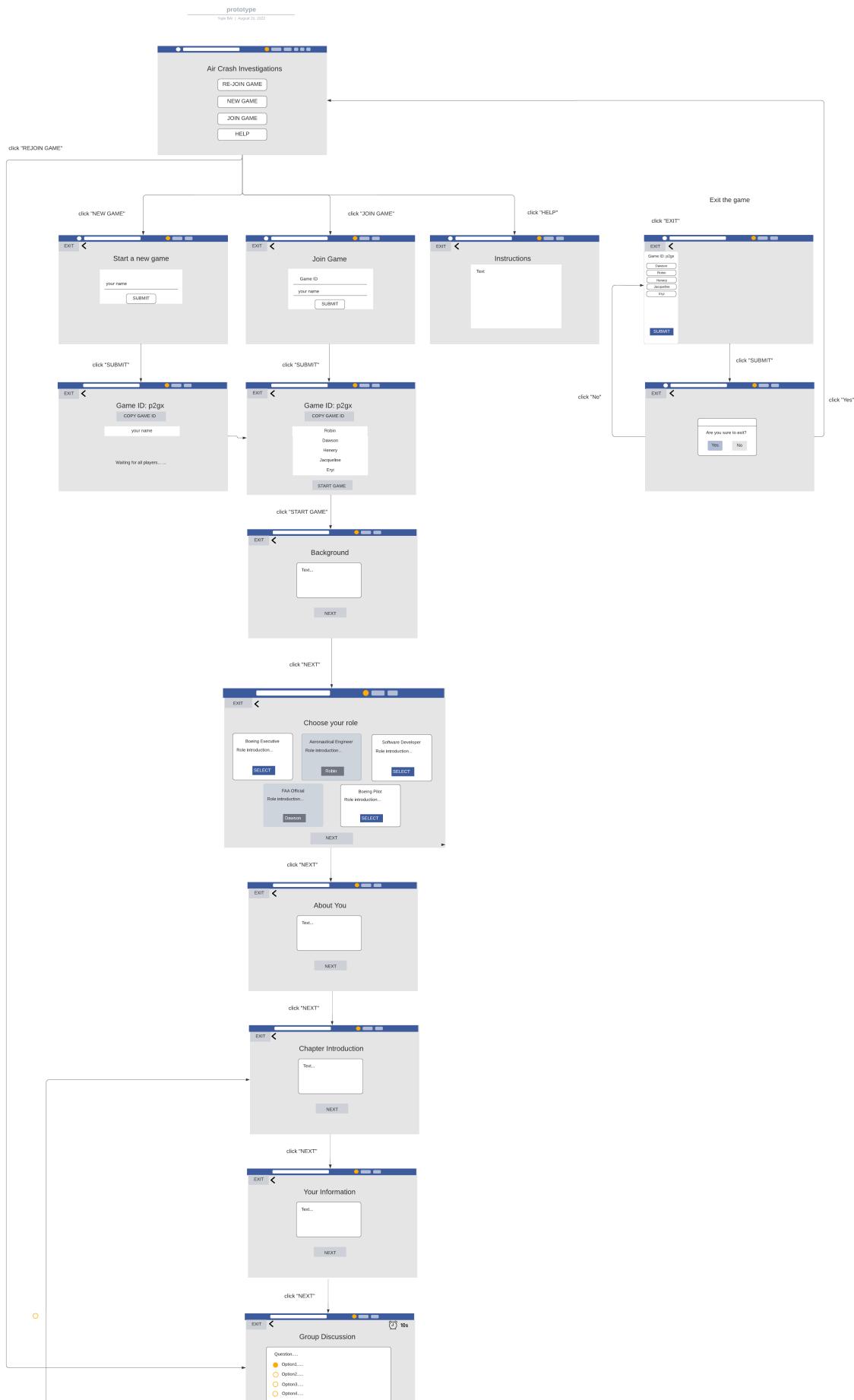
To learn more about the subsections, please follow the links below:

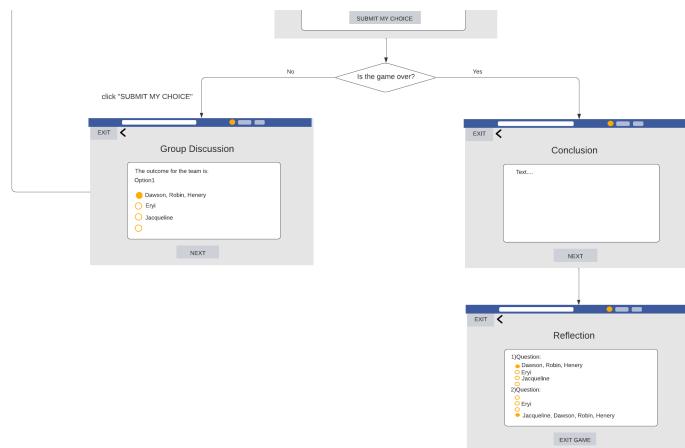
- [Paper prototype](#)
- [Digital prototype](#)

# Digital prototype

Digital prototype available at: <https://zc5znz.axshare.com>

# Paper prototype





# Architecture

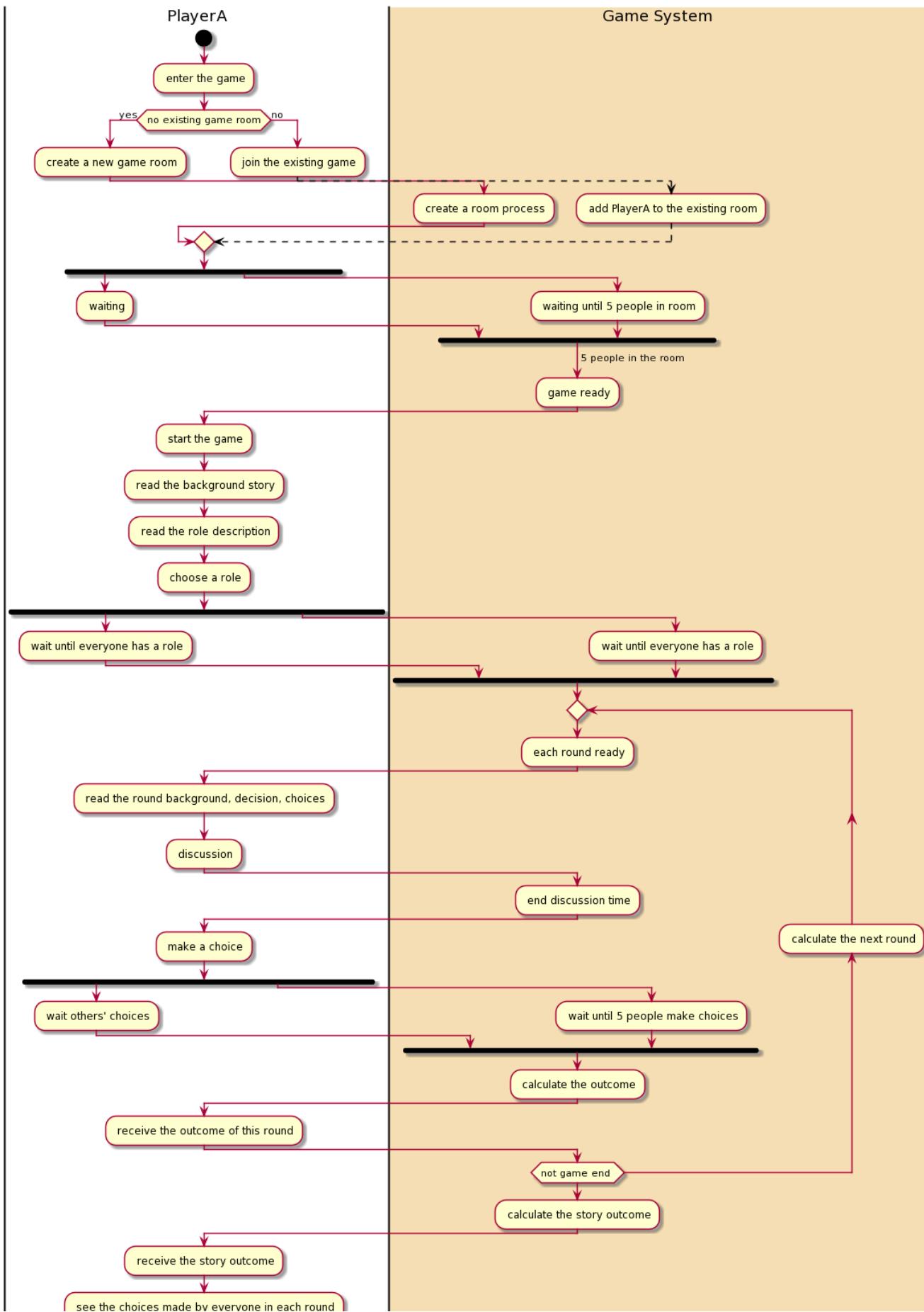
# Activity Diagram

Description:

The activity diagram is used to describe the activities that will be performed by the user (player) and the system (the game). It also describes the functionality that the system provides to the users.

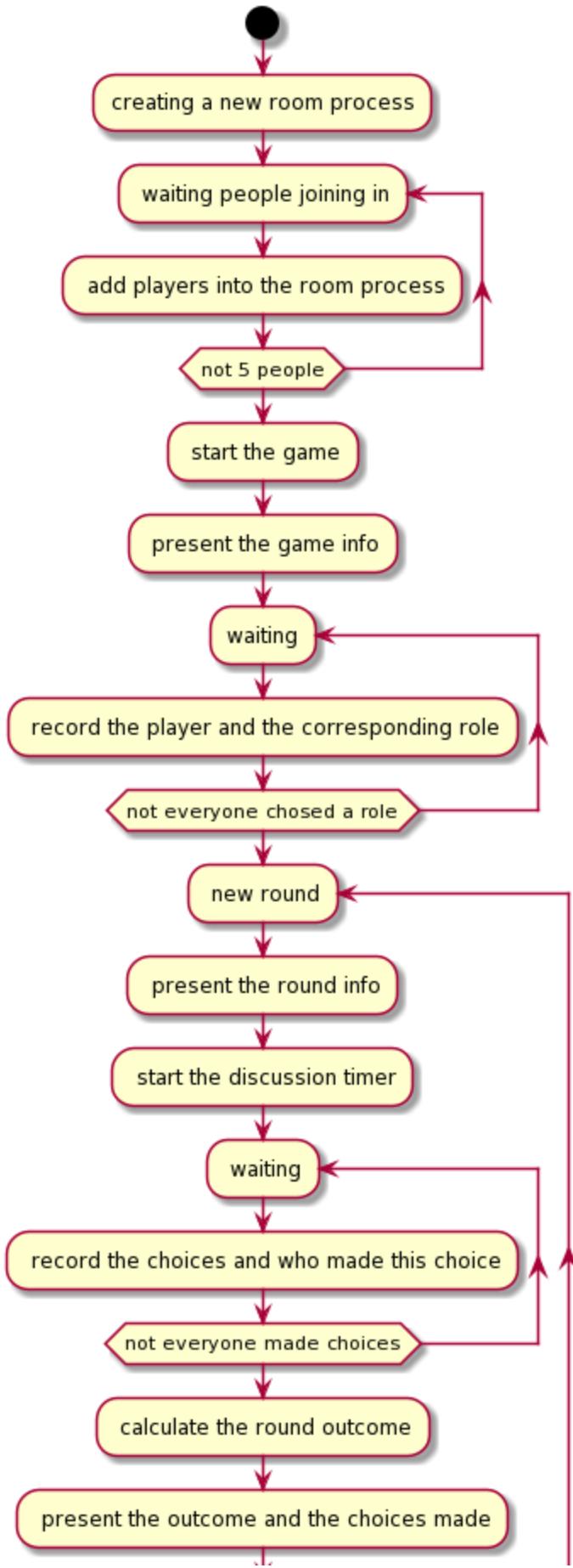
The activity is an abstraction of the behaviour excluding the details of interactions. For the interaction details between the frontend, the backend server and the database, please refer to [Sequence Diagram](#).

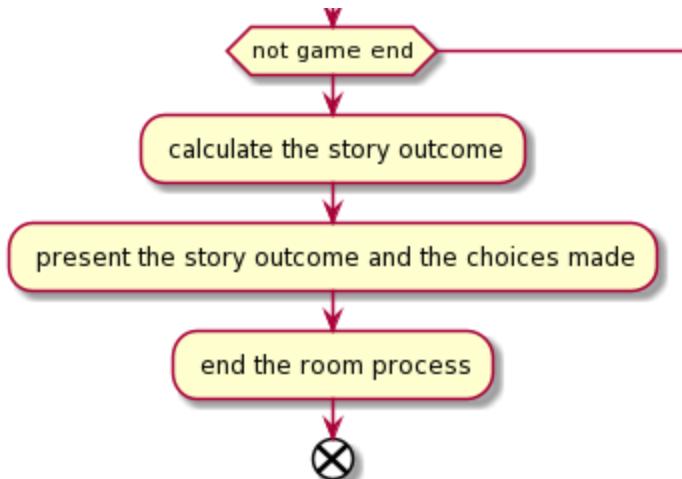
Player and Game System Activity Diagram





## Game System Activity Diagram





# API Specification

This page documents the API used to connect the front- and back-ends.

ID	Implemented at Back-End	Implemented at Front-End	Method	Endpoint	Description	Parameters	Responses
ID	Implemented at Back-End	Implemented at Front-End	Method	Endpoint	Description	Parameters	Responses
1	✓	✓	POST	/play/new	A user wants to start a new game	<ul style="list-style-type: none"> <li>body           <ul style="list-style-type: none"> <li>{ "name": "John", "duration": 30}</li> </ul> </li> <li>Note: if user does <b>not</b> select that they want a timed game, duration = 0</li> </ul>	<ul style="list-style-type: none"> <li>200</li> <li>400</li> </ul>
2	✓	✓	POST	/play/{gameID}/join	A user wants to join an existing game	<ul style="list-style-type: none"> <li>gameID           <ul style="list-style-type: none"> <li>(path, string)</li> </ul> </li> <li>body           <ul style="list-style-type: none"> <li>{ "name": "John"}</li> </ul> </li> </ul>	<ul style="list-style-type: none"> <li>200</li> <li>400</li> <li>e.g. invalid Game ID supplied</li> </ul>
3	✓	✓	GET	/play/{gameID}/wait	Player is waiting for all players to join the game	<ul style="list-style-type: none"> <li>gameID           <ul style="list-style-type: none"> <li>(path, string)</li> </ul> </li> </ul>	<ul style="list-style-type: none"> <li>200           <ul style="list-style-type: none"> <li>{"players": [             {"id": 0, "name": "John"}, {"id": 1, "name": "Georgia"}, {"id": 2, "name": "Emily"}], "ready": false}</li> </ul> </li> <li>400           <ul style="list-style-type: none"> <li>e.g. invalid Game ID</li> </ul> </li> <li>ready = true if all 5 players are in and the game is ready to start</li> </ul>

4	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	GET	/play/{gameID}/status	Retrieve information about the current game e.g. when viewing the in-game menu	• gameID • (path, string)	<ul style="list-style-type: none"> <li>• 200           <ul style="list-style-type: none"> <li>• { "status": "inProgress", "players": [ { "id": 0, "name": "John", "role": "Aeronautical Engineer"}, { "id": 1, "name": "Georgia", "role": "Boeing Executive"}, { "id": 2, "name": "Emily", "role": "Pilot"}, { "id": 3, "name": "Ming", "role": "FAA Official"}, { "id": 4, "name": "Steve", "role": "Software Developer"} ] }</li> </ul> </li> <li>• subject to change</li> <li>• status of the game can be:           <ul style="list-style-type: none"> <li>• waiting - waiting for all players to join the game</li> <li>• ready - all 5 players have joined, ready for players to press "Start game"</li> <li>• inProgress</li> <li>• completed</li> </ul> </li> <li>• 400           <ul style="list-style-type: none"> <li>• e.g. invalid Game ID</li> </ul> </li> </ul>
5	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	GET	/play/{gameID}/background	Get the "Background" information shown to all players	• gameID • (path, string)	<ul style="list-style-type: none"> <li>• 200           <ul style="list-style-type: none"> <li>• the content (perhaps as plain text including &lt;p&gt; tags? or maybe as JSON with .)</li> </ul> </li> <li>• 400</li> </ul>
6	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	GET	/play/{gameID}/roles	Get the "role" information shown to all players	• gameID • (path, string)	<ul style="list-style-type: none"> <li>• 200           <ul style="list-style-type: none"> <li>• { "roles": [ { "title": "Boeing Executive", "description": "Oversees the whole 737 MAX project, as well as the other operations occurring within the Boeing company."}, { "title": "Aeronautical Engineer", "description": "Acts as the project lead for the 737 MAX project, and is responsible for the continuous and stable operation of the 737 MAX in the future." } etc ] }</li> </ul> </li> </ul>
7	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	GET	/play/{gameID}/choose-role	Players select their roles. Gets all the roles, whether the role is available to be selected (and if not, who is it assigned to)	• gameID • (path, string)	<ul style="list-style-type: none"> <li>• 200           <ul style="list-style-type: none"> <li>• { "roles": [ { "roleID": 0, "role": "Boeing Executive", "isAvailable": true}, { "roleID": 1, "role": "Aeronautical Engineer", "isAvailable": true}, { "roleID": 2, "role": "Software Developer", "isAvailable": true}, { "roleID": 3, "role": "Pilot", "isAvailable": false, "player": "Ming"}, { "roleID": 4, "role": "FAA Official", "isAvailable": false, "player": "Georgia"} ] }</li> </ul> </li> <li>• 400</li> </ul>

8			POST	/play/{gameID}/choose-role	Player selects their role	<ul style="list-style-type: none"> <li>• gameID           <ul style="list-style-type: none"> <li>• (path, string)</li> </ul> </li> <li>• roleId           <ul style="list-style-type: none"> <li>• role ID (integer between 0 and 4, as specified in the previous request)</li> </ul> </li> </ul>	<ul style="list-style-type: none"> <li>• 200           <ul style="list-style-type: none"> <li>• player successfully assigned to role</li> </ul> </li> <li>• 400           <ul style="list-style-type: none"> <li>• e.g. invalid game ID</li> </ul> </li> <li>• 403           <ul style="list-style-type: none"> <li>• player is not assigned to the role e.g. the role is no longer available</li> </ul> </li> </ul>
9			GET	/role/introduction/	Get the "Role Introduction" content, i.e. a page of text showed to the player at the start of the game, based on the role they chose	<ul style="list-style-type: none"> <li>• body           <ul style="list-style-type: none"> <li>• { "role ID": 0 }</li> </ul> </li> </ul>	<ul style="list-style-type: none"> <li>• 200           <ul style="list-style-type: none"> <li>• the content</li> </ul> </li> <li>• 400</li> </ul>

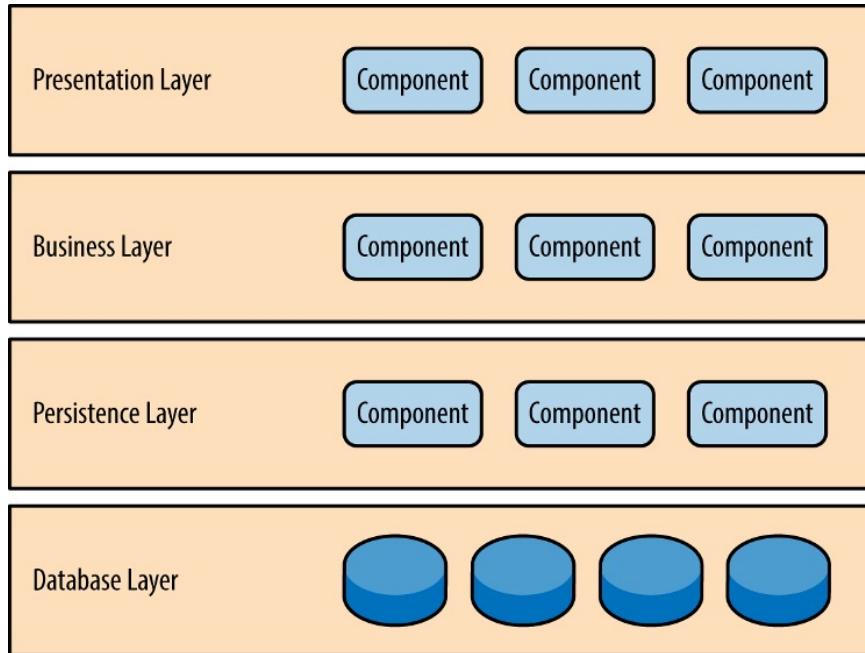
10		✓	GET	/play/make-decision/:gameID	Get all the decisions during the game in reflection phase.	<ul style="list-style-type: none"> <li>• gameID           <ul style="list-style-type: none"> <li>• (path, string)</li> </ul> </li> </ul>	<ul style="list-style-type: none"> <li>• 200           <ul style="list-style-type: none"> <li>• {               <pre>{"reflectionList": [                 {"questionID": 1,                   "questionText": "Do we need to halt development in order to further investigate the AoA sensor?"}                 ],                 "options": [                   {"id": 1,                     "text": "Halt development",                     "users": [{"name": "hh", "role": "Aeronautical Engineer"}, {"name": "Emily", "role": "Software Developer"}]                   },                   {"id": 2,                     "text": "Continue Development",                     "users": [{"name": "Steve", "role": "Executive"}, {"name": "Ming", "role": "Pilot"}, {"name": "Georgia", "role": "FAA Official"}]                   }                 ],                 "result": 1               }</pre> </li> <li>• {               <pre>{"questionID": 1,                   "questionText": "should we build an expensive new software module to help avoid potential issues with the sensor?"}                 ,                 "options": [                   {"id": 1,                     "text": "Develop new module",                     "users": [{"name": "hh", "role": "Aeronautical Engineer"}]                   },                   {"id": 2,                     "text": "Do not develop new module",                     "users": [{"name": "Steve", "role": "Executive"}, {"name": "Ming", "role": "Pilot"}, {"name": "Georgia", "role": "FAA Official"}, {"name": "Emily", "role": "Software Developer"}]                   }                 ],                 "result": 2               }</pre> </li> </ul> </li> <li>• 400           <ul style="list-style-type: none"> <li>• e.g. invalid game ID</li> </ul> </li> </ul>
----	--	---	-----	-----------------------------	--	--	---

11			GET	/play/{gameID}/round	Get all the information that the player needs to see for the current round, i.e. the information shown under "Chapter One" in the prototype: "Background", "Role Background" and "Decision"	<ul style="list-style-type: none"> <li>• gameID           <ul style="list-style-type: none"> <li>• (path, string)</li> </ul> </li> <li>• body           <ul style="list-style-type: none"> <li>• {"roleID": 1}</li> </ul> </li> </ul>	<ul style="list-style-type: none"> <li>• 200           <ul style="list-style-type: none"> <li>• { "questionID": 1, "background": [ "This is the first paragraph of text. Work is progressing on-time for the design of the Boeing 737 MAX; however it is critical to keep up the current pace to meet the deadline.", "This is another paragraph of text. The design team are currently focussed on perfecting the MCAS system to help manoeuvre the stabiliser nose of the aeroplane to reduce the risk of stalling." ], "roleinfo": [ "You discover that there may be an issue with the angle-of-attack (AoA) sensor used in the aeroplane design.", "The sensor usually is reliable, but in certain conditions during testing it appeared to report erroneous data.", "Further investigation could be performed in order to figure out the cause and severity of this issue." ], "question": "Should we build an expensive new software module to help avoid potential issues with the sensor?", "options": [ {"id": 1, "text": "Develop new module"}, {"id": 2, "text": "Do not develop new module"} ] }</li> </ul> </li> <li>• 400           <ul style="list-style-type: none"> <li>• e.g. invalid game ID</li> </ul> </li> </ul>
12			POST	/play/make-decision/:gameID	Player enters their selected choice for the question of the round	<ul style="list-style-type: none"> <li>• gameID           <ul style="list-style-type: none"> <li>• (path, string)</li> </ul> </li> <li>• body           <ul style="list-style-type: none"> <li>• {"questionID": 1, "optionID": 2, "roleID": 1, "roundID": 2, "oldID": 1}</li> </ul> </li> </ul>	<ul style="list-style-type: none"> <li>• 200 - choice was valid and recorded correctly           <ul style="list-style-type: none"> <li>• { "questionID": 1, "optionID": 2, "roleID": 1 }</li> </ul> </li> <li>• 400           <ul style="list-style-type: none"> <li>• e.g. invalid input, round has expired, etc.</li> </ul> </li> </ul>

13	✓		GET	/play/{gameID}/round-status	<p>Get information about the current round.</p> <p>The purpose of this is so the front-end can frequently check the state of the round, and know when to move on to the outcome.</p>	<ul style="list-style-type: none"> <li>• gameID           <ul style="list-style-type: none"> <li>• (path, string)</li> </ul> </li> </ul>	<ul style="list-style-type: none"> <li>• 200           <ul style="list-style-type: none"> <li>• { "questionID": 1, "status": "pre", "voted": 2, "remaining": 3 }</li> </ul> </li> <li>• roundID is the current round in play for this game           <ul style="list-style-type: none"> <li>• perhaps this should be 0 if we are not actually in a round (e.g. we are still in the role selection phase).</li> </ul> </li> <li>• status of the <b>round</b> can be:           <ul style="list-style-type: none"> <li>• <b>pre</b> - pre-decision. round is in progress, timer is running at back-end, players should be reading the information /discussing/entering their choice</li> <li>• <b>post</b> - post-decision, all players have voted and/or the round timer has expired. Players can no longer enter a choice, and are now shown to the outcome of the decision.</li> </ul> </li> <li>• voted: number of players that have entered their decision</li> <li>• remaining: number of players who still need to enter their decision</li> </ul>
14	✓		GET	/play/{gameID}/round-outcome	<p>Get the outcome and conclusion for the round</p> <p>i.e. "Result" page in the prototype</p>	<ul style="list-style-type: none"> <li>• gameID           <ul style="list-style-type: none"> <li>• (path, string)</li> </ul> </li> </ul>	<ul style="list-style-type: none"> <li>• 200           <ul style="list-style-type: none"> <li>• { "questionID": 1, "question": "Should we build an expensive new software module to help avoid potential issues with the sensor?", "options": [ { "id": 1, "text": "Develop new module"}, { "id": 2, "text": "Do not develop new module"} ], "result": 2, "outcome": ["First paragraph. The decision has been made not to develop this new module.", "Another paragraph. While the Aeronautical Engineer wanted this module to be developed, the other four decided this was not necessary." ] }</li> </ul> </li> <li>• 400           <ul style="list-style-type: none"> <li>• e.g. invalid game ID</li> </ul> </li> </ul>
15			GET	/play/{gameID}/game-outcome	<p>Get the final game outcome of game</p> <p>i.e. "Outcome" page in the prototype</p>	<ul style="list-style-type: none"> <li>• gameID           <ul style="list-style-type: none"> <li>• (path, string)</li> </ul> </li> </ul>	<ul style="list-style-type: none"> <li>• 200           <ul style="list-style-type: none"> <li>• The game outcome</li> </ul> </li> <li>• 400           <ul style="list-style-type: none"> <li>• e.g. invalid game ID</li> </ul> </li> </ul>

# Architecture Design

Architecture:



**Presentation layer:** retrieve and display the data for users

Controller: cooperate the business logic and front data

Page:

- Main page (create or join)
- Create a new game (input name)
- Join an existed game( input roomID, name)
- Present the whole story background
- Role responsibility introduction
- Choose role Page
- Each round information
- Each round decision
- Each round outcome
- Final game outcome
- Present all the choices

## Business logic

- User: login, choose role,
- Role: start the round, choose decision, discuss,
- Round: show the infomation, present the decision and choices, set up the timer, calculate decision and generate a final decision,
- Room: generate the final outcome, present all the choices, create a new room, set the room name

## Domain model:

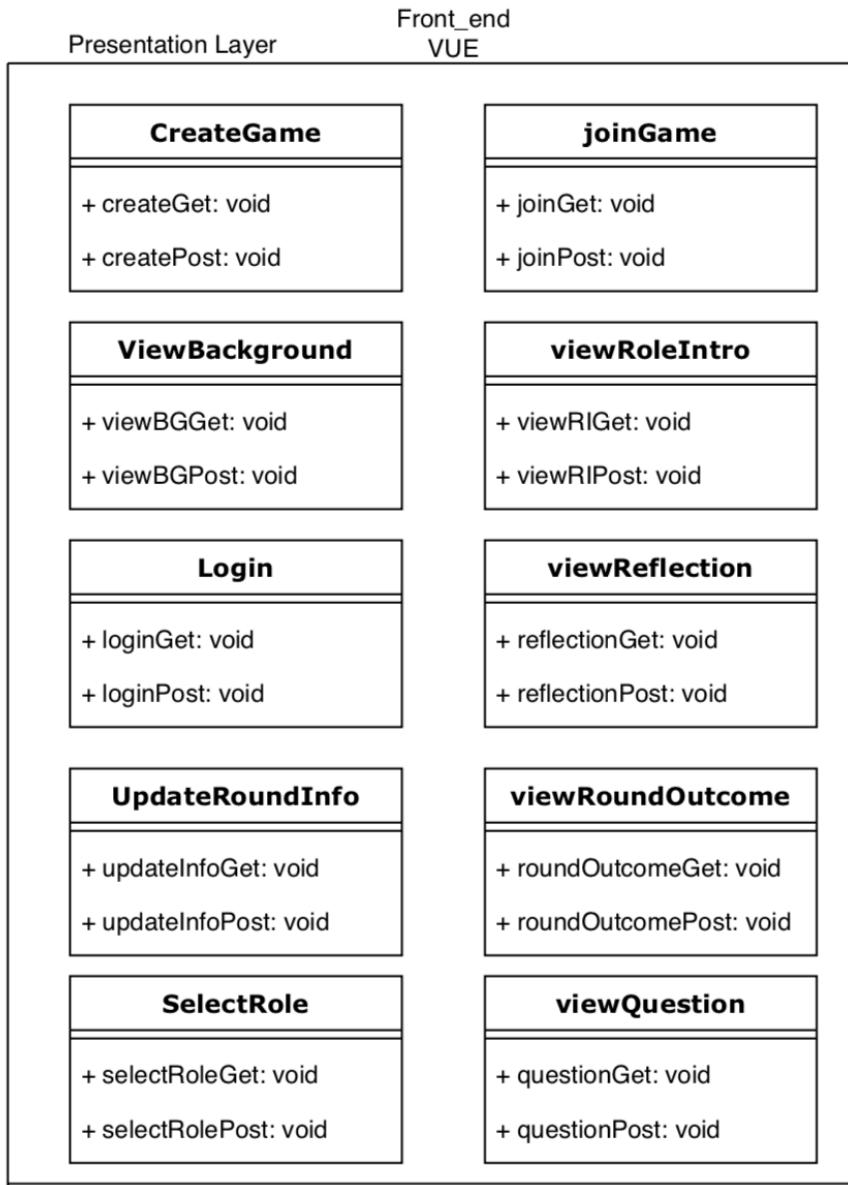
- Story ( -> n round -> 5 Role): background(string), outcome
- Round(>- 1 question): roundID, roundBackground, roundOutcome, method , nextRoundID
- Role: roleId(int), roleName (string), roleBackground
- RoleQuestion: roleId, questionID, weight, individualInfo, answer
- Question (>- n Decision): questionID(int), decision, questionText
- Decision: decisionID, decisionText, madeByCharacter
- Room (-> n User) : roomID(string) , playerNum (int), timeLimit
- User: userID, roleId

1. Enter the room,
2. User choose roles,
3. Present story:

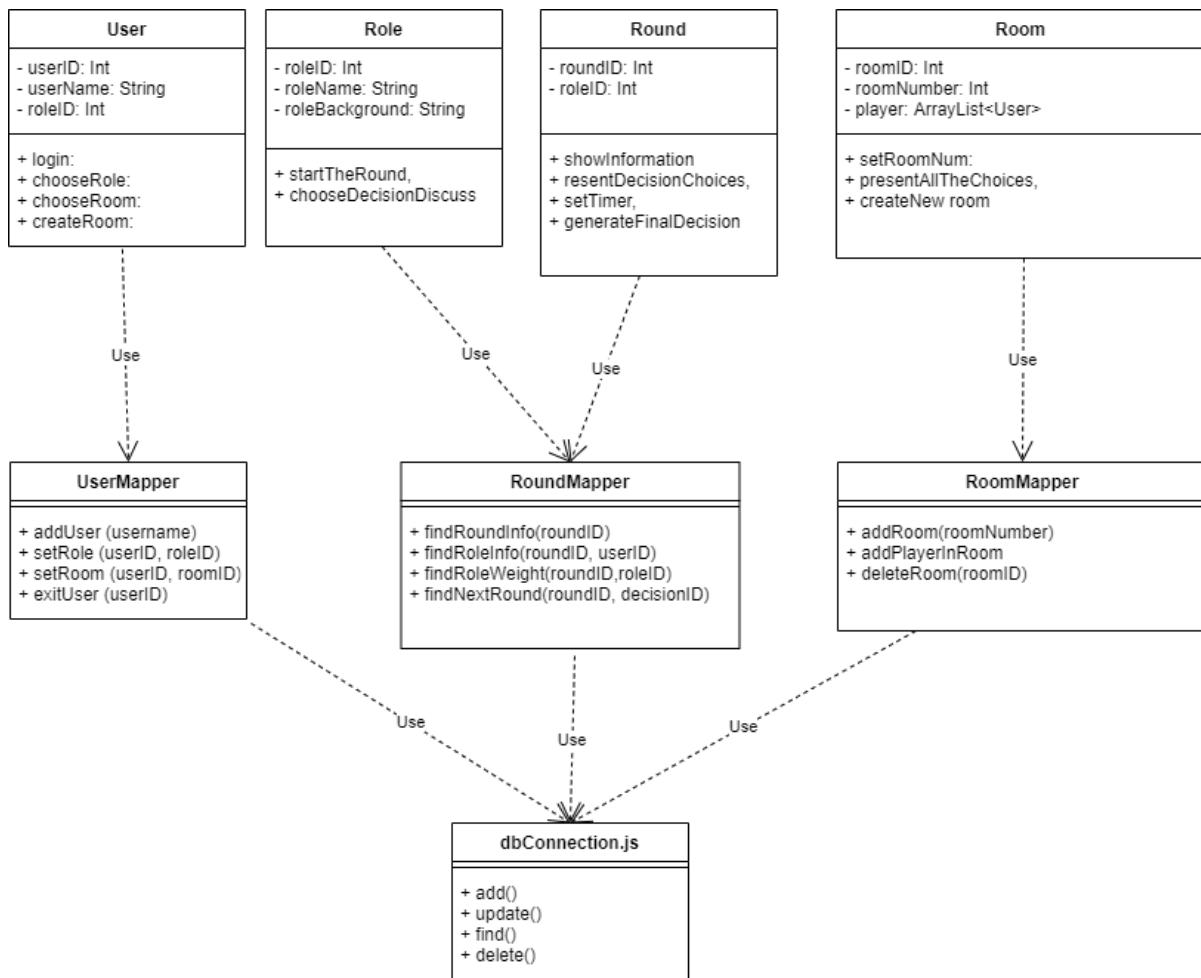
**Data source layer:** SQL

# Class Diagram

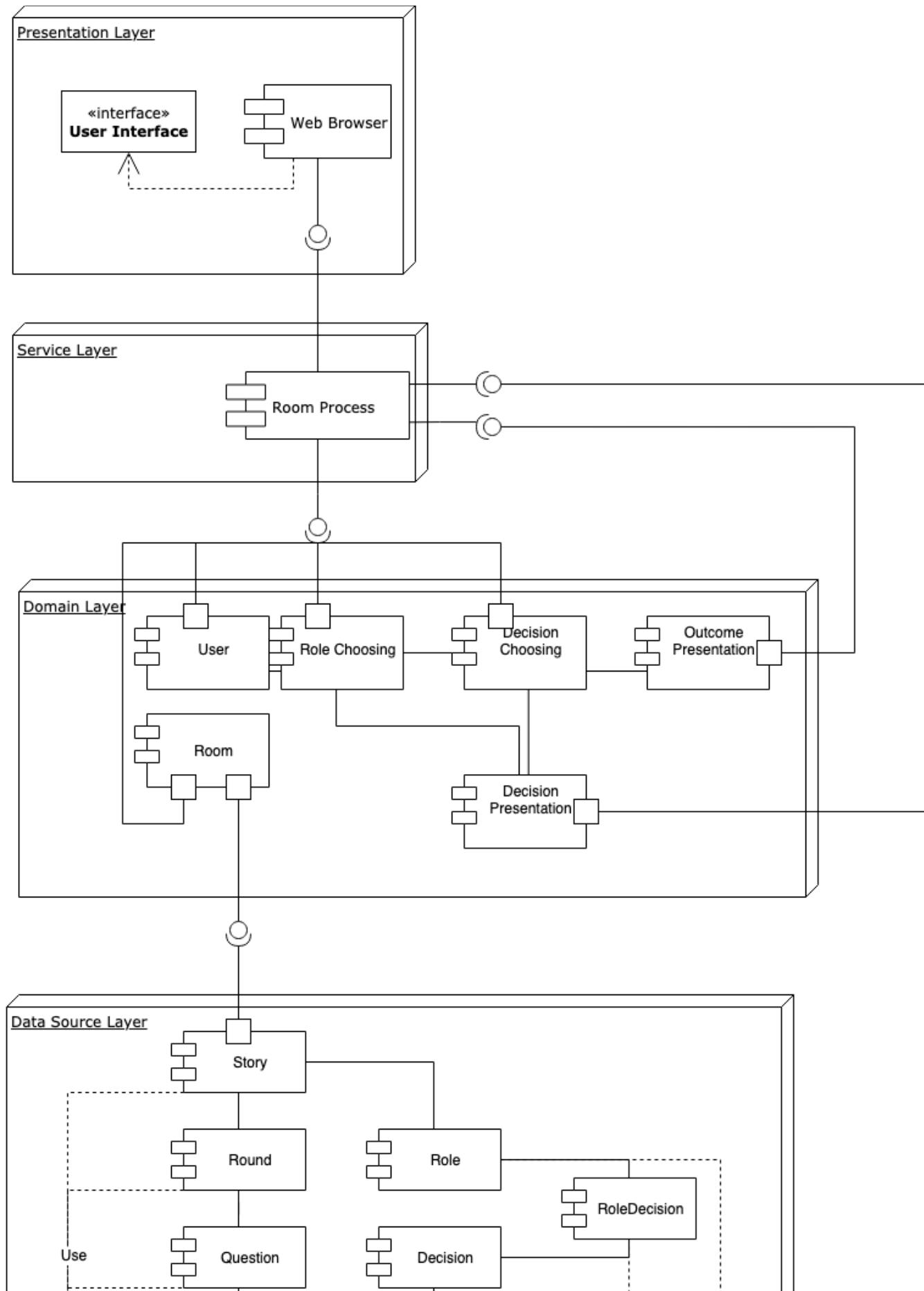
Class DiagramFrontend

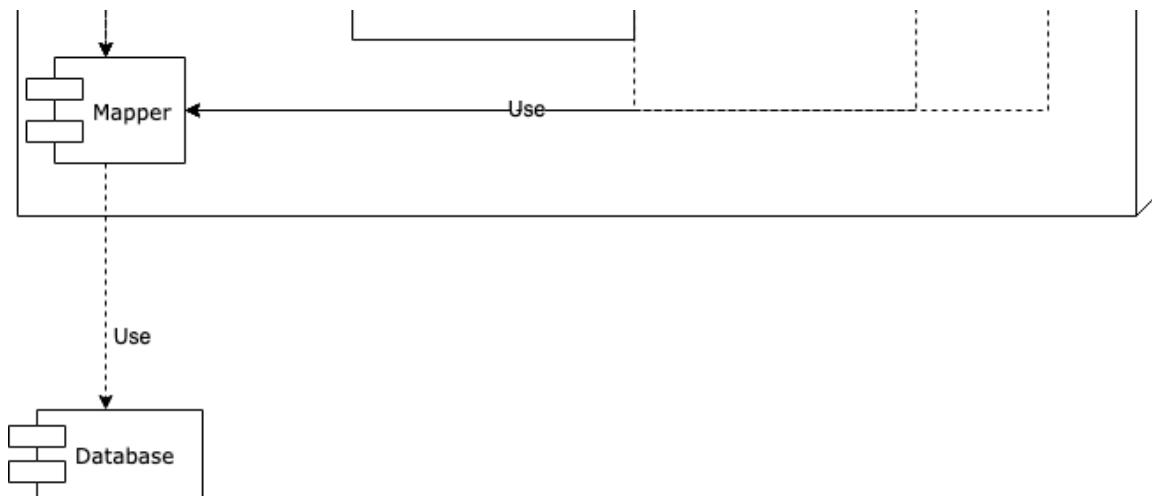


Class DiagramBackend

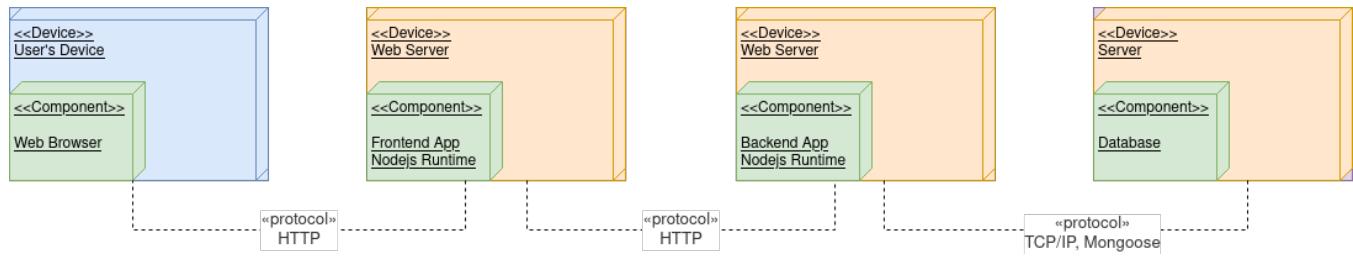


# Component Diagram





# Deployment Diagram



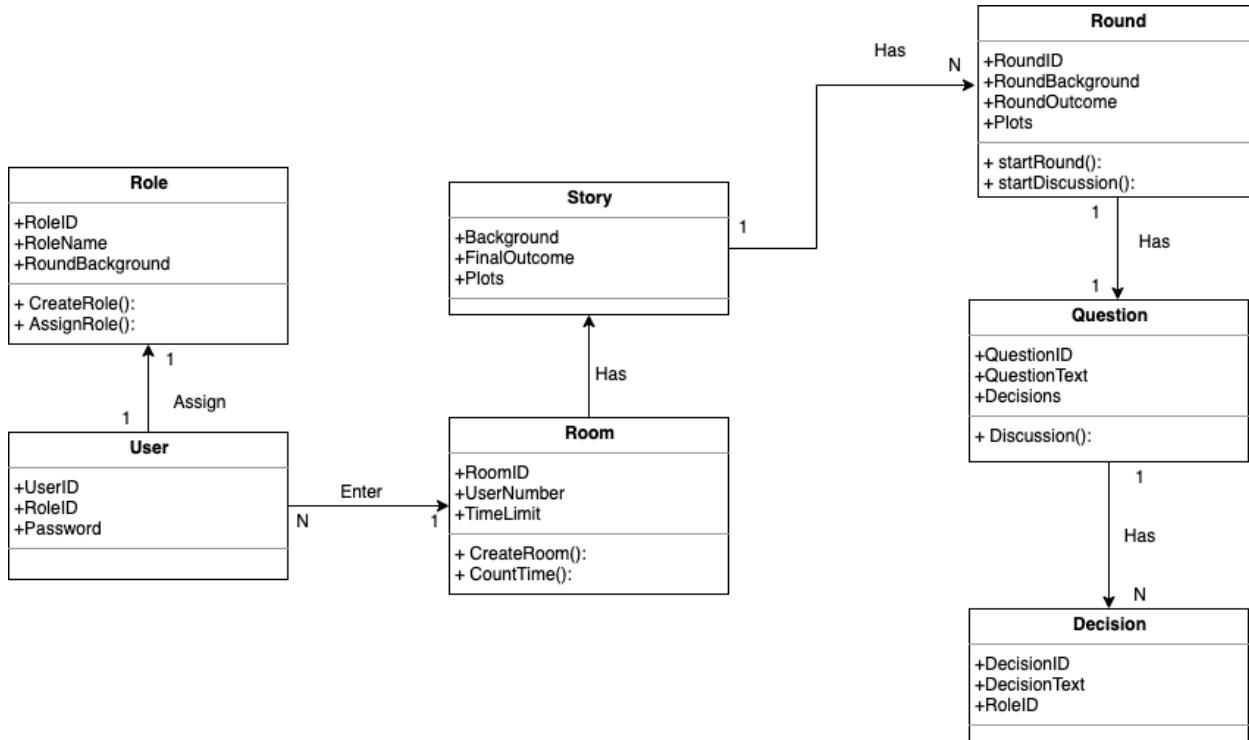
## Notes

- In our current iteration The Front-end and Back-end Applications reside in separate servers. This decoupling can make scaling the system easier in the future.

# Domain Model

This page is Domain model for the project

[https://app.diagrams.net/?state=%7B%22folderId%22:%2216hModzuCo\\_4oxqC1zJFYqjpUR\\_nlupZ6%22,%22action%22:%22create%22,%22userId%22:%2210999963077379038018%22%7D#G1B2Dn9RzMut6Gwm\\_7I-nlBh1hFpn8ghu0](https://app.diagrams.net/?state=%7B%22folderId%22:%2216hModzuCo_4oxqC1zJFYqjpUR_nlupZ6%22,%22action%22:%22create%22,%22userId%22:%2210999963077379038018%22%7D#G1B2Dn9RzMut6Gwm_7I-nlBh1hFpn8ghu0)



# ER Diagram (Database)

games:

```
{"_id": {"$oid": "5edb3b7a45996d0a69c4ca78"},  
"players": [],  
"roles": [{"roleID": {"$numberInt": "0"}, "role": "Boeing Executive", "isAvailable": true, "player": null},  
{"roleID": {"$numberInt": "1"}, "role": "Aeronautical Engineer", "isAvailable": true, "player": null},  
{"roleID": {"$numberInt": "2"}, "role": "Software Developer", "isAvailable": true, "player": null},  
{"roleID": {"$numberInt": "3"}, "role": "FAA Official", "isAvailable": true, "player": null},  
{"roleID": {"$numberInt": "4"}, "role": "Boeing Pilot", "isAvailable": true, "player": null}],  
"duration": {"$numberInt": "0"}, "playerCount": {"$numberInt": "1"},  
"status": "waiting", "__v": {"$numberInt": "0"}}
```

roles:

```
{"_id": {"$oid": "5edc6a4b75432464a82b9255"},  
"name": "jest",  
"__v": {"$numberInt": "0"}}
```

users:

```
{"_id": {"$oid": "5edc6a4a16736071ac6d878f"},  
"name": "jest",  
"university": "jest",  
"studentid": {"$numberInt": "1"},  
"__v": {"$numberInt": "0"}}
```

**non-relational database design in the format of JSON [datasample](#)**

An example:

```

round
{
    roundIndex:"1",
    information:"",
    question:"",
    decisions:
        {A:{content:"", nextRoundIndex:"2"}, .
         B:{content:"", nextRoundIndex:"3"}},
    weights:{role1:0.1, role2:0.4, role3:0.3, role4:0.1, role5:0.1}
    },
    roundIndex:"2",
    information:"",
    question:"",
    decisions:
        {A:{content:"", nextRoundIndex:"4"}, .
         B:{content:"", nextRoundIndex:"5"}},
    ...
}

user
{
    userID:"",
    roomID:""
}

```

## Decision-Tree Design

### Key Features

- As per a standard tree design, the tree will feature a singular root node that corresponds to the starting state of the game
- Throughout a game session, a singular 'company' object will be persisted, which will represent the full state of information about the status of the company. It is a subset of this 'perfect' information set that will be delivered to each of the player roles where appropriate
- A question will list the options the player can take, where each option itself includes a 'consequence' object that describes how choosing that option influences the 'company' information, as well as the next decision node to move to in the tree

### Logical Flow

- Question is moved to, the Question sends player-specific information from the 'company' object to all players, and the decision question to the necessary player/s, with all of the corresponding options
- The player/s make the decision, based on the 'consequence' object for that decision, the 'company' object is updated, the decision history is updated
- Repeat!

### Objects

#### Game

- Question currQuestion: A Question from a singleton 'Decision Tree' object persisted in back-end, representing the current Question the game is at
- Company companyInformation: An object storing all information about the company, which is updated based on decisions made during the game
  - Financial information, environmental information, productivity information, PR information etc.

#### Question

- int id: The identification number for the node
- string question: The decision question
- List<Option>: The options available for the decision, minimum 1, maximum 4
- int duration: The amount of time allocated for discussion/decision-making for this node
- List<int>: The weights for each role for the decision, should sum to 100 (corresponding to 100%)

### Option

- string description: The description for the particular option
- Consequence consequences: An object storing the deltas for values in the Company object
  - This will be an identical looking object to the Company object
  - e.g. Company has a value netWorth = 100, Consequence has a value netWorth = -10, final netWorth value = 100 - 10 = 90
- Question next: The node to move to if this decision is chosen

### Company

- Financials financialInfo
  - int contracts: Company contracts with airlines (Initial: 80)
  - int compContracts: Competitor contracts with airline (Initial: 20)
  - int shares: Company share value (Initial: 40)
- Reputations reputationInfo
  - int publicRep (0-100): Company reputation with the general public (Initial: 50)
  - int regulatorRep (0-100): Company reputation with the FAA (Initial: 50)
  - int internalRep (0-100): Company reputation with internal employees/managers (Initial: 50)
  - int boardRep (0-100): CEO reputation with board of directors (Initial: 50)
  - int enviroRep (0-100): Company reputation with environmental organizations (Initial: 50)
- Project projectInfo
  - int expectedProg (0-100): The expected level of progress of project according to board of directors (Initial: 0)
  - int actualProg (0-100): The actual level of progress of project (Initial: 0)
  - int actualCost: The actual level of project cost (Initial: 0)
  - int expectedCost (0-100): The expected level of project cost according to board of directors (Initial: 0)
  - int costDelay (0-100): The delay of the cost

We will need to update the Role object to include information about what values in the Company object that particular role can observe.

This model will require 1 unique Company object per game.

### **Advantages**

- This design supports the presentation of role-specific information to the five different supported roles in the game
- Only one decision tree object is required to be persisted in back-end for ALL GAMES, no need for redundant copying of data
- Supports granular control of decision consequences on the company, as well as the complex structure of a real-life decision tree
- Supports individual player decisions, group decisions, and whole game decisions
- Easy to make changes, fairly flexible structure

### **Extensions**

#### **Feedback**

- There needs to be an assurance on the decision tree that there is no cyclic behaviour, for example, if a child of a parent node directed back to the parent node, we could have an infinite loop, no node should be visited more than once

## **Decision History**

### **Decision-Tree Design**

**Solution:** Basic ArrayList/Linked List Implementation.

### **Objects**

#### List

- Decision head: The first decision in the list
- Decision last: The last decision in the list
- int size: The length of the decision list

#### Decision

- Question question: A Question object corresponding to the tree node visited in the game
- Option chosen: The Option object chosen
- Decision next: The next Decision object in the list

We will make one of these lists per game.

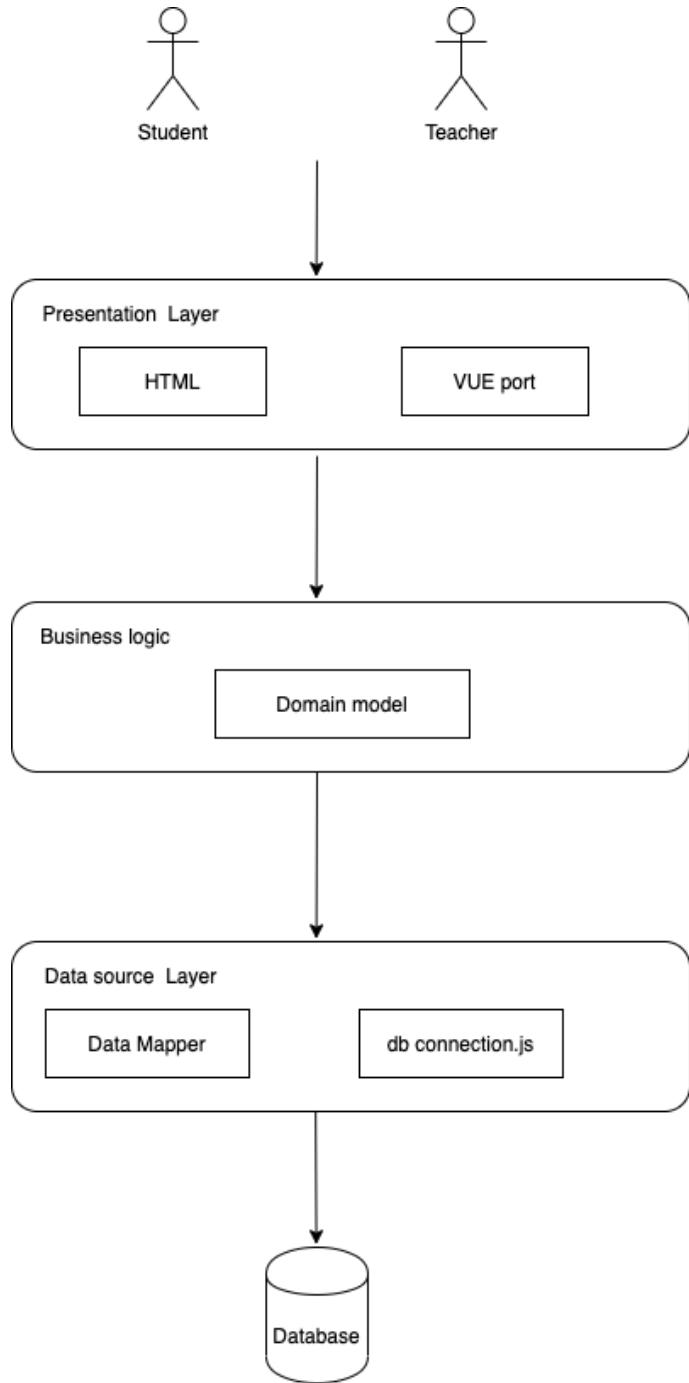
### **Advantages**

- Easy to iterate through this structure and present a decision history
- Can be naturally added to as the game progresses, uses objects from decision-tree implementation
- Again, no need to make any copies of Question themselves
- We get a natural ordering that can be accessed e.g. 1->5->10->15...

# High Level Diagram

This page is the high level diagram for the project.

[https://app.diagrams.net/#G1ty3dOC0gLK\\_KkIfQ2W1Ah9oSHx3FZ0a9](https://app.diagrams.net/#G1ty3dOC0gLK_KkIfQ2W1Ah9oSHx3FZ0a9)



# Sequence Diagram

Tool: <https://sequencediagram.org/>

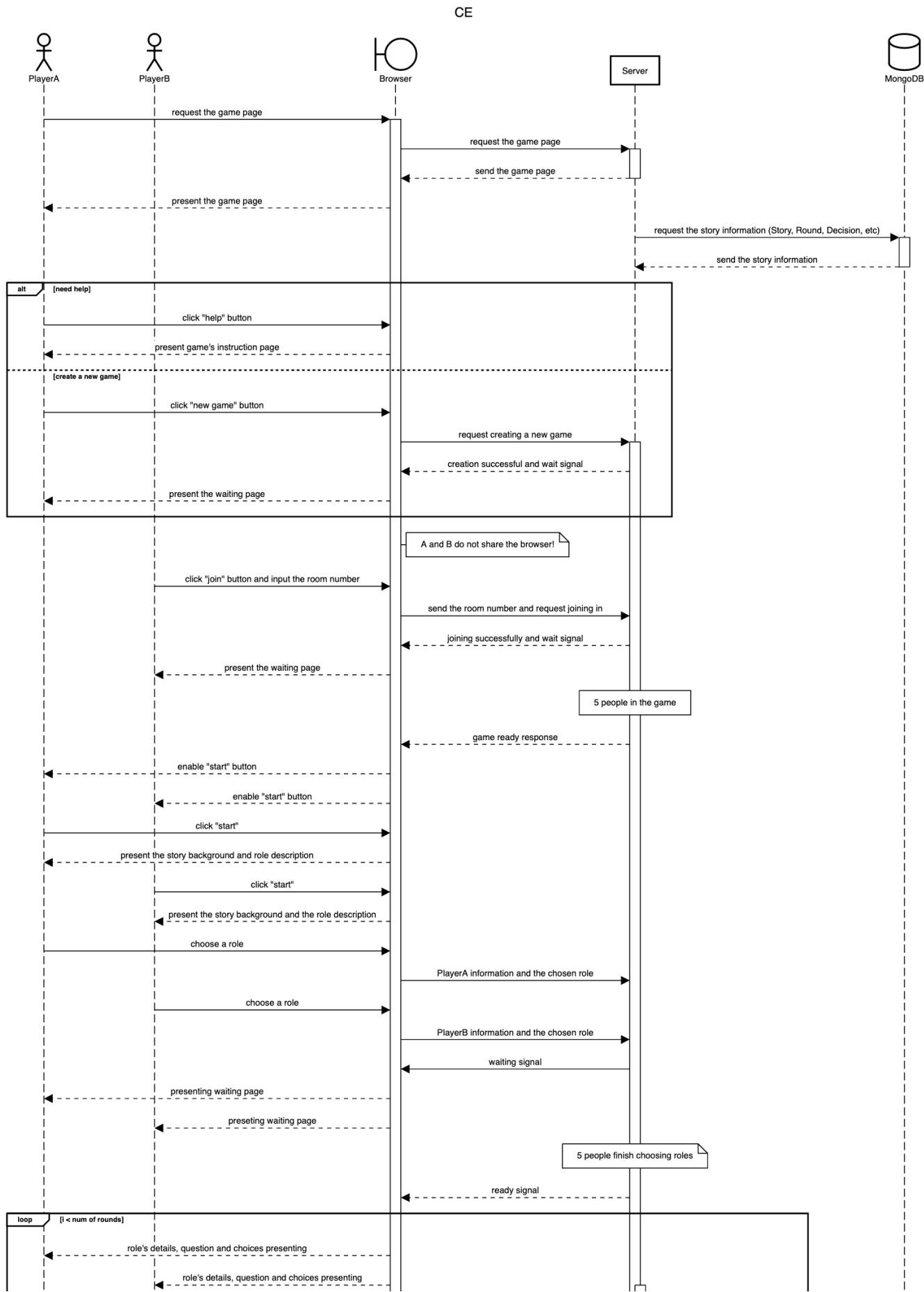
Link of the current diagram and code: [CE Sequence Diagram](#)

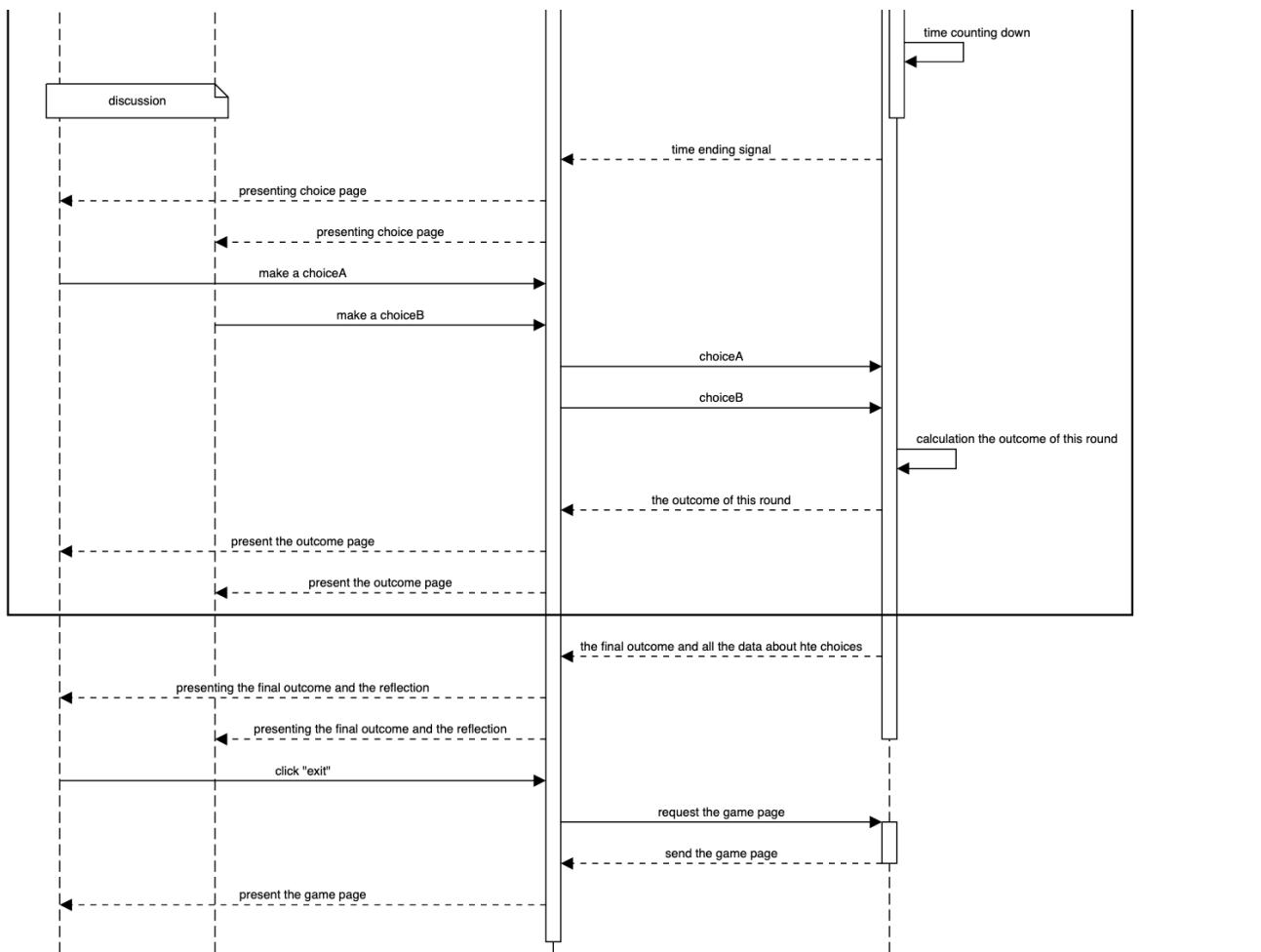
**Note:**

It is important to know that **some details of actions are not shown in this diagram**, otherwise this diagram will become really huge.

For example, the role description, the background information for each round and the decision for each round should be in different pages; the player can only see the next page when they click "next".

However, the assumption is that **all the data and resources should be transmitted back to the browser within one request and one response between the server and the browser (especially for playing in round part)**, so that the page forwarding can be done by the browser and no longer depend on the server.





# Technology

## Frontend

Vue.js

For more information, please refer to [Front-End Tools](#)

## Backend

Node.js

For more information, please refer to [Back-end Tools](#)

## Deployment

Docker: For setting up local environments

Gitlab: For building CI/CD pipelines

For more information, please refer to [Dev-Ops Tools](#)

# Project Plans

# Sprint1 Report(Planning)

## Sprint planning

We conducted a Sprint Planning meeting on 2022-8-19

With our requirements all prepared, decisions made and planning completed including a digital prototype and architectural design, the purpose of next 2 sprints is to complete user stories to improve the game.

This included initialising and deploying the frameworks used by the game, debug it to make sure it will run successfully on local device, then realize new features.

## User stories

The following user stories were planned for Sprint 2 and 3. Only 2 features are chosen for Sprint2 as we need time to get familiar with the source code and debug it. Also, we plan to condense new requirements from clients during Sprint 2 and 3 to make the game more interesting.

Sprint	User Story	Decomposed Case	Tasks	Point	Priority
2	US_01	Simplify the game process	1. Exploit the game and write a report for user experience.	5	Must Have
			2. Assess the existing decision tree.	9	
			3. Simplify the decision tree to shortcut the game progress	9	
			4. Limit one turn game with 10 minutes.	7	
		Improve choices for each character and report	1. Assess existing characters' choices and the final report.	6	Must Have
			2. Distinguish choices for each character.	5	
			3. Improve the report quality when finishing the game.	5	
	US_02	To submit my option successfully	1. Find out reasons why game gets stuck somehow during the game.	7	Must Have
			2. Reproduce bugs and fix.	5	
			3. Testing	6	
3	US_03	To get a overall report that shows decisions made by other users for each questions throughout the game	1. When the game finished server send back a track for each one's choices long with the report.	8	Must Have
			2. Show everyone's choices as well as the report in a specific interface.	6	
	US_04	To be able to rejoin the game session	1. Assess existing rejoin functionality.	5	Better Have
			2. Reproduce bugs and fix.	5	
			3. Testing	5	
	US_05	See other players' choices when playing	1. When other players' making decisions, send the choice to back end.	4	Better Have
			2. When server receives choices from other players, send it to front end.	4	
			3. Front end shows choices made by other players.	4	
		See previous choices when playing	1. When making each decision, front end will take a record on decisions made.	4	Better Have
			2. Build a list interface to show previous choices.	4	
			3. Show previous choices when user click on the list.	4	
	US_06	To add a timer to limit the discussion time for each decision in game	1. Add a timer on each choice when playing the game, and set it for 60 seconds.	5	Better Have

	2. When time's out and the player still make any choice, it will pick a choice randomly and send it back to server.	5
	3. Inform player that time is out, and show the random choice made for player.	5

**TO DO**

- US\_01\_1\_Task1: Exploit the game and write a report for user experience. (Due: 31 Aug, Assignees: DX, ZE)
- US\_01\_1\_Task2: Assess the existing decision tree. (Due: 29 Aug, Assignees: HH, JH, YB)
- US\_01\_1\_Task3: Simplify the decision tree to shortcut the game progress. (Due: 12 Sep, Assignees: DX, HH, ZE)
- US\_01\_1\_Task4: Limit one turn game with 10 minutes. (Due: 29 Aug, Assignees: JH, YB)
- US\_01\_2\_Task1: Assess existing characters' choices and the final report. (Due: 5 Sep, Assignees: DX, HH, ZE)
- US\_01\_2\_Task2: Distinguish choices for each character. (Due: 5 Sep, Assignees: DX, HH, ZE)
- US\_01\_2\_Task3: Improve the report quality when finishing the game. (Due: 10 Sep, Assignees: JH, YB)
- US\_02\_1\_Task1: When other players' making decisions, send the choice to back end. (Due: 3 Sep, Assignees: JH, YB, ZE)
- US\_02\_1\_Task2: When server receives choices from other players, send it to front end. (Due: 10 Sep, Assignees: DX, HH)
- US\_02\_1\_Task3: Front end shows choices made by other players. (Due: 10 Sep, Assignees: DX, HH)
- US\_02\_2\_Task1: When making each decision, front end will take a record on decisions made. (Due: 16 Sep, Assignees: DX, JH, ZE)

**DOING**

+ Add a card

**DONE**

+ Add a card

+ Add another list

**Board** **GE\_SPRINT3** COMP90082-2022-SM2-GE-RedBack

**To Do** **Doing** **Done** **+ Add another list**

**To Do**

- US\_01\_Task1: When the game finished server send back a track for each one's choices long with the report.  
🕒 30 Sep
- US\_01\_Task2: Show everyone's choices as well as the report in a specific interface.  
🕒 30 Sep
- US\_02\_Task1: Assess existing rejoin functionality.  
🕒 7 Oct
- US\_02\_Task2: Reproduce bugs and fix.  
🕒 7 Oct
- US\_02\_Task3: Testing  
🕒 7 Oct
- US\_03\_Task1: Find out reasons why game gets stuck somehow during the game.  
🕒 14 Oct
- US\_03\_Task2: reproduce bugs and fix.  
🕒 14 Oct
- US\_03\_Task3: Testing  
🕒 14 Oct
- US\_04\_Task1: Add a timer on each choice when playing the game, and set it for 60 seconds.  
🕒 20 Oct
- US\_04\_Task2: When time's out and the player still make any choice, it will pick a choice randomly and send it back to server.  
🕒 20 Oct
- US\_04\_Task3: Inform player that time is out, and show the random choice made for player.

**+ Add a card**

# Sprint2 Report

The main tasks we have done in Sprint 2 is debug the game and simplify the decision process.

- A bug-free version can be accessed via this link: <https://game-of-ethics-2022.herokuapp.com>
- Bugs fixed can be seen here: [Bug Report](#)
- The simplified decision process can be found here: [Decision Tree](#)

The frontend of this project is built by vue and backend is by JavaScript, which are not familiar for the whole team. We spent almost 2 weeks on learning the techs and figuring out the source code of the project.

There is a certain delay on sprint2 because that it took us more time than expected to debug and make sure this game can run successfully. For the rest of time, we focused on rebuild the decision tree to make sure the game can be finished within 20 minutes.

The main problem for this game that prevent it from running successfully is mainly caused by logical bugs. Our team got stuck for a long time during testing and reproducing because the logs didn't show enough information for us to locate the problems. We aim to polish the logs in the future to make debug easier.

Sprint	User Story	Decomposed Case	Tasks	Priority	Achieved
2	US_01	Simplify the game process	1. Exploit the game and write a report for user experience.	5	<span style="color: red;">Must Have</span>
			2. Assess the existing decision tree.	9	
			3. Simplify the decision tree to shortcut the game progress	9	
			4. Limit one turn game with 10 minutes.	7	
		Improve choices for each character and report	1. Assess existing characters' choices and the final report.	6	<span style="color: red;">Must Have</span>
			2. Distinguish choices for each character.	5	
			3. Improve the report quality when finishing the game.	5	
	US_02	To submit my option successfully	1. Find out reasons why game gets stuck somehow during the game.	7	<span style="color: red;">Must Have</span> 
			2. reproduce bugs and fix.	5	
			3. Testing	6	

In Sprint 3, we plan to impletent the following user stories:

Sprint	User Story	Decomposed Case	Tasks	Priority
3	US_03	To get a overall report that shows decisions made by other users for each questions throughout the game	1. When the game finished server send back a track for each one's choices long with the report.	8
			2. Show everyone's choices as well as the report in a specific interface.	6
	US_04	To be able to rejoin the game session	1. Assess existing rejoin functionality.	5
			2. Reproduce bugs and fix.	5
			3. Testing	5
	US_05	See other players' choices when playing	1. When other players' making decisions, send the choice to back end.	4
				<span style="color: green;">Better Have</span>

		2. When server receives choices from other players, send it to front end.	4	
		3. Front end shows choices made by other players.	4	
See previous choices when playing		1. When making each decision, front end will take a record on decisions made.	4	Better Have
		2. Build a list interface to show previous choices.	4	
		3. Show previous choices when user click on the list.	4	
US_06 To add a timer to limit the discussion time for each decision in game		1. Add a timer on each choice when playing the game, and set it for 60 seconds.	5	Better Have
		2. When time's out and the player still make any choice, it will pick a choice randomly and send it back to server.	5	
		3. Inform player that time is out, and show the random choice made for player.	5	

**Board** **GE\_SPRINT2**

**Power-Ups** **Automation** **Filter** **Show menu**

**TO DO**

US\_02\_2\_Task3: Show previous choices when user click on the list.  
 2 15 Sep

+ Add a card

**DOING**

US\_02\_1\_Task3: Front end shows choices made by other players.  
 1 10 Sep

US\_02\_2\_Task2: Build a list interface to show previous choices.  
 1 16 Sep

+ Add a card

**DONE**

US\_02\_2\_Task1: When making each decision, front end will take a record on decisions made.  
 16 Sep

US\_01\_1\_Task3: Simplify the decision tree to shortcut the game progress.  
 12 Sep

US\_02\_1\_Task2: When server receives choices from other players, send it to front end.  
 10 Sep

US\_01\_2\_Task3: Improve the report quality when finishing the game.  
 10 Sep

US\_01\_2\_Task2: Distinguish choices for each character.  
 5 Sep

US\_01\_2\_Task1: Assess existing characters' choices and the final report.  
 5 Sep

US\_02\_1\_Task1: When other players' making decisions, send the choice to back end.  
 3 Sep

US\_01\_1\_Task4: Limit one turn game with 10 minutes.  
 29 Aug

US\_01\_1\_Task2: Assess the existing decision tree.  
 29 Aug

US\_01\_1\_Task1: Exploit the game and write a report for user experience.  
 31 Aug

+ Add a card

**+ Add another list**

**To Do**

- US\_01\_Task1: When the game finished server send back a track for each one's choices long with the report.  
Due: 30 Sep  
Assigned: HH, DX, ZE
- US\_01\_Task2: Show everyone's choices as well as the report in a specific interface.  
Due: 30 Sep  
Assigned: JH, YB
- US\_02\_Task1: Assess existing rejoin functionality.  
Due: 7 Oct  
Assigned: DX, ZE
- US\_02\_Task2: Reproduce bugs and fix.  
Due: 7 Oct  
Assigned: JH, YB
- US\_02\_Task3: Testing  
Due: 7 Oct  
Assigned: HH
- US\_03\_Task1: Find out reasons why game gets stuck somehow during the game.  
Due: 14 Oct  
Assigned: DX, YB
- US\_03\_Task2: reproduce bugs and fix.  
Due: 14 Oct  
Assigned: HH, ZE
- US\_03\_Task3: Testing  
Due: 14 Oct  
Assigned: JH
- US\_04\_Task1: Add a timer on each choice when playing the game, and set it for 60 seconds.  
Due: 20 Oct  
Assigned: DX, HH
- US\_04\_Task2: When time's out and the player still make any choice, it will pick a choice randomly and send it back to server.  
Due: 20 Oct  
Assigned: JH, YB, ZE
- US\_04\_Task3: Inform player that time is out, and show the random choice made for player.

**Doing**

**Done**

**+ Add another list**

**Burndown**



A burn down chart is a work chart used to represent the remaining workload. The horizontal axis represents the time and the vertical axis represents the amount of work. This chart provides a visual forecast of when the work will be completed. The blue line in the chart shows the planned remaining workload and the orange line shows the actual remaining workload.

In the early days of sprint2 we were on track to complete our tasks every day. However, after September 2 we encountered some difficulties that slowed down the overall progress and we ended up with some unmanageable tasks left. We plan to address these difficulties in sprint3

# Sprint3 Report

The main tasks we have done in Sprint 2 is to implement the new decision tree and polish UI.

We modified the structure of [decision tree](#) based on Sprint 2. However, the workload of the implementation of the decision tree exceeded our expectation.

As we reconstructed the structure, implement of the decision tree is divided into three parts: questions, options, and frontend backend interaction.

Moreover, we improved the UI to make the game more user-friendly and attractive for players.

Sprint	User Story	Decomposed Case	Tasks	Priority	Achieved
3	US_01	Simplify the game process	1. Exploit the game and write a report for user experience.	5	Must Have
			2. Assess the existing decision tree.	9	
			3. Simplify the decision tree to shortcut the game progress	9	
			4. Limit one turn game with 10 minutes.	7	
US_04		To be able to rejoin the game session	1. Assess existing rejoin functionality.	5	Better Have
			2. Reproduce bugs and fix.	5	
			3. Testing	5	
US_05		See other players' choices when playing	1. When other players' making decisions, send the choice to back end.	4	Deleted. n/a - Players will have different questions in reconstructed decision tree.
			2. When server receives choices from other players, send it to front end.	4	
			3. Front end shows choices made by other players.	4	
		See previous choices when playing	1. When making each decision, front end will take a record on decisions made.	4	
			2. Build a list interface to show previous choices.	4	
			3. Show previous choices when user click on the list.	4	
US_06		To add a timer to limit the discussion time for each decision in game	1. Add a timer on each choice when playing the game, and set it for 60 seconds.	5	Better Have
			2. When time's out and the player still make any choice, it will pick a choice randomly and send it back to server.	5	
			3. Inform player that time is out, and show the random choice made for player.	5	
US_07		To have an uncluttered and appealing user interface	1. Set animations to buttons.	5	Better Have
			2. Polish the background of the webpage.	5	

Trello

**To Do**

+ Add a card

**Doing**

+ Add a card

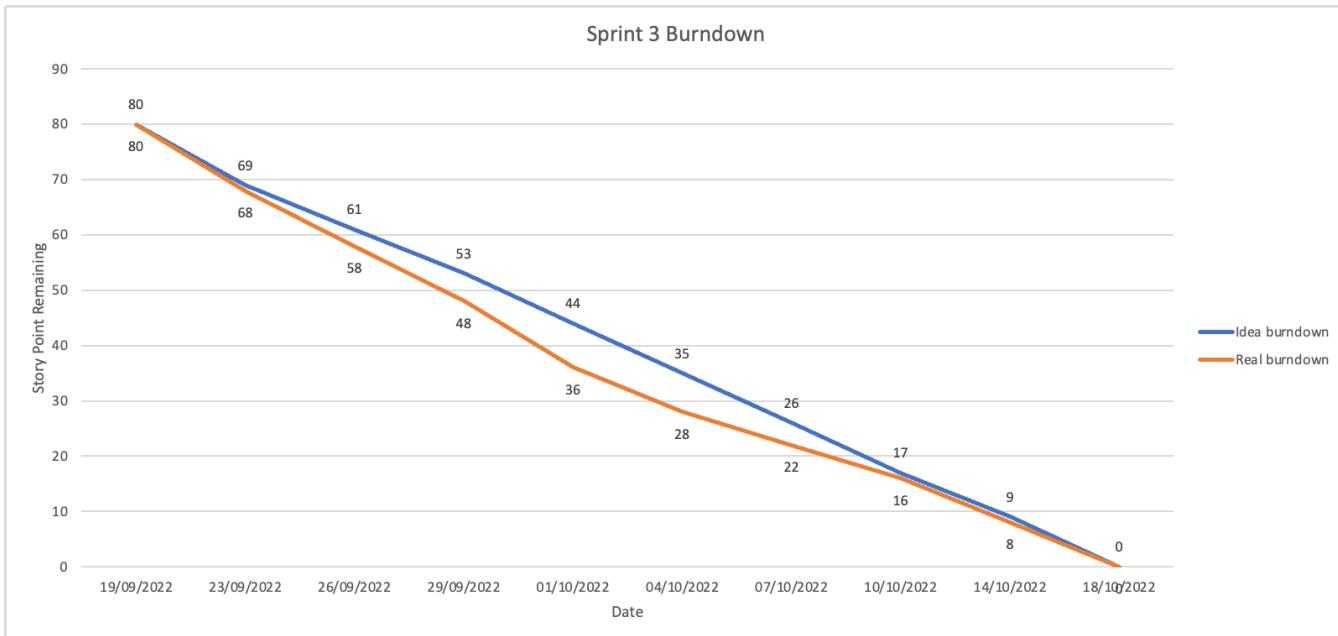
**Done**

+ Add another list

- US\_01\_Task1: Exploit the game and write a report for user experience
  - ⌚ 30 Sep HH DX JH YB ZE
- US\_01\_Task2: Assess the existing decision tree.
  - ⌚ 30 Sep JH YB
- US\_01\_Task3: Simplify the decision tree to shortcut the game progress
  - ⌚ 7 Oct JH YB
- US\_01\_Task4: Limit one turn game with 10 minutes.
  - ⌚ 7 Oct HH
- US\_04\_Task1: Assess existing rejoin functionality
  - ⌚ 14 Oct DX YB
- US\_04\_Task2: Reproduce bugs and fix.
  - ⌚ 14 Oct HH ZE
- US\_04\_Task3: Testing
  - ⌚ 7 Oct DX ZE
- US\_06\_Task1: Add a timer on each choice when playing the game, and set it for 60 seconds.
  - ⌚ 14 Oct JH
- US\_06\_Task2: When time's out and the player still make any choice, it will pick a choice randomly and send it back to server.
  - ⌚ 20 Oct DX HH
- US\_06\_Task3: Inform player that time is out, and show the random choice made for player.
  - ⌚ 20 Oct JH YB ZE
- US\_07\_Task1: Set animations to buttons.
  - ⌚ 20 Oct HH YB
- US\_07\_Task2: Polish the background of the webpage.
  - ⌚ 20 Oct DX ZE

+ Add a card

Burndown



A burn down chart is a work chart used to represent the remaining workload. The horizontal axis represents the time and the vertical axis represents the amount of work. This chart provides a visual forecast of when the work will be completed. The blue line in the chart shows the planned remaining workload and the orange line shows the actual remaining workload.

In sprint3, we first complete the tasks that were left undone in sprint2. Overall in sprint3 we were on schedule every day, and between September 28 and October 7 we overshot some tasks, preventing us from having time to deal with new tasks that came in before the submission deadline.

# Quality Assurance

## Quality Assurance plan and procedures

The section describes the quality assurance procedures and plans to ensure high product quality.

The quality assurance plan will be divided into the following phases:

- Verification phase
- Validation phase
- Code Review phase

### Verification phase

The verification phase should contain the following procedures:

1. Prepare the acceptance test to ensure that the functionality described in the user story works properly and as required by the user story.
2. Test the functionality using scripts
3. Test the functionality using manual testing where usage of scripts is irrelevant.
4. In case there found irrelevant behaviour in the program functionality, record the behaviour on the Confluence and let the responsible team/person know about a bug.

### Validation phase

The validation phase should contain the following procedures:

1. Identify the requirements the client needs for the current sprint.
2. The assumptions made by the team during the implementation which were not verified by the client must be recorded on the confluence page.
3. If the meeting with the client has occurred, record client's feedback and comments about the assumptions.
4. Make adjustment (if any) to the previous sprint requirements based on client's feedback.

### The Code Review phase.

Note: the code review is highly recommended to be completed by several people in each team to identify the possible mistakes and confusions in the code.

The code review should contain the following procedures:

- Check the completed code for the major/critical mistakes and make sure that the code is easy to understand for another person who reads it.
- If there found an unclear code part or part which potentially lead to unreliable behaviour, contact the developing person for the explanation. If the code leads to unintentional behaviour, record it on the Confluence page.

# Coding Standards

This section describes the coding standards for the project for consistent standards across the whole project

## General Guidelines:

1. For **Javascript Code** Please adhere to the [Airbnb javascript style guide](#)

## Models:

The models' names should be exported using the **Upper Camel Case** for the models created specifically for this project.

E.g: user model has to be exported for usage in controllers as "User" with the upper case as the first letter.

## Controllers:

The controller names should be constructed using the **Lower Camel Case** for procedures which are exported to the routes for future usage.

## Routes:

The routes names for URL addresses should be built using **Lower Camel Case** for usage together with Views

The exported **models** used in the route has to use **Upper Camel Case** for development.

The exported **controllers** used in the route has to use **Lower Camel Case** for development.

# Review process

The review process occurs during the **review** phase of the task on **Trello** platform.

- 1**Review process of the project coding tasks**
  - 1.1**Placing the coding task under review**
  - 1.2**Reviewing process of coding tasks:**
- 2**Review process of the project documentation**
  - 2.1**Placing the documentation task under review**
  - 2.2**Reviewing process of documentation tasks:**
- 3**Team members responsibility**

## Review process of the project coding tasks

### Placing the coding task under review

The assigning process includes the following steps:

- 1) Creating a **pull request** on the GitHub platform to the develop branch
- 2) Assign a person using Jira to complete the review process before merging with the **develop** branch. Note: it is preferred to choose a person who is currently working or worked on the similar/same feature before and has an idea of how the feature was implemented for better testing results. Otherwise, choose a person who is responsible for testing or leader of the team.
- 3) For consistent communication between team members, it is advised to also notify the reviewer via Slack.

### Reviewing process of coding tasks:

The reviewing process includes the following:

- 1) Using testing scripts provided by the testing team on the GitHub platform
- 2) Test the new feature implemented in the pull request for correct functionality
- 3) Test other features which might potentially be related to the current feature and ensure that it does not cause new errors.
- 4) If the user story has more than 5 points, it is required for 2 team members to review the current process

## Review process of the project documentation

### Placing the documentation task under review

The assigning process includes the following steps:

- 1) Placing the task in under review on **Trello** platform and assigning a person who will review it. Note: it is preferred to assign a person who is familiar with the similar job for better results and better efficiency.
- 2) For consistent communication between team members, it is advised to also notify the reviewer via Slack

### Reviewing process of documentation tasks:

- 1) Check the documentation on **Confluence** platform which was assigned to the team member.
- 2) Check the documentation consistency with other documents and ensure that it is clear and understandable for all team members to follow the documentation guidelines.
- 3) Check the documentation for grammatical mistakes and typos to ensure the high quality of the document.
- 4) Check the reference links for correctness and clear explanations for the corresponding document.
- 5) Write a review as a comment on the JIRA task (or, as a comment in Confluence, in which case leave a comment on the JIRA task linking to the Confluence comment).

## Team members responsibility

To ensure the consistency of the review process each team member is required to follow the next process:

- 1) Check the **Trello** and **Slack** platforms at least once a day to identify the required review goals assigned to each team member
- 2) Make a review for the task assigned to the person
- 3) Comment the pull request and approve/reject the request based on performance.

# Version Control

# Git Workflow

## Motivation

A git workflow is an agreed-upon approach used to using git tools within a team. This is important from both a Quality Assurance and a Dev-Ops perspective. For Quality Assurance it allows us to ensure that unstable/untested code isn't mixed with a branch that is designed to hold stable code. From a Dev-Ops standpoint, it allows us to have set conventions as to which branch represents what environment (Development, Staging, Production)

## Gitflow

Gitflow is a git workflow suited for collaboration amongst a team.

Frontend branches:

(<https://github.com/COMP90082-2022-SM2/GE-RedBack-FrontEnd>)

- **Master branch:** This is used for production (The product after a sprint).
- **Reqmodify branch:** At the end of a sprint we create a release branch from the develop branch to apply final changes before merging to master. This could be bug-fixes or removing features that aren't ready. This branch is then merged to master.
- **Timer\_v1:** This branch is made for development of timer.
- **ui\_improve:** This branch is made for adding ui animations or changing background.

Backend branches:

(<https://github.com/COMP90082-2022-SM2/GE-RedBack-BackEnd>)

- **Master branch:** As only the logic of decision tree need be changed, all the changed is pushed to master branch

## Workflow Conventions

### Developing a feature

1. Developers building a feature should only be working in the relevant **feature branch**.
2. Once a feature is considered ready for review, a pull request is made to merge the feature in the **develop branch**.
3. If the pull request is approved the feature is merged to the develop branch. These new changes on the develop branch **must** now be merged into **all feature branches**.

### Creating a release

1. Towards the conclusion of the sprint, a release branch is made by the **deployment team**. The deployment team will check for any issues before deployment to a production environment and talk to the product owner, front-end and back-end team leads about the removal of any features if necessary.
2. Once the release branch is ready, it is then merged to **master** and **develop** (only the bugfixes for develop)
3. A release will then be created. The release tag must include
  - a. Some general notes on the changes that have occurred
  - b. A list of all the User stories and their subtasks that were added since the previous release

### Bugs In production

1. If any issues are found in the production environment, the deployment team **must** be consulted. A hotfix branch will then be made from the master branch by the relevant developer. Once the fix is ready, a pull request is made where **at least one** reviewer needs to be on the deployment team.
2. Once the hotfix is approved it is merged to both **master** and **develop** branches. Bug report can be seen [here](#).

# Testing

Testing aims at assuring the functionalities of the product fit our client's expectation and the product being delivered is of high quality.

Refer to the following pages for details of our test procedure, and test results:

# Back-End Unit Testing/Integration Testing

## Back-End Test Plan

### Goal:

Testing of the features and overall performance on the Back-End side. The elements to test are server and database.

### Post-condition:

Report to the back-end lead about the test results. Approve a pull request to the develop branch if the test is successful.

### Testing framework:

Jest framework is used for the unit testing of the software. The information on how to use the framework can be found [here](#).

Class	API	Method	Pre-Route	Route
game-controller.js	newGame	post	/play	/new
	joinGame	post		/join/:gameID
	getPlayerStatus	get		/wait/:gameID
	getGameStatus	get		/:gameID/status/
	getBackground	get		background/:gameID
	getRolesInfo	get		/:gameID/roles/
	showAllRoles	get		/choose-role/:gameID
	chooseRoles	post		/choose-role/:gameID
	fetchDecision	get		/make-decision/:gameID
	makeDecision	post		/make-decision/:gameID
	readyToDiscuss	post		/:gameID/start-discussion/
	getRoundDescription	get		/:gameID/round/
	getRoundStatus	get		/:gameID/round-status
	getRoundOutcome	get		/:gameID/round-outcome
option-controller.js	getOption	get	/options	/
	newAllOptions	post		/
	dropOption	post		/drop
question-controller.js	getQuestion	get	/questions	/
	newQuestion	post		/
	dropQuestion	post		/drop
roles.js	getRole	get	/roles	/
	newRole	post		/
	getRolesIntro	get		/introduction

## Back-End Test Results

Test suite	Test case - functionality	Sprint 3 Results (Pass/Fail)	Sprint 4 Results (Pass/Fail)
game_model.test.js	create & save game successfully	Pass	Pass
	insert game successfully	Pass	Pass
play.test.js	connect to server	Pass	Pass
	insert options	Pass	Pass
	create new game	Pass	Pass
	choose available role	Pass	Pass

	choose unavailable role	Pass	Pass
	make decision	Pass	Pass
	get role information	Pass	Pass
	get outcome	Pass	Pass
role_model.test.js	create & save role successfully	Pass	Pass
	insert role successfully	Pass	Pass
	create & save user successfully	Pass	Pass
	insert user successfully	Pass	Pass

# Front-End Unit Testing

- Front-end test plan
  - Goal
  - Start time:
  - Post-condition:
  - Testing framework:
  - Why choose Jest?
  - Further resources
- Front-end unit testing plan
  - Example:
- Unit Test Results
  - Key
  - Test results

## Front-end test plan

### Goal

Testing of the features and overall performance on the Front-End side. The elements to test our interaction with Front-End elements, visual bugs and inconsistencies.

### Start time:

After completion of the feature implementation and making a pull request to the **develop** branch.

### Post-condition:

Report to the front-end lead about the test results. Approve a pull request to the develop branch if the test is successful.

### Testing framework:

Jest framework is used for the unit testing of the software. The information on how to use Jest can be found [here](#).

### Why choose Jest?

1. Jest can use its unique snapshot test function to automatically test common frameworks such as React by comparing snapshot files generated by UI code. In addition, Jest's test cases are executed in parallel, and only the tests corresponding to the changed files are executed, which improves the test speed.
2. Easy to install and configure, very easy to use, almost zero-configuration, can be run directly by npm command installation
3. Jest has built-in test coverage tool Istanbul, which can be opened by command or configured in more detail in the package.json file. Running Istanbul in addition to the terminal display test coverage, a coverage directory will be produced under the project, with a report of test coverage, so that we can clearly see the test of the branch code.
4. Integrated assertion library, no need to introduce third-party assertion library, and very support React component testing.

### Further resources

<https://jestjs.io/>

[https://www.youtube.com/results?search\\_query=jest](https://www.youtube.com/results?search_query=jest)

## Front-end unit testing plan

The following test cases are planned:

1. rendering the correct title of each page
2. display the correct cartoon background of the main page
3. check the button of each page is correct and displays ok
4. check the HTML render the right div class
5. check the HTML render the content
6. check the HTML render the correct router
7. check the HTML render the correct layout

8. check the HTML render the correct colour

Unit tests that are nice to have:

1. mimic the clickable button events
2. check the format of the returned data
3. checking system status

## Example:

### Home Page

1. When the "New Game" button is pressed does it go to the "NewGame" page
2. When the "Join Game" button is pressed does it go to the "JoinGame" page
3. When the "Help" button is press does a instructions popup appear
4. Does clicking "Got It" on the instructions popup cause it to disappear

### New Game Page

1. When the back arrow is pressed does it go to the home page
2. Is the Submit button only active when the name text box is filled
3. Is the round duration slider only active when the timed game checkbox is selected
4. When the submit button is pressed does it go to the "Waiting" page

### Join Game Page

1. When the back arrow is pressed does it go to the home page
2. Is the Submit button only active when both the gameId and name field is filled
3. When the submit button is pressed does it go to the "Waiting" page

## Unit Test Results

### Key

Sprint	Colour
2	
3	

### Test results

ID	Test Goal	Test Suite Path	Test Result
1	check if home page can render New Game button and it is clickable	Home.spec.js	pass
2	check if home page can render Join Game button and it is clickable	Home.spec.js	pass
3	check if home page can render Help Got it button and check whether it is clickable	Home.spec.js	pass
4	check if JoinGame page can display a left arrow button and it is clickable	JoinGame.spec.js	pass
5	check if NewGame page can display a left arrow button and it is clickable	NewGame.spec.js	pass
6	check Next button going to the chapterBackground page	ChooseYourRole.spec.js	pass
7	check Background page can get correct data from backend	BackGround.spec.js	pass
8	check Next button going to the Choose Your role page	BackGround.spec.js	pass
9	check Next button going to the IndividualBackground page	ChapterBackground.spec.js	pass
10	check Next button going to the Discussion page	IndividualBackground.spec.js	pass
11	check Next button going to the ChapterBackground page	RoleBackground.spec.js	pass
12	check Next button going to the background page	Waiting.spec.js	pass
13	check if menu button going to IndividualBackground page	Discussion.spec.js	pass
14	check if RoleOutcome Text class can show correctly	Outcome.spec.js	pass
15	check if Outcome page router works well	Outcome.spec.js	pass
16	check whether title and content exists	Ready.spec.js	pass
17	check whether title and class of reflection page shows correctly	Reflection.spec.js	pass

18	check whether All Decisions Table class show correctly	Reflection.spec.js	pass
19	check if exit game button works well	Reflection.spec.js	pass
20	check if timer shows	Discussion.spec.js	pass
21	check if time can change between personal and general questions	Discussion.spec.js	pass
22	check if icon can show for timer	Discussion.spec.js	pass
23	check animations works well	Home.spec.js	pass
24	check if tile shows well	Home.spec.js	pass
25	check new clouds and plane show well on background	Home.spec.js	pass

# Acceptance Testing

## Detailed test table and results

User Story ID	User story	Acceptance Criteria ID	Given	When	Then	Acceptance Test ID	Test*	Expected Result	Test Pass /Fail
US_01	As a teacher I want to be relatively simple for the decision-making process so the game is fast-paced, which should improve student engagement and outcomes.	AC_01	The teacher control the game time.	The student plays the game.	The game is fast-paced, which should improve student engagement and outcomes.	AT_01	Play the whole game and test the gaming time.	The game process is in 20 minutes.	✓
US_02	As a student I want to submit my option successfully so I can play the game without getting stuck.	AC_02	The student submit the option.	Submit the option of each questions.	The game continues to next question fluently without stuck.	AT_02	All Users click the submit bottom in question page.	Next question page is displayed	✓
US_03	As a student I want to get a overall report that shows decisions made by other users for each questions throughout the game so I can clearly see whether or not I achieved my character goals or not, and also reflect upon whether personal success, if attained, came at any cost to plot outcomes.	AC_03	The student submit the option.	Submit the option of each questions.	The student can see overall report of other players submitted.	AT_03	All Users click the submit bottom in question page.	Display whether game characters goals achieved and the overall report of other users decision for the question.	✓
US_04	As a student/teacher I want to be able to rejoin the game session so I can continue the game if it is interrupted.	AC_04	The student want to rejoin the game.	The game is interrupted and return to the index page.	The game can continue and locate the latest process.	AT_04	Quit the game and click the rejoin bottom to rejoin the game in the home page.	Rejoin the game and transform to the latest game process.	✓
US_05	As a student I want to be able to see all chosen decisions made for all of my personal and group decisions. so I can get some insight into what other group members decided, and recall any forgotten decisions made earlier in the game.	AC_05	The student sees all the options chosen.	The student reaches the outcome page.	The student can get some insight into what other group members decided, and recall any forgotten decisions made earlier in the game.	AT_05	Submit the final question and reach the outcome page	All chosen decisions are shown in the outcome page	Deleted
US_06	As a teacher I want to add a timer to limit the discussion time for each decision in game so I can make students put more focus on the contradictive information and force them to make decisions, which will improve their critical thinking of whether their decisions are right to do.	AC_06	The student checks how much time they left to answer the questions.	The student answers the questions.	The student puts more focus on the contradictive information and forced to make decisions in limited time.	AT_06	Enter the question's pages	A timer is working and shows how many times left	✓
US_07	As a student/teacher I want to have an uncluttered and appealing user interface, so I can have a better experience.	AC_07	The student creates a game.	The student plays the game.	The student can see different animations or changes when click button or choose an option.	AT_07	The animations and background are rendered successfully.	The webpage is user-friendly and attractive.	✓

# End-to-End Testing

## Background

End-to-end tests aim to replicate user behaviour to perform tasks in the deployed software.

## Tests performed

All tests were performed by running five instances of the game, to replicate five users playing the game on their own devices.

Workflow	Test Passed
1. User creates a game, and four other users join	✓
2. Five users each choose a unique role	✓
3. Read through some initial background information (general game context, plus role-specific info) at the start of the game	✓
4. Read through relevant information for each round (general information, plus role-specific info)	✓
5. Be presented with a question to discuss with the group	✓
6. Enter a preferred option for the decision	✓
7. Receive the result of the round after receiving the option from all five users	✓
8. View the in-game menu in-game, to see a list of other players	✓
9. Exit the game from the in-game menu	✓

# Bug Report

## List of bugs

ID	Description	Fixed
1	The backend server is not available by the frontend. The program can not go any further after users put in their names and click on the start button.	Yes
2	The database is null after the server is launched. When a query of questions is sent at the beginning of the game, the backend program crashes immediately, leaving frontend website stuck at the waiting page.	Yes
3	After the last question is answered by every user, the program fails to give an outcome as expected and leaves the webpage stuck at the waiting status.	Yes
4	At the end of a round when everyone has answered the question and the program is expected to switch to the next question, there is a chance that the backend stuck in the waiting process.	No

# Testing Approach

- 1 Purpose
- 2 Testing approach
  - 2.1 Guidelines
- 3 Test types
  - 3.1 Unit tests
  - 3.2 Integration tests
  - 3.3 End-to-end tests & user acceptance testing
  - 3.4 Exploratory testing
- 4 Testing methodology
- 5 Further reading

## Purpose

Software testing is an important part of the Verification phase of our [Quality Assurance plan](#).

This document outlines our approach for testing, and justifies our choice of strategy.

## Testing approach

We will follow an agile testing approach.

This primarily means that testing is performed by all developers, as part of their development work - unlike under traditional software development lifecycles, where testing would be a separate phase after the code is written and might be performed by separate team members. This helps us ensure we are deploying high-quality product at the end of each sprint. It ensures that all developers are responsible for code that is high-quality and stands up to tests.

Most importantly, it allows us to identify bugs and errors continuously throughout development, instead of waiting until the later stages of the project which may uncover unexpected issues and cause delays.

## Guidelines

1. All Code changes **should contain** a test in order to prove the correctness if the new code added. This includes adding tests to demonstrate correctness of new functionality or to verify a bugfix is working as expected.
2. To achieve an appropriate level of confidence in our system we should write tests where:
  - a. **Correct Input is used** (Should reasonably encase the set of valid inputs)
    - i. If your program takes in an array, write a test for the inputs of 0, 1, many values in array
    - ii. Use [Boundary Value Analysis & Equivalence Partitioning](#) concepts to help
  - b. **Incorrect Input is used** (Because programs *should* fail expectedly/gracefully)
3. Please aim to **at least** achieve **statement coverage** (if possible), that is your tests suite should execute each line at least once.

## Test types

A number of different test types are appropriate for the project and are described below. Further information on how these will be carried out is provided under ....

### Unit tests

Unit testing aims to evaluate individual pieces of code to ensure they function as intended. For this project, unit testing will be performed on each function in the code. They will be automated using appropriate testing frameworks for the front- and back-end, and can be run as part of continuous integration.

### Integration tests

Integration testing aims to evaluate that the different components of the software correctly interface together. The components of this software system include the database, back-end and front-end. Similar to unit tests, integration tests will be automated. We will employ a bottom-up approach - that is, two components will be tested at a time, before testing the integration of all the components together.

### End-to-end tests & user acceptance testing

**End-to-end testing** replicate a user's behaviour to use our project in a realistic scenarios. We will investigate the use of testing frameworks that can automate these tests, otherwise these can be performed manually (sparingly).

As user stories are completed during a sprint, the corresponding [User Stories & Acceptance Criteria](#) will be performed manually to check if the software meets the requirements.

## Exploratory testing

Exploratory testing is a less structured approach where an individual tester, rather than formally designing test cases, simply uses the system and attempts to discover bugs on-the-fly, taking into consideration common weaknesses and potential faults. This practice exercises the judgement of the developer and can help discover errors that are not uncovered by automated testing. This will be performed towards the end of each sprint.

## Testing methodology

Automated test cases, in particular, will mainly be documented in the repository source code.

Manual test cases will be documented in Confluence. Tests are to be documented fully, including for each test case a description of the reason for the test case and choice of test input, and the expected output. When performing the testing, sufficient information must be captured from the input and output must be captured, such that the test is easily reproducible with exactly the same input.

Any failures in tests are to be documented before making fixes to the code.

When testing, the developer/tester should fix any straightforward, simple bugs. Any complicated or unsolvable bugs identified during testing are to be communicated to the team and logged on Jira.

## Further reading

- [https://en.wikipedia.org/wiki/Agile\\_testing](https://en.wikipedia.org/wiki/Agile_testing)
- <https://www.guru99.com/agile-testing-a-beginner-s-guide.html>
- <https://www.atlassian.com/continuous-delivery/software-testing/types-of-software-testing>
- <https://www.atlassian.com/continuous-delivery/software-testing/exploratory-testing>