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 contradictive 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 senario 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

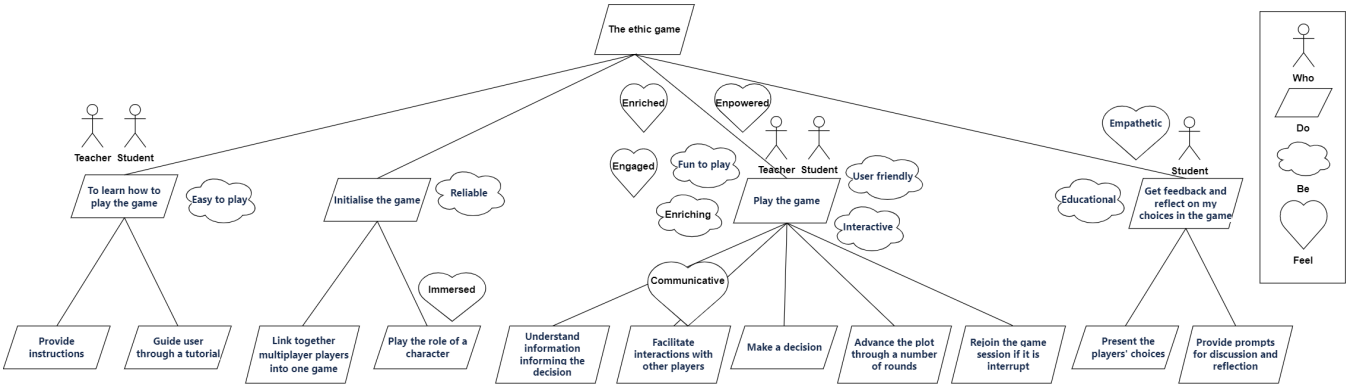
The basic goal was the need to successfully deploy the game to the cloud, fix situations where the game would get stuck halfway through and other bug fixes. Shorten the whole game play process by reducing the number of in-game issues. Improve the user's gaming experience. Test the entire game process 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	Enriched
Teacher		Guide user through a tutorial	Easy to play	Empowered
	Initialise the game	Link together multiplayer players into one game	Fun to play	Engaged
		Play the role of a character	User friendly	Empathetic
	Play the game	Understand information informing the decision	Interactive	Immersed
		Facilitate interactions with other players	Reliable	Communicative
		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

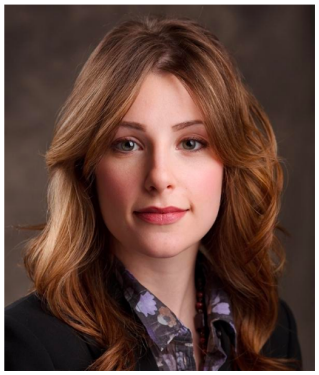
- study ethics
- study engineering

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

Introvert	Extrovert
Thinking	Feeling
Sensing	Intuition
Judging	Perceiving

Skills

- knowledge of ethics
- Programming Ability
- Logical thinking
- Communication 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

Introvert — Extrovert
Thinking — Feeling
Sensing — Intuition
Judging — Perceiving

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

knowledge of ethics
Programming Ability
Logical thinking
Communication 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

Introvert	Extrovert
Thinking	Feeling
Sensing	Intuition
Judging	Perceiving

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 ehcis 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

- knowledge of ethics
- Programming Ability
- Logical thinking
- Communication Skills

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?

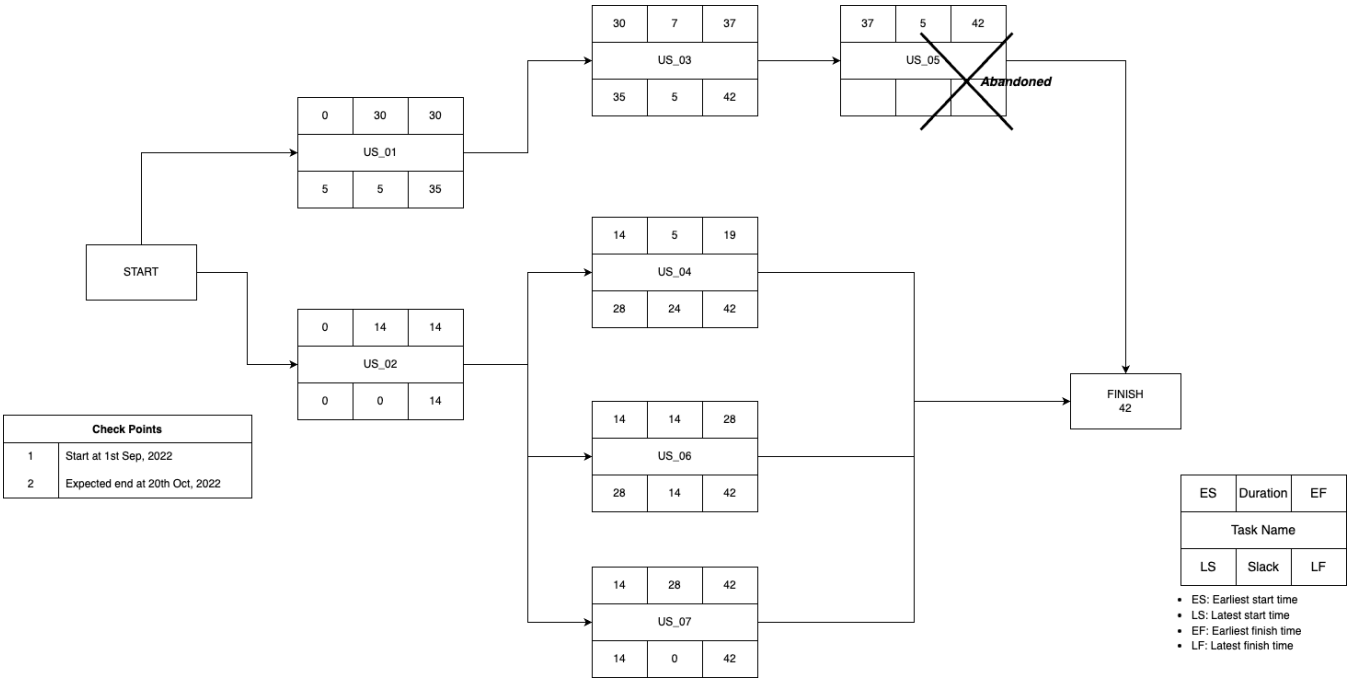
As probable expansions of the system, the following requirements should not be taken into consideration.

- 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	Sprint	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 questions 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.	✅

General Process Documentation

General Process Documentation contains:

- Roles and Responsibilities
- Risk Management

Roles and Responsibilities

We do rotation of roles for each sprint.

Sprint 1 Roles

	Role	Person	Responsibility
1	Scrum Master	Rainer	• Facilitate teamwork, help remove impediments, and promote agile principles • Plan meetings and ceremonies • Jira reporting
2	Product Owner	Angus	• Represent interests of the client and maximise value • Maintain product backlog • Client liaison: Handles communication with the client
3	Quality Manager	Akhmetzhan	• Make sure confluence is up-to-date. • Check the consistency of code. • Ensuring high quality commenting and documentation.
4	Deployment Lead	Eric	• Documentation of how the product runs • Leads the Dev-Ops team • Integrate the frontend and backend
5	Architecture Lead	Chenling	• Design components, patterns and interfaces • Make decisions regarding architecture and design of software
6	Front-End Testing Lead	Peiwen	• Create testing strategy for front-end • Make sure the display of the project (GUI) meets the requirement
7	Back-End Testing Lead	Yu	• Create testing strategy for back-end • Make sure the database works correctly
8	UI Design Lead	Christina	• Create consistent style and theme for the product • Design the UI, ensuring high-quality visual appearance and usability
9	Front-End Lead	Yuhen	• Coordinate the front-end team • Deliver the updates of sub-team
10	Back-End Lead	Lu	• Coordinate the back-end team • Deliver the updates of sub-team

Risk Management

Risk ID	Risk Description	Consequence	Probability	Severity	Impact	Mitigation Strategy

Communication Plan

Communication	Method	Frequency	Goal	Owner	Audience
Stand-Up Meeting	Zoom	Weekly	▪ Ensure accountability, that is, all team members are making meaningful contributions ▪ Give all sub-team members updates about state of other teams ▪ Ensure JIRA and Confluence spaces are up-to-date	Scrum Master	Internal Team + Supervisor
Supervisor Meeting	Zoom	Weekly	▪ Ensure team is producing necessary artefacts and performing at sufficient velocity to achieve a strong project outcome ▪ Cross-check team expectations with supervisor expectations	Scrum Master	Internal Team + Supervisor
Sprint Planning	Zoom	Once Every 3 Weeks	▪ Task breakdown of user story epics into actionable points ▪ Story point estimation for each subtask, and initial allocation of responsibility	Scrum Master	Internal Team
Sprint Retrospective	Zoom	Once Every 3 Weeks	▪ Reflection and review of strengths and weaknesses of sprint ▪ Discussion of actionable points to work on for next sprint, and how to action them	Scrum Master	Internal Team
Sprint Review	Zoom	Once Every 3 Weeks	▪ Formal client and supervisor review of sprint artefacts ▪ Feedback and discussion regarding team project outcomes compared to client expectations	Communication Manager	Internal Team + Supervisor + Client
Client Correspondence	Email	Once Every 2 Weeks	▪ Ensure client is kept up-to-date about any progressions the team is making ▪ If necessary, arrange meetings and discussions	Communication Manager	Communication Manager + Supervisor + Client

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, corns, and pros.

Related Games

This page shows some real-world examples of games related to formate for selection and discussion.

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

Related Games

Platform

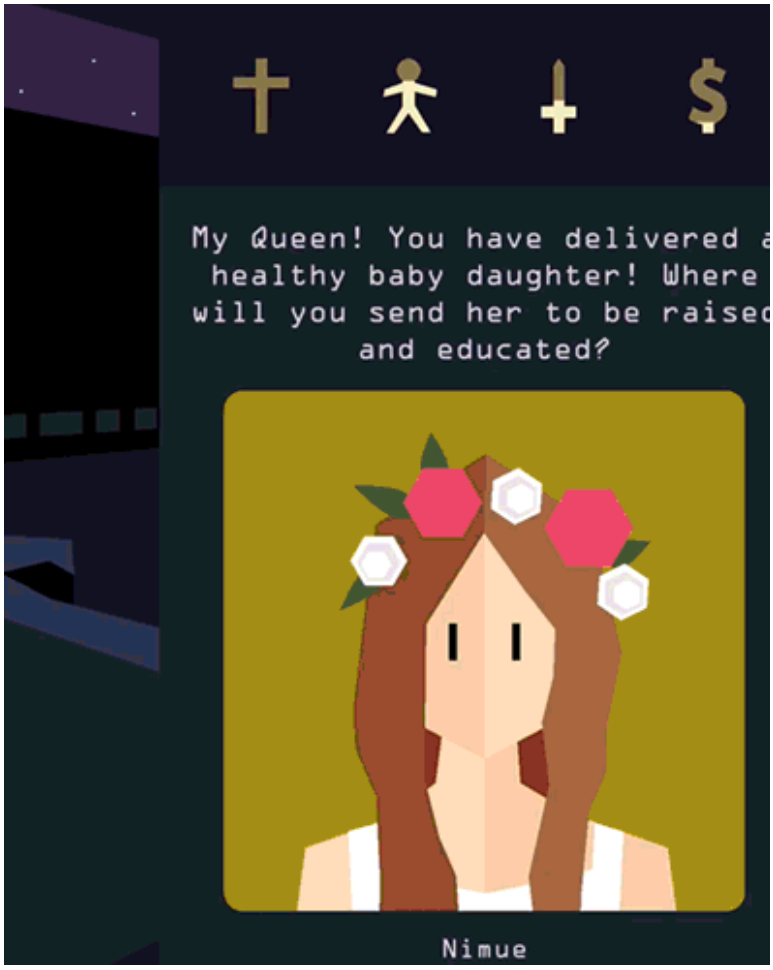
Proposed gameplay designs

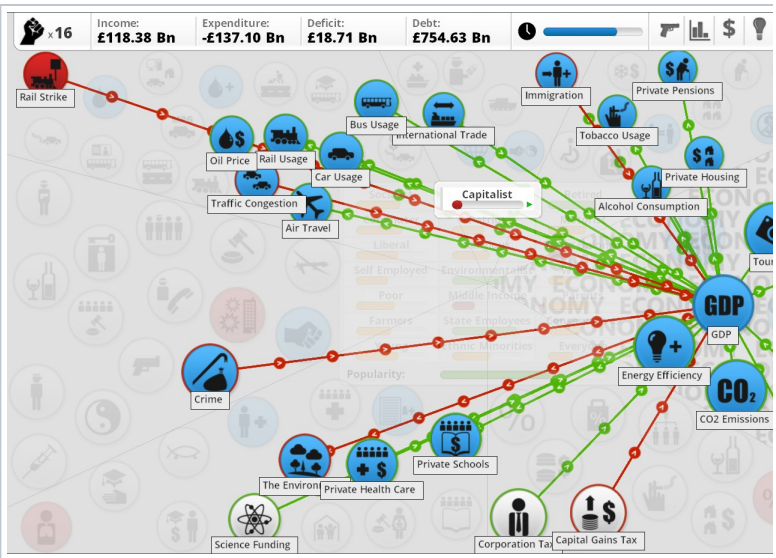
Dev-Ops Tools

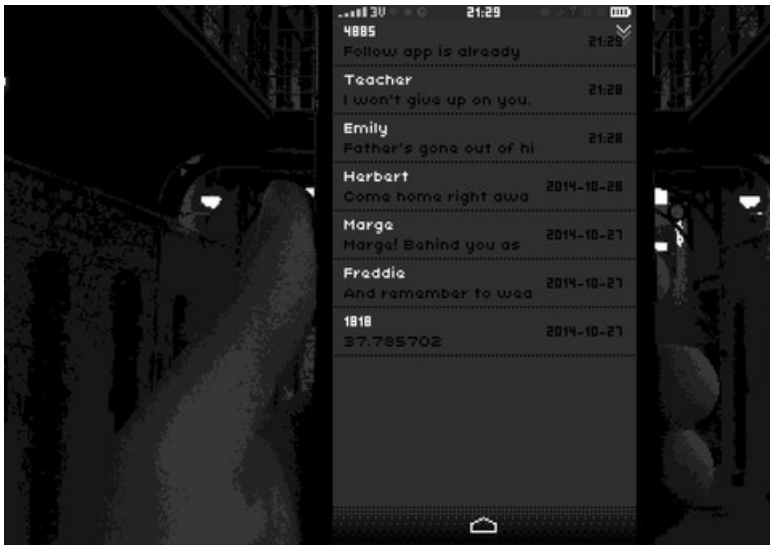
Game Formats

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.	• Allows us to focus on a deep story line as development is simplified	• May potentially make decisions seem less important • Format doesn't hide that the game is decision based	• Reigns
Interactive /Adventure	User makes decisions by clicking among choices. Choices will then change the environment	• A dynamic environment can make choices feel more meaningful • Multiple ends • More close to the real world	• Can potentially limit depth of story line as development time will need to be dedicated to visuals • Heavily depend on the story script	• Talk To strangers • Replica • Beholder • Detroit: Become Human
Turn-Based Strategy	Players take turns playing (Making choices). Strategy games tend to have more complicated metrics systems	• 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	• Learning curve for users to play game compared to other formats • May take a long time for each round	• Democracy 3 • Sid Meier's Civilization VI • Chess • Werewolf
SIM (Simulation Game) **building or managing**	The player will play a role in an aspect of reality and control the development.	• Combine the role play and strategy • Engage players more	• No specific ending and rewards	• SimCity
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.	• Combine the role play and strategy • The player has a strong sense of substitution • Based on the player's view • The player has overall image of the game structure	• Too much flexibility may add complexity	• Detroit: Become human

Related Games

Game	Format of the game	Game description	Advantages	Disadvantages	Visual
Reigns	Card Based Game	As royalty, make decisions on the kingdom by swiping left or right on cards.	• Clean, Simple interface (players stats are represented by symbols above cards) • Simple operation • Short playing time for each round of the game • Visualized rewards and punishment	• Good UI design needed to attract players	
Talk To strangers	Interactive	**Talk to Strangers** is a short game of everyday survival. Survive door to door conversations ranging from bizarre to mundane (in this order) while trying to sell products to **strangers** **Source:** https://store.steampowered.com/app/963280/Talk_to_Strangers/	• Pixel art is simplified game development • Simple operation • Could have multiple ends.	• Good UI design needed to attract players • Good script needed • May take a long time for multiple users to make decisions turn by turn	

Democracy 3	**Turn-Based Strategy**	In **Democracy 3**you are a country leader who just won the election, and you need to take a series of measures to make sure your people live good lives so that you could win in the next election.	• see the influences of decisions directly • Visualized rewards and punishment	• Complex User Interface • Long time for each round • Complex play rules • Complex decision tree • Limited outcomes • Single player	
Sid Meier's Civilization VI	**Turn-Based Strategy**	You are a country leader, and your aim is to build your empire and make it become the strongest country in the whole world.	• Multiple players	• Long time for each round • 2 outcomes: Succeed or Fail • Complex play rules	

Replica	**Interactive** **First Person Perspective** **Role Play** **Detective**	You are given a cellphone of an unknow owner. You must look for evidences of terrorism by hacking into the cellphone owner's account, under governmental coercion by inspecting the cellphone usage history and social media activity records (Stream game description). Otherside you will be accused of terrorism.	• Multiple ends • Simple interface • Simple play rules • Real situation stimulation (experience moral dilemma when making choices)	• Single player • Good UI design needed • Good story script needed • No visualized rewards or punishment	  https://steamcdn-a.akamaihd.net/steam/apps/256668436/movie480.webm?t=1470670633
Beholder (1 & 2)	**Interactive** **Third Person Perspective** **Role Play** **Strategy**	You're a Stat-installed Landlord in a totalitarian State. You must spy on tenants, peep, eavesdrop and profile! You must report on anyone capable of plotting subversion against the State. You MUST! But WILL you? (Stream game description) You will experience moral dilemma when you spy other's life. You have to make choice whether to offer a hand or not when other face problems in the risk of your own life.	• Multiple ends • Visualized rewards and punishment • Simple play rules • Real situation stimulation and have time limited for some actions • Visualized rewards and punishment	• Single player • Require long time for each round • Good story script needed	 https://steamcdn-a.akamaihd.net/steam/apps/256672664/movie480.webm?t=1476703328

Werewolf	**Card game** **Turn-Based Strategy**	**Werewolf** is a multiplayer game. Use special abilities to uncover the roles of other players.	• Simple UI	• Multiplayer game interaction (requires sharing data and parallel interaction problems during turn-based games)	
SimCity (2013)	**SIM (Simulation Game)** **Role Play**	Player(s) will act as the leader(s) of a city. They can make the rank of the city grow by making decisions of policy, culture, transport, industry and dealing with the crisis	• Managing a city is close to managing a company. So we can reference it	• Need complicated design of situations and issues in the game. • Good UI needed	
Detroit: Become human	**First Person Perspective**	*Detroit*: *Become Human* is an adventure game played from a *third-person* view. There are multiple playable characters. The game is broken into epsidodes. Source: https://en.wikipedia.org/wiki/Detroit:_Become_Human Detroit 2038. Technology has evolved to a point where human like androids are everywhere. They speak, move and behave like human beings, but they are only machines serving humans.	• Episodic, the game is broken in to smaller pieces (may want to structure game like this). • Each episode focuses on a particular character, however choices made by one character can influence all (This allows simplicity by avoiding interactions between characters but also keeps the choices meaningful)	• Graphics are complex (not within time and skill constraints)	

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 finial 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 ther 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 assymmetry).
- 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 evem 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 simplifed 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 assymmetry 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 assymmetry).

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 descisions
- Game is structures 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 being flying one those things tomorrow

Work colleague: ahh, 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.

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	• Powerful event-based system • Easy to operate				
GameMaker Studio 2	Dragging (no language skill needed) OR C	free for development paid for publishing	• Extensive integration toolkit	• paid for publishing			
Unity	C#	free	• Full-featured • Much information and tutorials • Extensible (Customizable)	• Unfriendly to a beginner • More resource needed (CPU, GPU...)			
Godot	GDScript (similar to python) OR C++ (can be imported as GDScript)	free	• 3D and 2D, better in 2D • Multi-platform • Visual editor • Lightweight • Easier animation and scene system	• less tutorialdemo plugin template	asset lib (Godot Asset Library)	• Spooky Ghost Dotosts Dot Com (Steam) • Grimante (Steam)	
Unreal Engine 4 (UE4)	C++ (Encapsulated via Epic, less difficult than C++)	free for studying purpose	• 3D 2D • better for multithreading • better effects	• Hard to start • More complex logic needed			
cocos2d	C++, Lua or JavaScript	free	• Lightweight and efficient cross-platform • Develop for multiple platforms: OSAndroidHTML5 PC • Open-source • 2D	• Long compilation time • Need time to learn			
Phaser	JavaScript	free	• Friendly for beginners • Opensource codabase • Keep together WebGL and Canvas components				https://phaser.io

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.

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 allows 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.	• Easy to learn and use for beginners • Tiny size • Virtual DOM rendering and performance • Reactive two-way data binding • single file component and readability • Concise documentation	• Lack of support for large-scale projects • Reactivity complexity
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.	• Powerful framework • Virtual DOM • Open source facebook library • Widely-used, in-demand skill	• Difficult to set-up and use • Documentation is only adequate

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.

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.	▪ Extensive concurrency, networking and GUI support ▪ As a low-level programming language, it is extremely flexible and allows us to customise in almost any conceivable way ▪ Extensive libraries and documentation for building back-end systems with Java exist	▪ Using this means that there will be a disconnect between front-end language and back-end language, which complicates integration ▪ More difficult to build systems, since lower-level languages lack the constructs necessary to speed development
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.	▪ Can be utilised to write BOTH front-end and back-end code, it allows for use of the same Javascript code for both ends ▪ Server-side capabilities are extensively provided, Node.js was originally meant for back-end development ▪ An event-based model to address scalability, and rich libraries to simplify coding ▪ Much faster and much more scalable than Java	▪ Less flexibility, we can only work within the capabilities Node.js provides ▪ Heavy reliance on third-party libraries leads to reliability risk, what if a library is bugged? ▪ Weaker concurrency constructs than Java

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	• Developers don't need to learn a new tool	• Ability to have a consistent environment across all developers is limited even with documentation
Docker	Use Containers (essentially a lightweight virtual machine) during development, developers use the containerised services	• Developers will need to learn how to set up the software and run containers	• Allows for fine-grained control of the development environment (version of tools)
Virtual Machine	A virtual machine is an emulation of a computer system.	• Full control over the environment (more so than docker)	• Higher performance overhead • More complex to set up and use than other solutions

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	• Strong Integration with Bitbucket (since they are developed by the same company) • Most likely freely available for the project (will have to check if Melbourne Uni provides us with the tool)	• Unsure if unimelb provides hosted bamboo (the service appears to have no recent activity)

Jenkins	Open source automation server designed for CI/CD processes	• Has a large community backing
		  • Many plugins
		  • Well Documented
			• Requires us to set up our own infrastructure for compiling and testing builds

CircleCi	CI/CD tool, Closed source.	• Has "orbs" (Preconfigured templates for integrating with many services)	• Limited feature set on the free tier
			  • Can only run 1 job at a time
			  • Has a credits system (limited number of builds per month)

BuildKite	A platform for running fast, secure, and scalable continuous integration pipelines on your own infrastructure.	• Unlimited number of builds and number of concurrent jobs on the free tier	• Requires us to set up our own infrastructure for compiling and testing builds
Travis CI	Travis CI is a hosted continuous integration service used to build and test software projects hosted at GitHub and Bitbucket.	• Easy setup (compared to other tools)	• Free only for public repositories, closed repositories have a 1-month free trial
Gitlab	GitLab is a web-based DevOps lifecycle tool that provides a Git-repository manager providing wiki, issue-tracking and continuous	• Has an integration with Bitbucket	

	integration/continuous deployment pipeline features	• Offers a free tier cloud-hosted solution	• 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.	Offers GitHub integration through a tool called "Cloud Deploy". We would need to further investigate how we deploy the application (docker image on ec2 instance)	Extremely Flexible. AWS offers services for many technologies	AWS offers both always free and 12 month free tiers for there services. The Limitations depend on the service. For example, EC2 instances offer 750 hours free per month for 12 months. For info here
Heroku	Platform as a service. Built on top of AWS.	Yes, Has GitHub integration built-in. Heroku will automatically install dependencies and run the application.	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 here.	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 here	$300 credit for 12 months. In addition, they offer free tiers for their services. More info here. 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.

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	Price
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.	• Supports features like Master-Slave Replication, Scale-Out • Query Cache for repeatedly used statements • You can easily learn and troubleshoot MySQL from different sources like blogs, white papers, and books.	• Transactions related to system catalog are not ACID compliant • Sometimes a server crash can corrupt the system catalog • Stored procedures are not cacheable • MYSQL tables which is used for the procedure or trigger are most pre-locked.	
	SQL Azure	Microsoft Azure SQL Database is a managed cloud database provided as part of Microsoft Azure.	• High Availability • Data Security • Scalability	• Requires Management • Requires Platform Expertise	
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.	• No schema • Fast	• Transactions using MongoDB are complex • 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.	

Testing Frameworks

JavaScript testing frameworks	Introduction	Advantage
MochaJS	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.	• Provides compatibility for both frontend and backend testing • NodeJS debugger is supported which makes error tracing easier • Accurate reporting • Provides support for all browsers including the headless Chrome library • Very convenient framework for the developers to write test cases
JEST	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.	• Compatible with NodeJS, React, Angular, VueJS and other Babel based projects • Standard syntax with documentation support • Very fast and highly performant • Managing tests with larger objects is possible using Live Snapshots
Jasmine	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.	• Provides small, clean and straightforward syntax for easy testing • Does not require any Document Object Model (DOM) • Provides support for both frontend and backend tests • Ease in coding as the syntax used is very similar to a natural language • Strong documentation and community support

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

Characters

Plot

Decision Tree

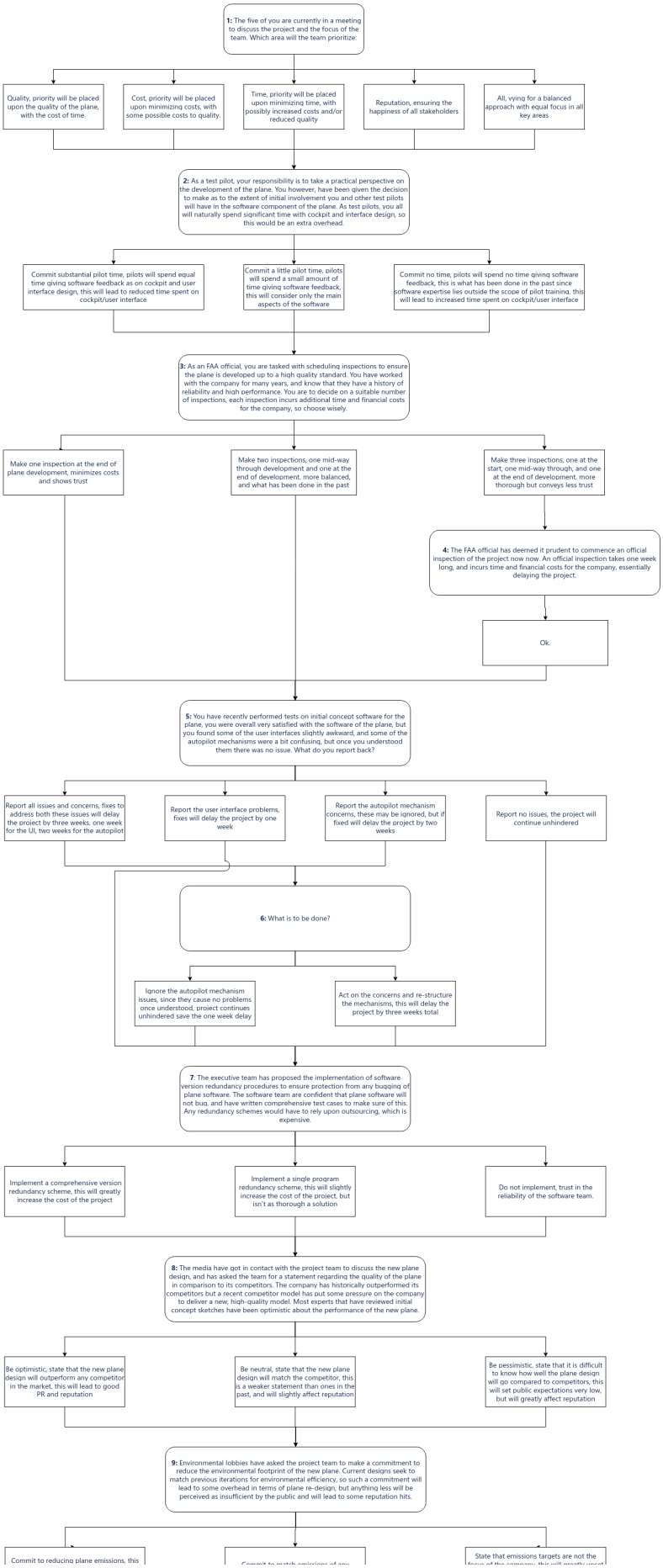
The initial decision tree consists of 27 questions. The game would not be end until every player answeres all questions. Moreover, there is no limit to the discussion time of questions, which lead to the game is extremely long with over 30 minutes per round.

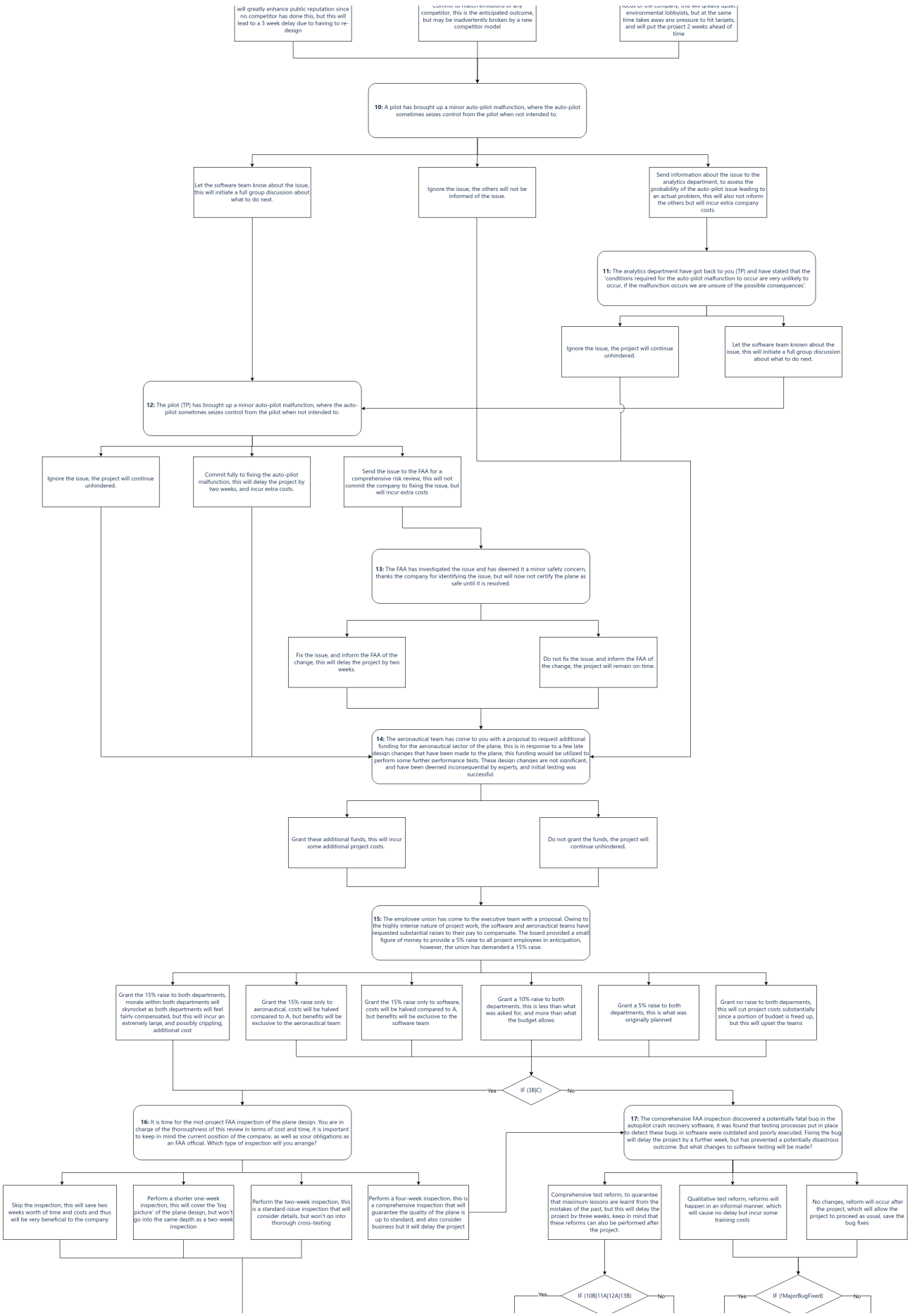
The simplification process of the decision tree is shown below:

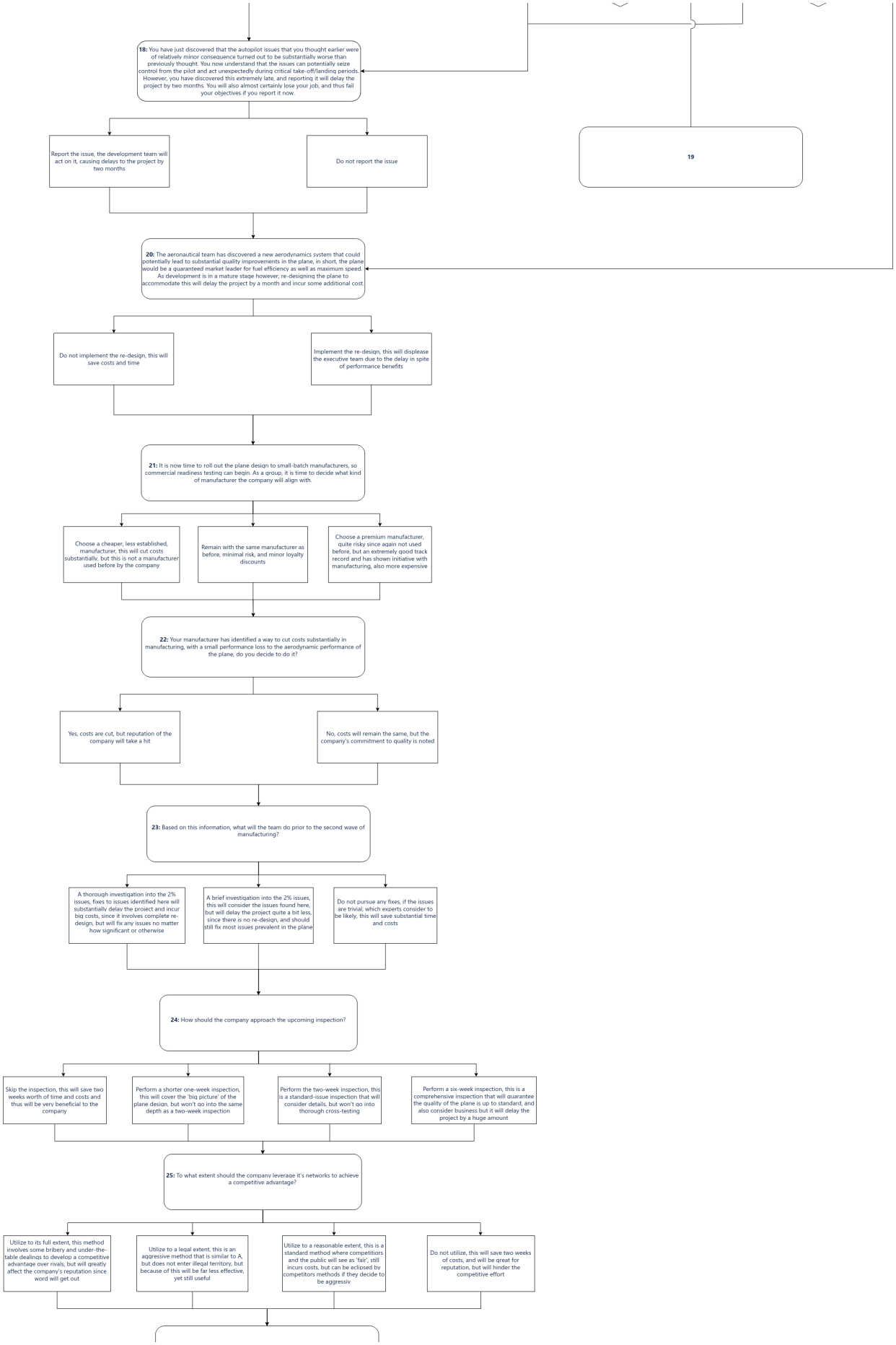
Simplification methods	Question Number	Reasons
Delete questions	1, 4	Question 1 is about a tip of the game to help bring players into the situation, which is not essential to the result of the game. Question 4 only has one choice "OK", which has no effect on game progression.
Personal answered questions (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.

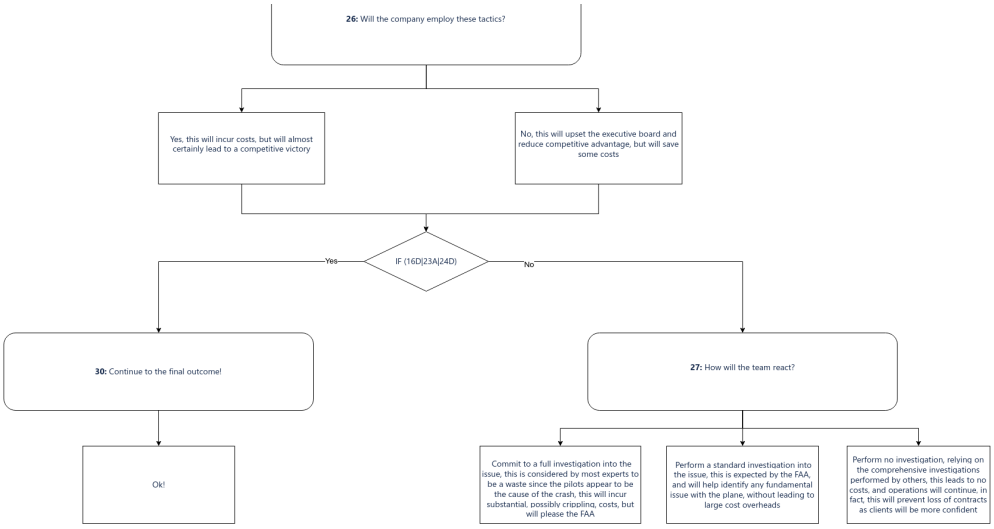
Current decision tree consists of 25 questions. Each players are requested to answer 4 to 5 personal questions and 8 group discussion questions. Play time is limited to 20 minutes.

Decision Tree (version 1)


1: The five of you are currently in a meeting to discuss the project and the focus of the team. Which area will the team prioritize:
Quality, priority will be placed upon the quality of the plane, with the cost of time.
Cost, priority will be placed upon minimizing costs, with some possible costs to quality.
Time, priority will be placed upon minimizing time, with possibly increased costs and/or reduced quality
Reputation, ensuring the happiness of all stakeholders
All, vying for a balanced approach with equal focus in all key areas
2: As a test pilot, your responsibility is to take a practical perspective on the development of the plane. You however, have been given the decision to make as to the extent of initial involvement you and other test pilots will have in the software component of the plane. As test pilots, you all will naturally spend significant time with cockpit and interface design, so this would be an extra overhead.
Commit substantial pilot time, pilots will spend equal time giving software feedback as on cockpit and user-interface design, this will lead to reduced time spent on cockpit/user interface
Commit a little pilot time, pilots will spend a small amount of time giving software feedback, this will consider only the main aspects of the software
Commit no time, pilots will spend no time giving software feedback, this is what has been done in the past since software expertise lies outside the scope of pilot training, this will lead to increased time spent on cockpit/user interface
3: As an FAA official, you are tasked with scheduling inspections to ensure the plane is developed up to a high quality standard. You have worked with the company for many years, and know that they have a history of reliability and high performance. You are to decide on a suitable number of inspections, each inspection incurs additional time and financial costs for the company, so choose wisely.
Make one inspection at the end of plane development, minimizes costs and shows trust
Make two inspections, one mid-way through development and one at the end of development, more balanced, and what has been done in the past
Make three inspections, one at the start, one mid-way through, and one at the end of development, more thorough but conveys less trust
4: The FAA official has deemed it prudent to commence an official inspection of the project now now. An official inspection takes one week long, and incurs time and financial costs for the company, essentially delaying the project.
Ok.
5: You have recently performed tests on initial concept software for the plane, you were overall very satisfied with the software of the plane, but you found some of the user interfaces slightly awkward, and some of the autopilot mechanisms were a bit confusing, but once you understood them there was no issue. What do you report back?
Report all issues and concerns, fixes to address both these issues will delay the project by three weeks, one week for the UI, two weeks for the autopilot
Report the user interface problems, fixes will delay the project by one week
Report the autopilot mechanism concerns, these may be ignored, but if fixed will delay the project by two weeks
Report no issues, the project will continue unhindered
6: What is to be done?
Ignore the autopilot mechanism issues, since they cause no problems once understood, project continues unhindered save the one week delay
Act on the concerns and re-structure the mechanisms, this will delay the project by three weeks total
7: The executive team has proposed the implementation of software version redundancy procedures to ensure protection from any bugging of plane software. The software team are confident that plane software will not bug, and have written comprehensive test cases to make sure of this. Any redundancy schemes would have to rely upon outsourcing, which is expensive.
Implement a comprehensive version redundancy scheme, this will greatly increase the cost of the project
Implement a single program redundancy scheme, this will slightly increase the cost of the project, but isn't as thorough a solution
Do not implement, trust in the reliability of the software team.
8: The media have got in contact with the project team to discuss the new plane design, and has asked the team for a statement regarding the quality of the plane in comparison to its competitors. The company has historically outperformed its competitors but a recent competitor model has put some pressure on the company to deliver a new, high-quality model. Most experts that have reviewed initial concept sketches have been optimistic about the performance of the new plane.
Be optimistic, state that the new plane design will outperform any competitor in the market, this will lead to good PR and reputation
Be neutral, state that the new plane design will match the competitor, this is a weaker statement than ones in the past, and will slightly affect reputation
Be pessimistic, state that it is difficult to know how well the plane design will go compared to competitors, this will set public expectations very low, but will greatly affect reputation
9: Environmental lobbies have asked the project team to make a commitment to reduce the environmental footprint of the new plane. Current designs seek to match previous iterations for environmental efficiency, so such a commitment will lead to some overhead in terms of plane re-design, but anything less will be perceived as insufficient by the public and will lead to some reputation hits.
Commit to reducing plane emissions, this
State that emissions targets are not the

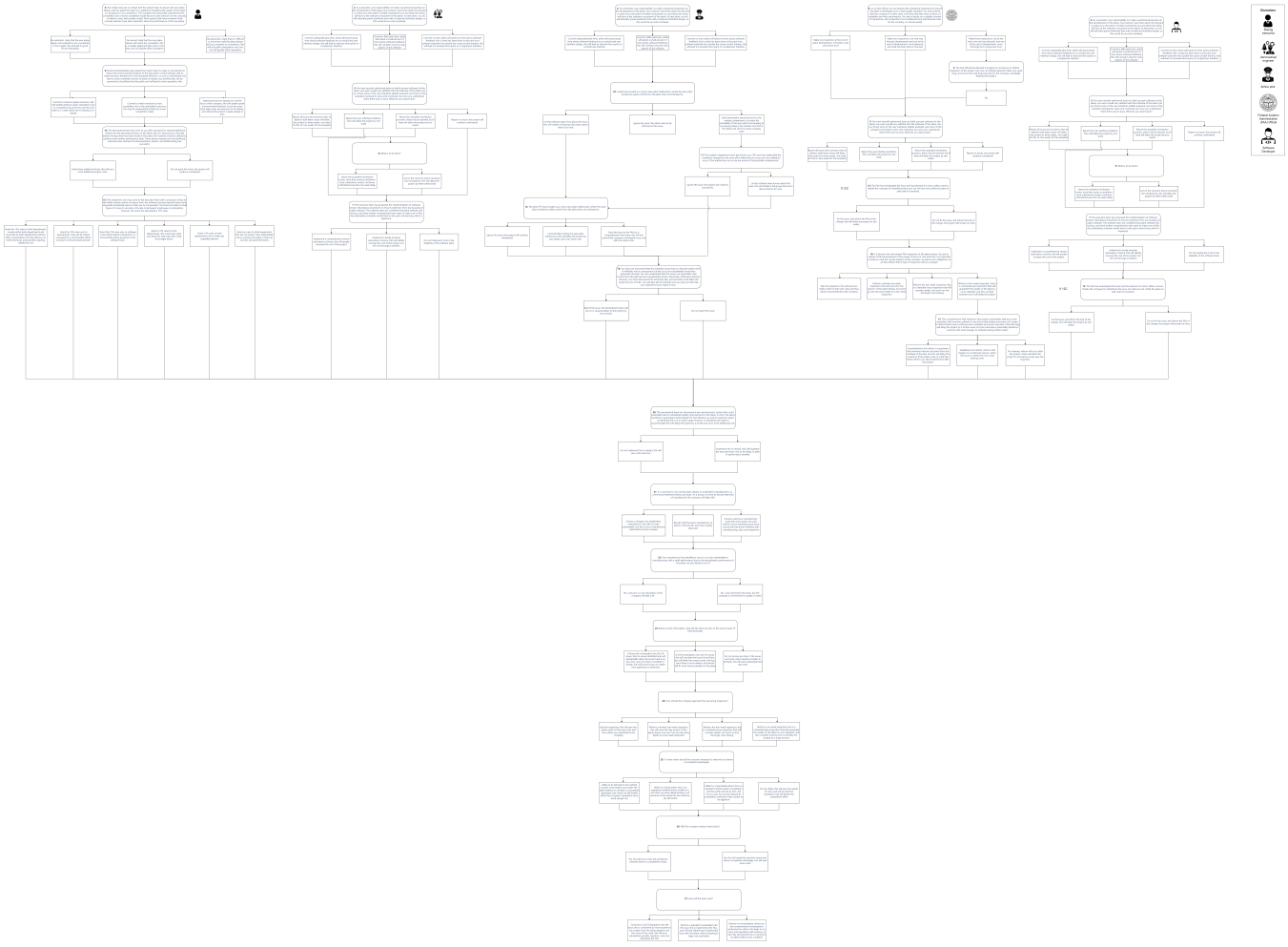

will greatly enhance public reputation since no competitor has done this, but this will lead to a 3 week delay due to having to re-design
competitor, this is the anticipated outcome, but may be inadvertently broken by a new competitor model
environmental lobbyists, but at the same time takes away any pressure to hit targets, and will put the project 2 weeks ahead of time
10: A pilot has brought up a minor auto-pilot malfunction, where the auto-pilot sometimes seizes control from the pilot when not intended to.
Let the software team know about the issue, this will initiate a full group discussion about what to do next.
Ignore the issue, the others will not be informed of the issue.
Send information about the issue to the analytics department, to assess the probability of the auto-pilot issue leading to an actual problem, this will also not inform the others but will incur extra company costs.
11: The analytics department have got back to you (TP) and have stated that the 'conditions required for the auto-pilot malfunction to occur are very unlikely to occur, if the malfunction occurs we are unsure of the possible consequences'.
Ignore the issue, the project will continue unhindered.
Let the software team known about the issue, this will initiate a full group discussion about what to do next.
12: The pilot (TP) has brought up a minor auto-pilot malfunction, where the auto-pilot sometimes seizes control from the pilot when not intended to.
Ignore the issue, the project will continue unhindered.
Commit fully to fixing the auto-pilot malfunction, this will delay the project by two weeks, and incur extra costs.
Send the issue to the FAA for a comprehensive risk review, this will not commit the company to fixing the issue, but will incur extra costs
13: The FAA has investigated the issue and has deemed it a minor safety concern, thanks the company for identifying the issue, but will now not certify the plane as safe until it is resolved.
Fix the issue, and inform the FAA of the change, this will delay the project by two weeks.
Do not fix the issue, and inform the FAA of the change, the project will remain on-time.
14: The aeronautical team has come to you with a proposal to request additional funding for the aeronautical sector of the plane, this is in response to a few late design changes that have been made to the plane, this funding would be utilized to perform some further performance tests. These design changes are not significant, and have been deemed inconsequential by experts, and initial testing was successful.
Grant these additional funds, this will incur some additional project costs.
Do not grant the funds, the project will continue unhindered.
15: The emplovee union has come to the executive team with a proposal. Owing to the highly intense nature of project work, the software and aeronautical teams have requested substantial raises to their pay to compensate. The board provided a small figure of money to provide a 5% raise to all project employees in anticipation, however, the union has demanded a 15% raise.
Grant the 15% raise to both departments, morale within both departments will skyrocket as both departments will feel fairly compensated, but this will incur an extremely large, and possibly crippling, additional cost
Grant the 15% raise only to aeronautical, costs will be halved compared to A, but benefits will be exclusive to the aeronautical team
Grant the 15% raise only to software, costs will be halved compared to A, but benefits will be exclusive to the software team
Grant a 10% raise to both departments, this is less than what was asked for, and more than what the budget allows
Grant a 5% raise to both departments, this is what was originally planned
Grant no raise to both deparments, this will cut project costs substantially since a portion of budget is freed up, but this will upset the teams
IF (3B|C)
Yes
No
16: It is time for the mid-project FAA inspection of the plane design. You are in charge of the thoroughness of this review in terms of cost and time, it is important to keep in mind the current position of the company, as well as your obligations as an FAA official. Which type of inspection will you arrange?
17: The comprehensive FAA inspection discovered a potentially fatal bug in the autopilot crash recovery software, it was found that testing processes put in place to detect these bugs in software were outdated and poorly executed. Fixing the bug will delay the project by a further week, but has prevented a potentially disastrous outcome. But what changes to software testing will be made?
Skip the inspection, this will save two weeks worth of time and costs and thus will be very beneficial to the company
Perform a shorter one-week inspection, this will cover the 'big picture' of the plane design, but won't go into the same depth as a two-week inspection
Perform the two-week inspection, this is a standard-issue inspection that will consider details, but won't go into thorough cross-testing
Perform a four-week inspection, this is a comprehensive inspection that will guarantee the quality of the plane is up to standard, and also consider business but it will delay the project
Comprehensive test reform, to guarantee that maximum lessons are learnt from the mistakes of the past, but this will delay the project by three weeks, keep in mind that these reforms can also be performed after the project.
Qualitative test reform, reforms will happen in an informal manner, which will cause no delay but incur some training costs
No changes, reform will occur after the project, which will allow the project to proceed as usual, save the bug fixes
IF (10B|11A|12A|13B)
Yes
No
IF (!MajorBugFixed)
Yes
No


18: You have just discovered that the autopilot issues that you thought earlier were of relatively minor consequence turned out to be substantially worse than previously thought. You now understand that the issues can potentially seize control from the pilot and act unexpectedly during critical take-off/landing periods. However, you have discovered this extremely late, and reporting it will delay the project by two months. You will also almost certainly lose your job, and thus fail your objectives if you report it now.
Report the issue, the development team will act on it, causing delays to the project by two months
Do not report the issue
19
20: The aeronautical team has discovered a new aerodynamics system that could potentially lead to substantial quality improvements in the plane, in short, the plane would be a guaranteed market leader for fuel efficiency as well as maximum speed. As development is in a mature stage however, re-designing the plane to accommodate this will delay the project by a month and incur some additional cost.
Do not implement the re-design, this will save costs and time
Implement the re-design, this will displease the executive team due to the delay in spite of performance benefits
21: It is now time to roll out the plane design to small-batch manufacturers, so commercial readiness testing can begin. As a group, it is time to decide what kind of manufacturer the company will align with.
Choose a cheaper, less established, manufacturer, this will cut costs substantially, but this is not a manufacturer used before by the company
Remain with the same manufacturer as before, minimal risk, and minor loyalty discounts
Choose a premium manufacturer, quite risky since again not used before, but an extremely good track record and has shown initiative with manufacturing, also more expensive
22: Your manufacturer has identified a way to cut costs substantially in manufacturing, with a small performance loss to the aerodynamic performance of the plane, do you decide to do it?
Yes, costs are cut, but reputation of the company will take a hit
No, costs will remain the same, but the company's commitment to quality is noted
23: Based on this information, what will the team do prior to the second wave of manufacturing?
A thorough investigation into the 2% issues, fixes to issues identified here will substantially delay the project and incur big costs, since it involves complete re-design, but will fix any issues no matter how significant or otherwise
A brief investigation into the 2% issues, this will consider the issues found here, but will delay the project quite a bit less, since there is no re-design, and should still fix most issues prevalent in the plane
Do not pursue any fixes, if the issues are trivial, which experts consider to be likely, this will save substantial time and costs
24: How should the company approach the upcoming inspection?
Skip the inspection, this will save two weeks worth of time and costs and thus will be very beneficial to the company
Perform a shorter one-week inspection, this will cover the 'big picture' of the plane design, but won't go into the same depth as a two-week inspection
Perform the two-week inspection, this is a standard-issue inspection that will consider details, but won't go into thorough cross-testing
Perform a six-week inspection, this is a comprehensive inspection that will guarantee the quality of the plane is up to standard, and also consider business but it will delay the project by a huge amount
25: To what extent should the company leverage it's networks to achieve a competitive advantage?
Utilize to its full extent, this method involves some bribery and under-the-table dealings to develop a competitive advantage over rivals, but will greatly affect the company's reputation since word will get out
Utilize to a legal extent, this is an aggressive method that is similar to A, but does not enter illegal territory, but because of this will be far less effective, yet still useful
Utilize to a reasonable extent, this is a standard method where competitors and the public will see as 'fair', still incurs costs, but can be eclipsed by competitors methods if they decide to be aggressiv
Do not utilize, this will save two weeks of costs, and will be great for reputation, but will hinder the competitive effort


26: Will the company employ these tactics?
Yes, this will incur costs, but will almost certainly lead to a competitive victory
No, this will upset the executive board and reduce competitive advantage, but will save some costs
IF (16D|23A|24D)
Yes
No
30: Continue to the final outcome!
27: How will the team react?
Ok!
Commit to a full investigation into the issue, this is considered by most experts to be a waste since the pilots appear to be the cause of the crash, this will incur substantial, possibly crippling, costs, but will please the FAA
Perform a standard investigation into the issue, this is expected by the FAA, and will help identify any fundamental issue with the plane, without leading to large cost overheads
Perform no investigation, relying on the comprehensive investigations performed by others, this leads to no costs, and operations will continue, in fact, this will prevent loss of contracts as clients will be more confident

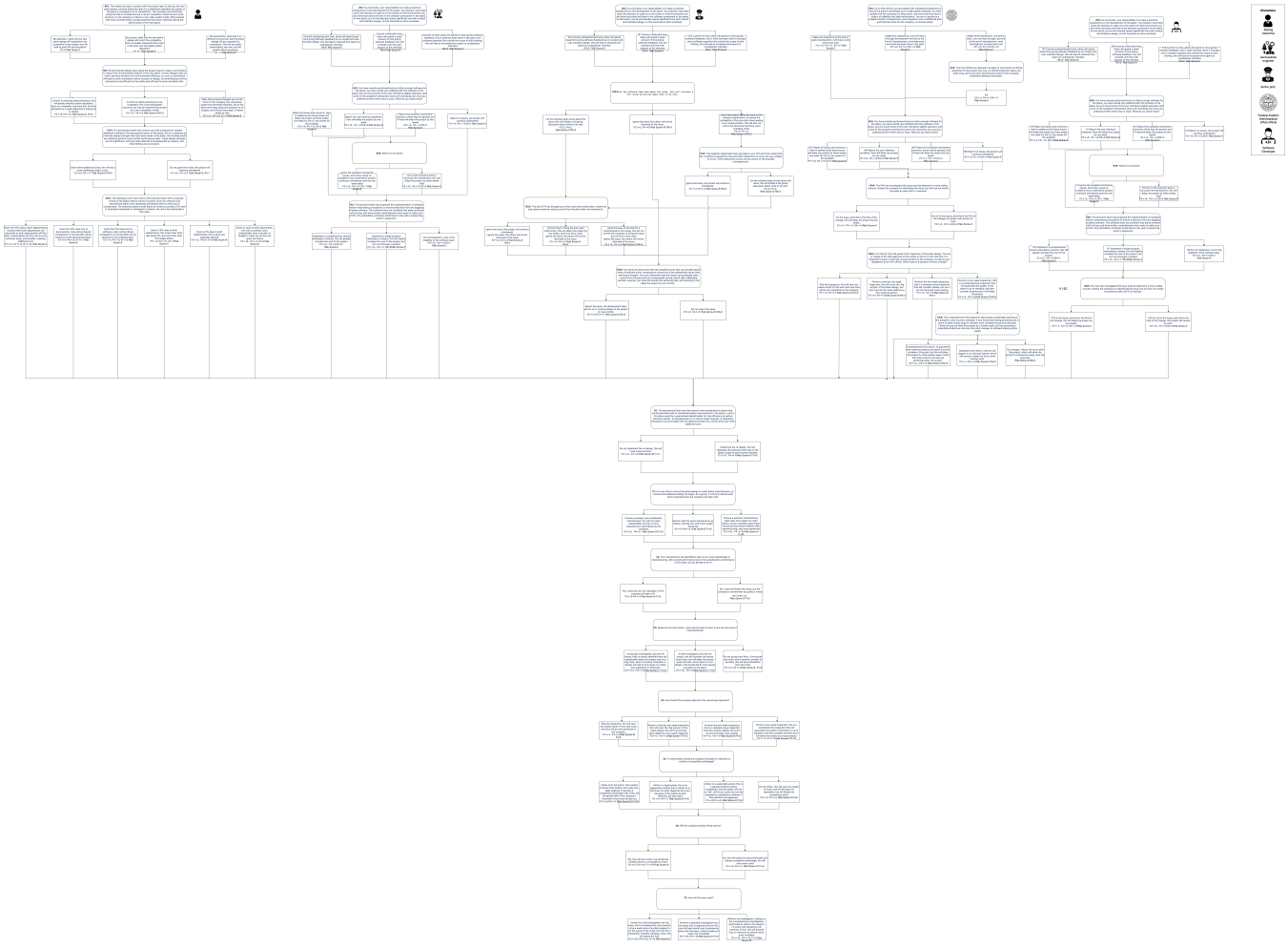
Decision Tree (version 2)

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



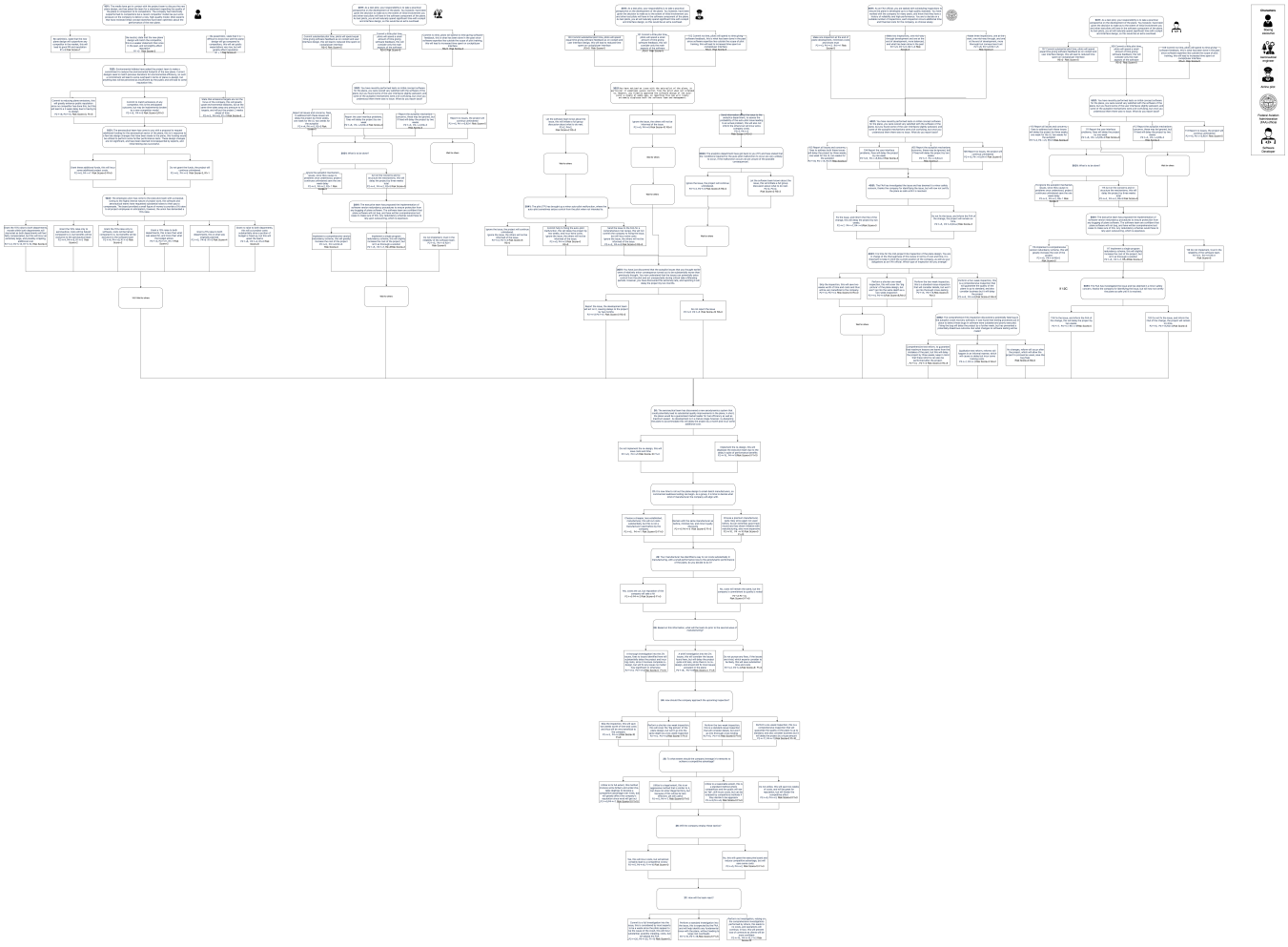
Decision Tree (version 3)

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.

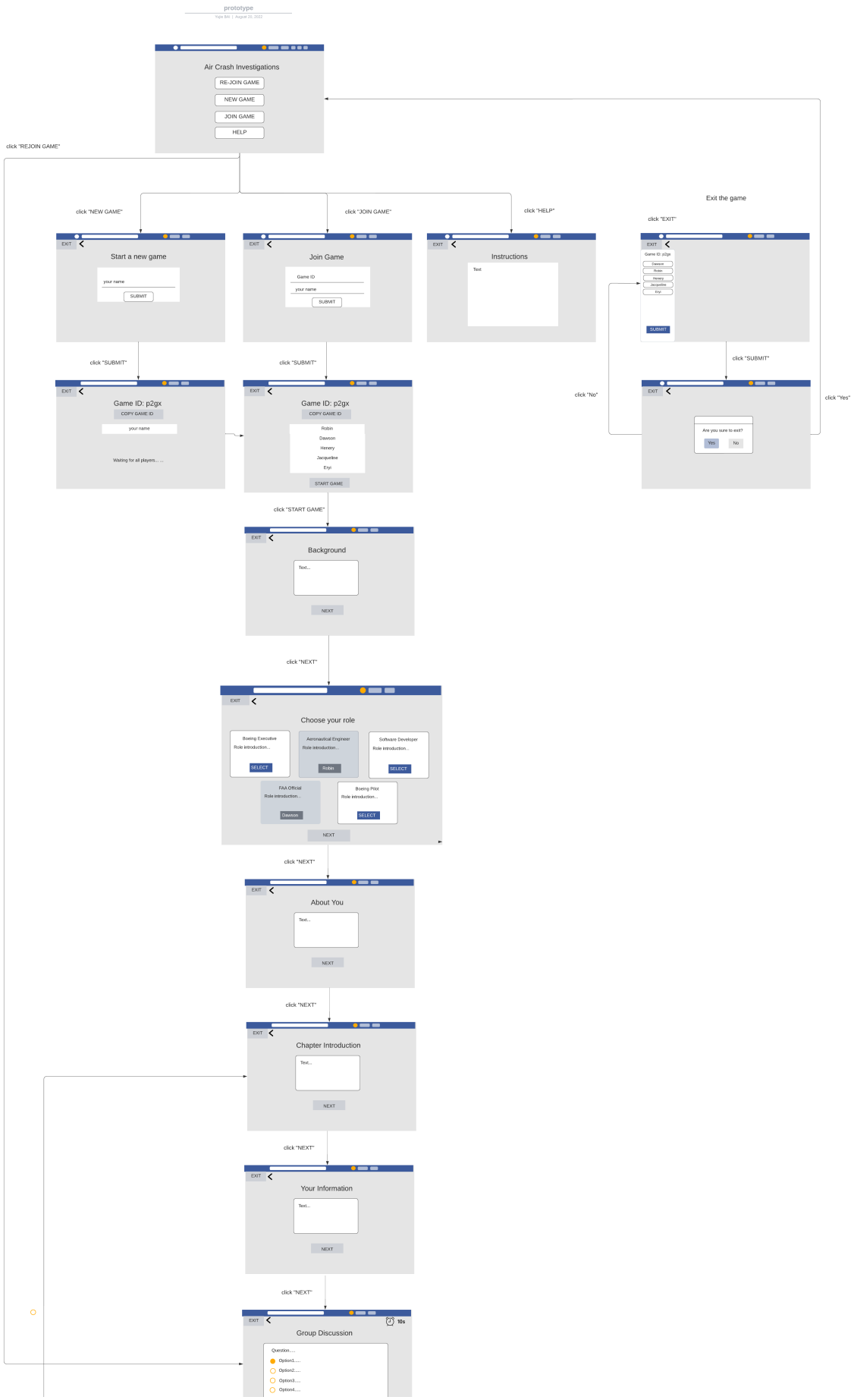
To learn more about the subsections, please follow the links below:

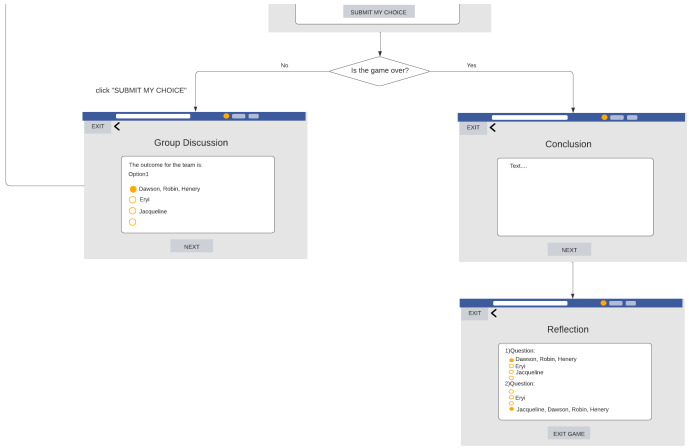
- Paper prototype
- Digital prototype

Digital prototype

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

Paper prototype


prototype
Yujie Bai | August 20, 2022
Air Crash Investigations
RE-JOIN GAME
NEW GAME
JOIN GAME
HELP
click "REJOIN GAME"
click "NEW GAME"
click "JOIN GAME"
click "HELP"
Exit the game
click "EXIT"
EXIT
Start a new game
your name
SUBMIT
Join Game
Game ID
your name
SUBMIT
Instructions
Text
Game ID: p2gx
Dawson
Robin
Henery
Jacqueline
Eryi
SUBMIT
click "SUBMIT"
click "SUBMIT"
click "SUBMIT"
Game ID: p2gx
COPY GAME ID
your name
Waiting for all players... ...
Game ID: p2gx
COPY GAME ID
Robin
Dawson
Henery
Jacqueline
Eryi
START GAME
Are you sure to exit?
Yes
No
click "No"
click "Yes"
click "START GAME"
Background
Text...
NEXT
click "NEXT"
Choose your role
Boeing Executive
Role introduction...
SELECT
Aeronautical Engineer
Role introduction...
Robin
Software Developer
Role introduction...
SELECT
FAA Official
Role introduction...
Dawson
Boeing Pilot
Role introduction...
SELECT
NEXT
click "NEXT"
About You
Text...
NEXT
click "NEXT"
Chapter Introduction
Text...
NEXT
click "NEXT"
Your Information
Text...
NEXT
click "NEXT"
10s
Group Discussion
Question....
Option1....
Option2....
Option3....
Option4....


SUBMIT MY CHOICE
Is the game over?
No
Yes
click "SUBMIT MY CHOICE"
EXIT
Group Discussion
The outcome for the team is:
Option1
Dawson, Robin, Henery
Eryl
Jacqueline
NEXT
EXIT
Conclusion
Text....
NEXT
EXIT
Reflection
1)Question:
Dawson, Robin, Henery
Eryl
Jacqueline
2)Question:
Eryl
Jacqueline, Dawson, Robin, Henery
EXIT GAME

Architecture

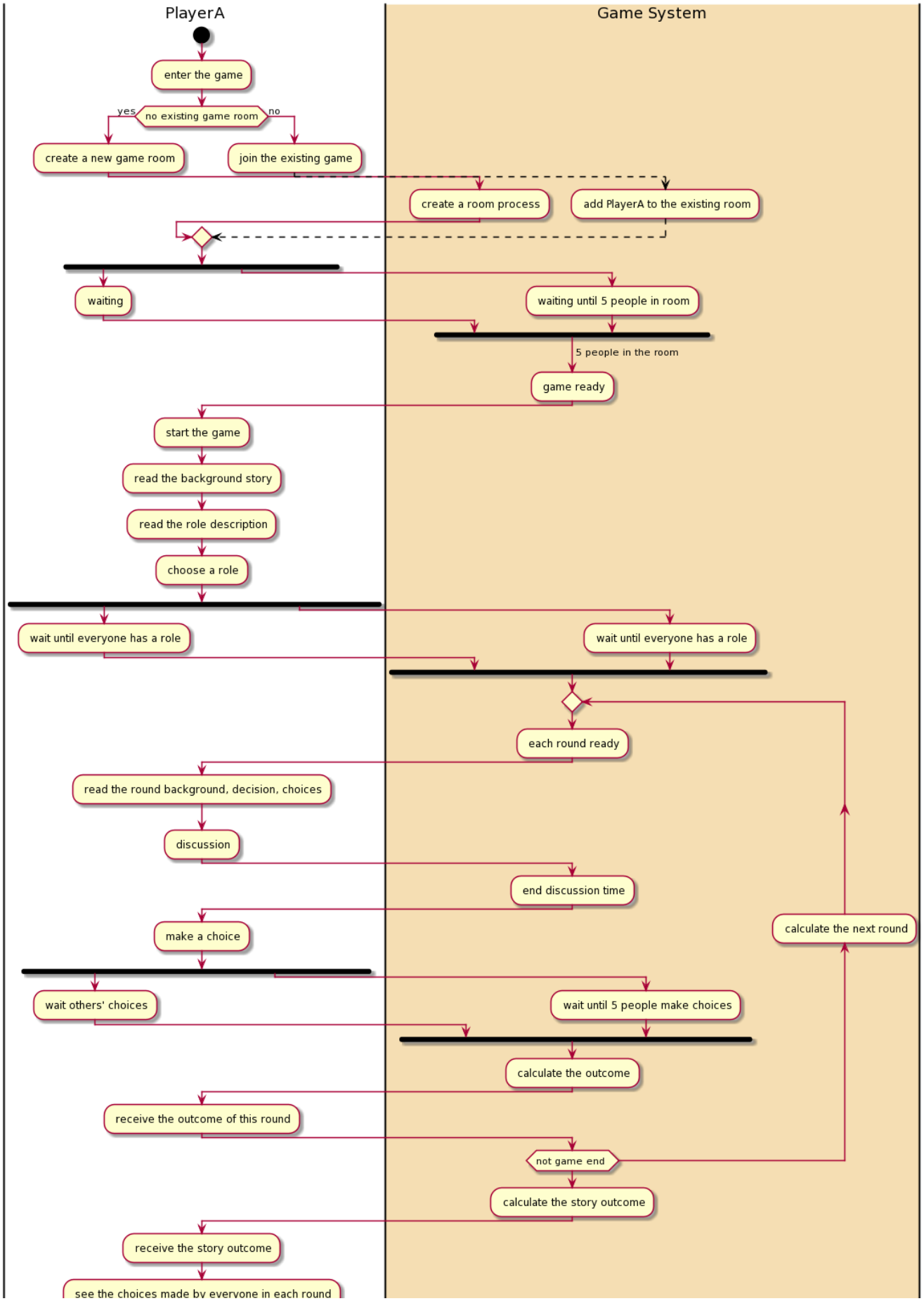
Activity Diagram

Description:

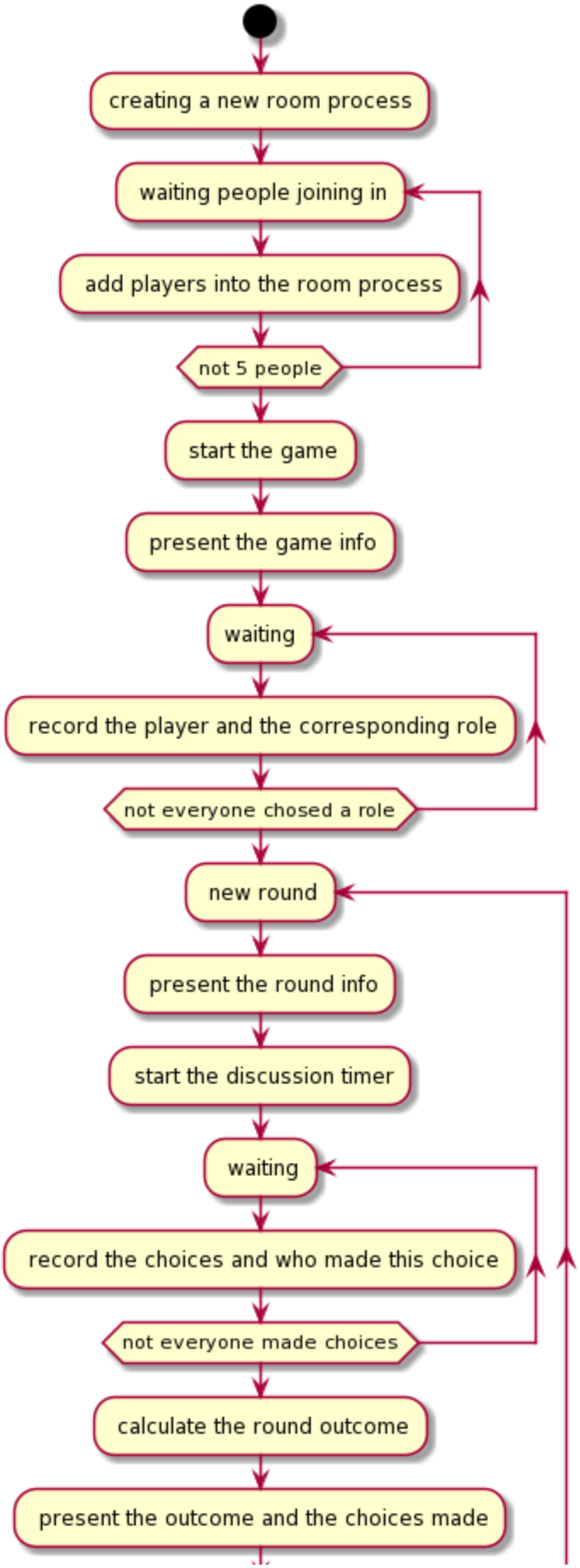
The activity diagram is used to describe the activities that will be performed by the user (player) and the system (the game). It also describe the functionality that the system provides to the users.

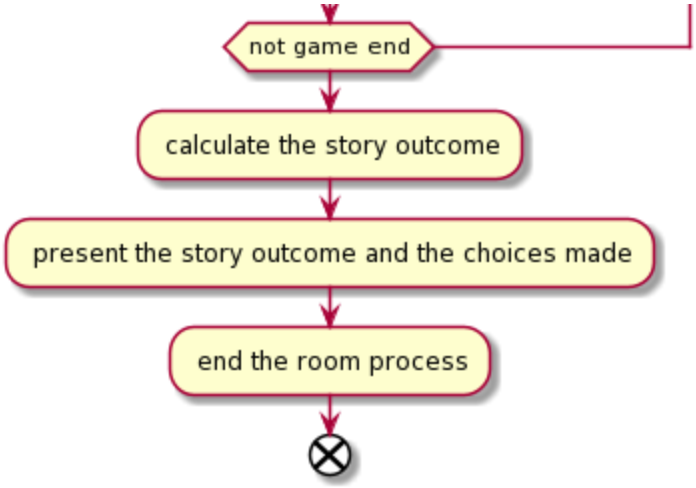
The activity is an abstraction of the behaviour excluding the details of interations. For the interation details between the frondend, the backend server and the database, please refer to Sequence Diagram.

Player and Game System Activity Diagram


PlayerA
Game System
enter the game
no existing game room
yes
no
create a new game room
join the existing game
create a room process
add PlayerA to the existing room
waiting
waiting until 5 people in room
5 people in the room
game ready
start the game
read the background story
read the role description
choose a role
wait until everyone has a role
wait until everyone has a role
each round ready
read the round background, decision, choices
discussion
end discussion time
make a choice
calculate the next round
wait others' choices
wait until 5 people make choices
calculate the outcome
receive the outcome of this round
not game end
calculate the story outcome
receive the story outcome
see the choices made by everyone in each round

Game System Activity Diagram


creating a new room process
waiting people joining in
add players into the room process
not 5 people
start the game
present the game info
waiting
record the player and the corresponding role
not everyone chosed a role
new round
present the round info
start the discussion timer
waiting
record the choices and who made this choice
not everyone made choices
calculate the round outcome
present the outcome and the choices made


not game end
calculate the story outcome
present the story outcome and the choices made
end the room process

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	• body • `{"name": "Johon", "duration": 30}` • Note: if user does **not** select that they want a timed game, duration = 0	• 200 • 400

2	✅	✅	POST	/play/{gameID}/join	A user wants to join an existing game	• gameID
						  • (path, string)
						• body
						  • `{"name": "Johon"}`
							• 200
							• 400
							  • e.g. invalid Game ID supplied

3	✅	✅	GET	/play/{gameID}/wait	Player is waiting for all players to join the game	• gameID
						  • (path, string)
							• 200
							  • `{"players": [ {"id": 0, "name": "John"}, {"id": 1, "name": "Georgia"}, {"id": 2, "name": "Emily"} ] "ready": false}`
							  • ready = true if all 5 players are in and the game is ready to start
							• 400
							  • e.g. invalid Game ID

4	✅	✅	GET	/play/{gameID}/status	Retrieve information about the current game e.g. when viewing the in-game menu	• gameID
						  • (path, string)
							• 200
							  • <pre>{"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"},
							]
							}</pre>
							  • subject to change
							  • status of the game can be:
							  • waiting - waiting for all players to join the game
							  • ready - all 5 players have joined, ready for players to press "Start game"
							  • inProgress
							  • completed
							• 400
							  • e.g. invalid Game ID

5	✅	✅	GET	/play/{gameID}/background	Get the "Background" information shown to all players	• gameID
						  • (path, string)
							• 200
							  • the content (perhaps as plain text including <p> tags? or maybe as JSON with .)
							• 400

6	✅	✅	GET	/play/{gameID}/roles	Get the "role" information shown to all players	• gameID
						  • (path, string)
							• 200
							  • <pre>{"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
							]
							}</pre>

7	✅	✅	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)
							• 200
							  • <pre>{"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"}
							]
							}</pre>
							• 400

8	✅	✅	POST	/play/{gameID}/choose-role	Player selects their role	• gameID
						  • (path, string)
						• roleID
						  • role ID (int between 0 and 4, as specified in the previous request)
							• 200
							  • player successfully assigned to role
							• 400
							  • e.g. invalid game ID
							• 403
							  • player is not assigned to the role e.g. the role is no longer available

9	✅
	update
		✅
		now have a bug
			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	• body
						  • `{"roleID":0}`
							• 200
							  • the content
							• 400

10		✅	GET	/play/make-decision/:gameID	Get all the decisions during the game in reflection phase.	• gameID
						  • (path, string)
							• 200
							  • {"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
							},
							{"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
							},
							]
							}
							• 400
							  • e.g. invalid game ID

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"	• gameID
						  • (path, string)
						• body
						  • `{"roleID": 1}`
							• 200
							  • `{"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"}] }`
							• 400
							  • e.g. invalid game ID

12			POST	/play/make-decision/:gameID	Player enters their selected choice for the question of the round	• gameID
						  • (path, string)
						• body
						  • `{"questionID": 1, "optionID": 2, "roleID": 1}`
							• 200 - choice was valid and recorded correctly
							  • `{"questionID": 1, "optionID": 2, "roleID": 1}`
							• 400
							  • e.g. invalid input, round has expired, etc.

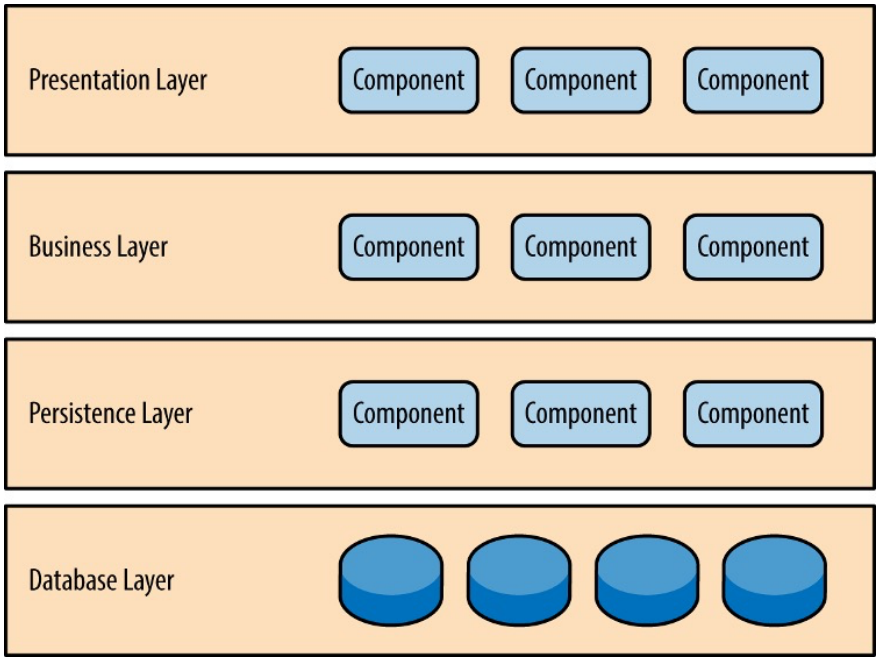
13	✅		GET	/play/{gameID}/round-status	Get information about the current round.
					
					The purpose of this is so the front-end can frequently check the state of the round, and know when to move on to the outcome.
						• gameID
						  • (path, string)
							• 200
							  • `{"questionID": 1, "status": "pre" "voted": 2 "remaining": 3 }`
							
							  • roundID is the current round in play for this game
							  • perhaps this should be 0 if we are not actually in a round (e.g. we are still in the role selection phase).
							  • status of the **round** can be:
							  • **pre** - pre-decision. round is in progress, timer is running at back-end, players should be reading the information /discussing/entering their choice
							  • **post** - 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.
							  • voted: number of players that have entered their decision
							  • remaining: number of players who still need to enter their decision
							  • 400
							  • e.g. invalid game ID

14	✅		GET	/play/{gameID}/round-outcome	Get the outcome and conclusion for the round
					
					i.e. "Result" page in the prototype
						• gameID
						  • (path, string)
							• 200
							  • `{"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."] }`
							
							• 400
							  • e.g. invalid game ID

15			GET	/play/{gameID}/game-outcome	Get the final game outcome of game i.e. "Outcome" page in the prototype	• gameID
						  • (path, string)
							• 200
							  • The game outcome
							
							• 400
							  • e.g. invalid game ID

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:

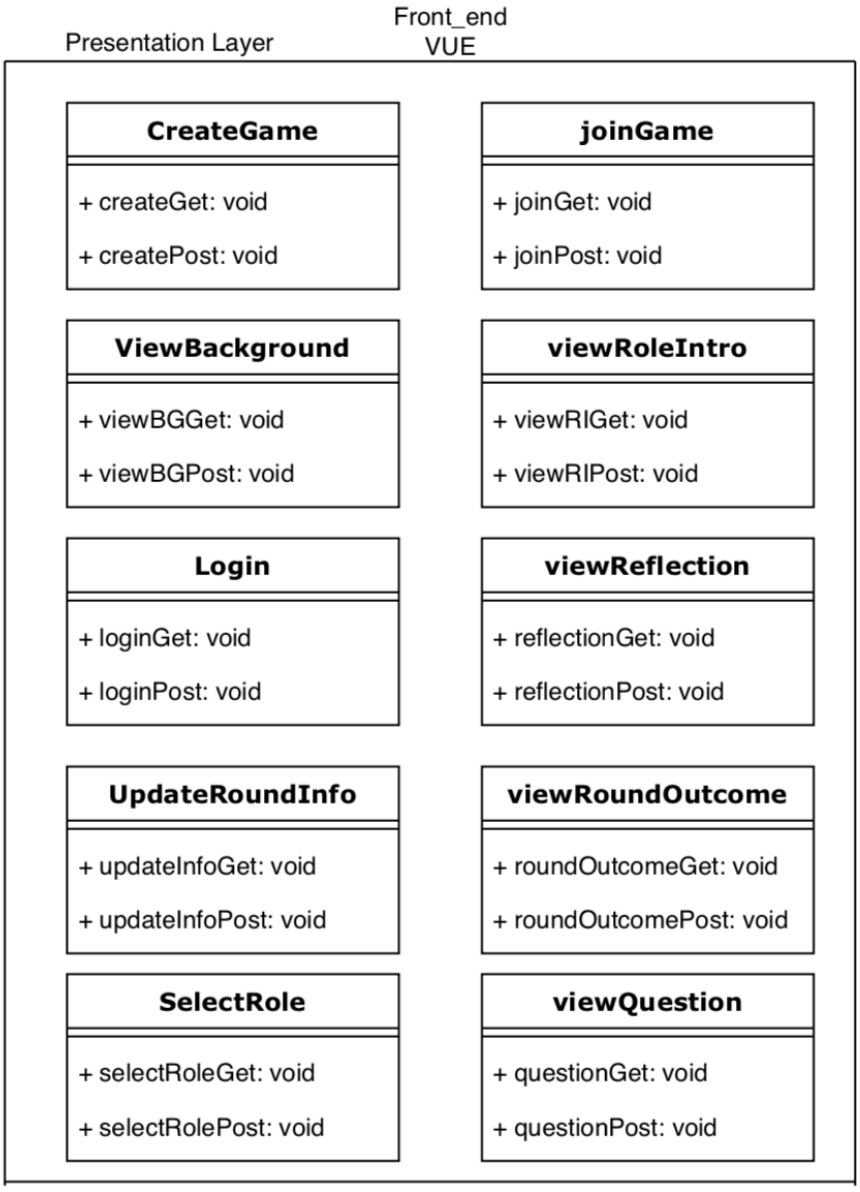
- Story (-> n round -> 5 Role): background(string), outcome
- Round(-> 1 question): roundID, roundBackground, roundOutcome, method , nextRoundID
- Role: roleID(int), roleName (string), roleBackground
- RoleQuestion: roleID, questionID, weight, individualInfo, answer
- Question (-> n Decision): questionID(int), 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:

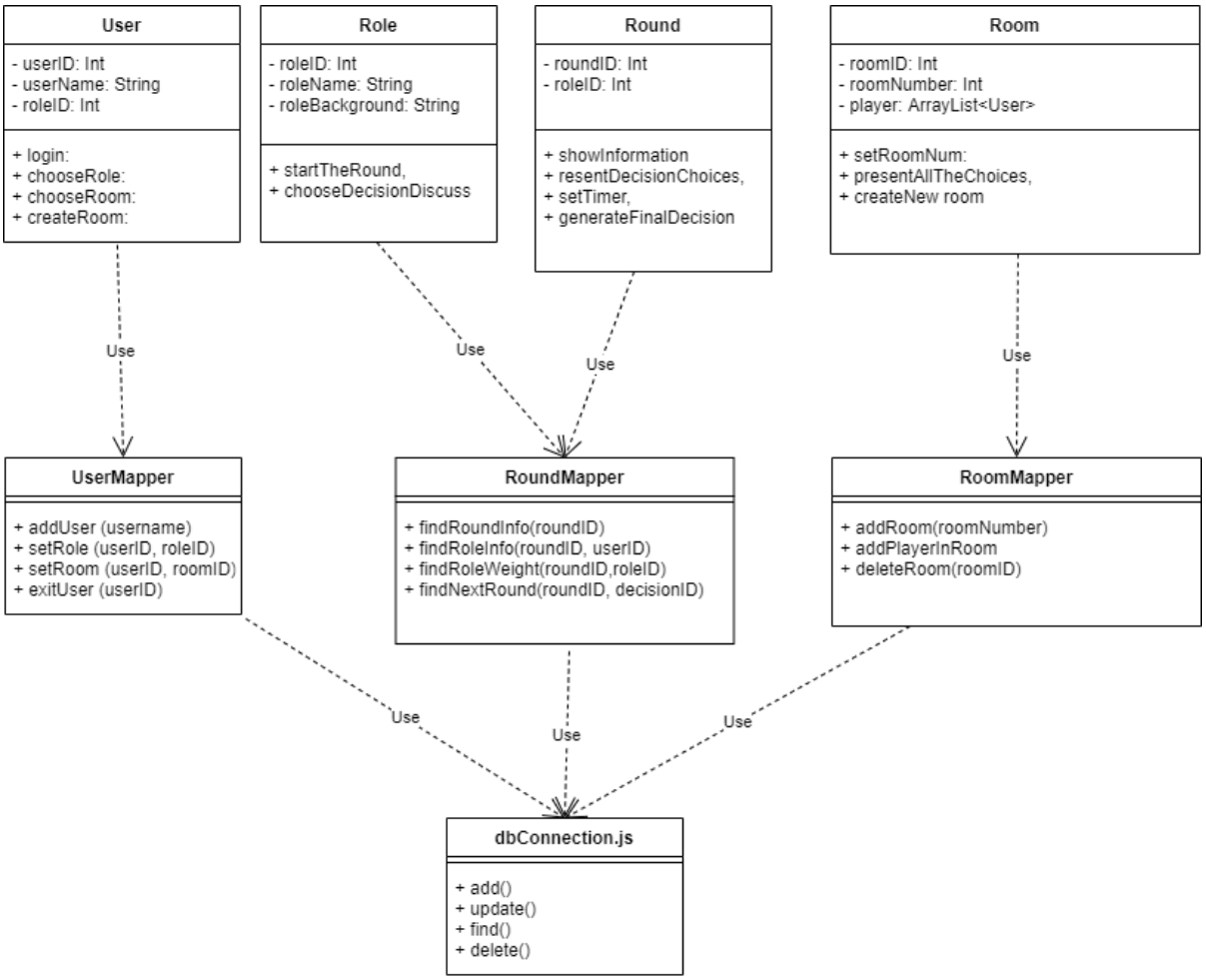
Data source layer: SQL

Class Diagram

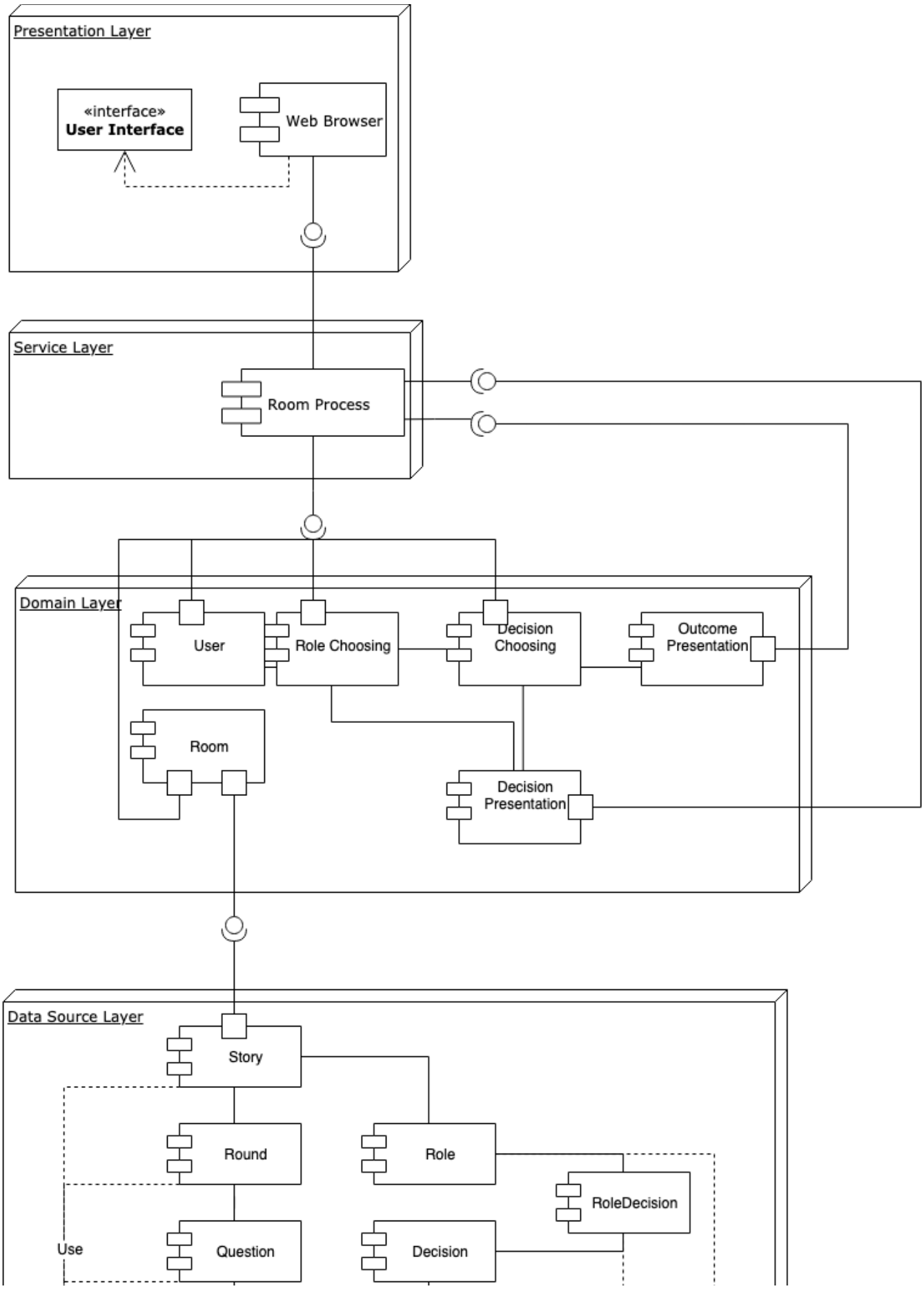
Class DiagramFrontend

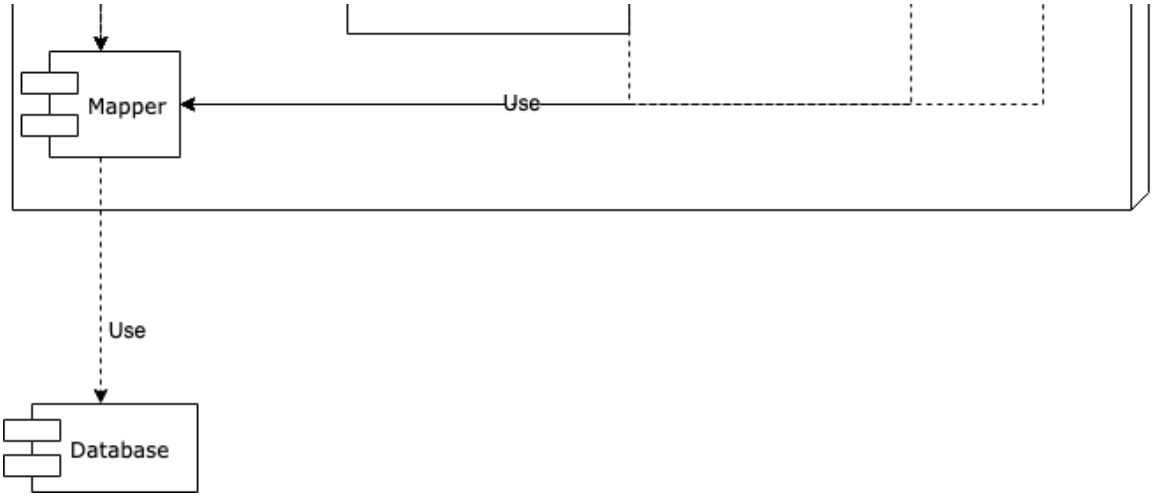


Class DiagramBackend


User
- userID: Int
- userName: String
- roleID: Int
+ login:
+ chooseRole:
+ chooseRoom:
+ createRoom:
Role
- roleID: Int
- roleName: String
- roleBackground: String
+ startTheRound,
+ chooseDecisionDiscuss
Round
- roundID: Int
- roleID: Int
+ showInformation
+ resentDecisionChoices,
+ setTimer,
+ generateFinalDecision
Room
- roomID: Int
- roomNumber: Int
- player: ArrayList<User>
+ setRoomNum:
+ presentAllTheChoices,
+ createNew room
Use
Use
Use
Use
UserMapper
+ addUser (username)
+ setRole (userID, roleID)
+ setRoom (userID, roomID)
+ exitUser (userID)
RoundMapper
+ findRoundInfo(roundID)
+ findRoleInfo(roundID, userID)
+ findRoleWeight(roundID,roleID)
+ findNextRound(roundID, decisionID)
RoomMapper
+ addRoom(roomNumber)
+ addPlayerInRoom
+ deleteRoom(roomID)
Use
Use
Use
dbConnection.js
+ add()
+ update()
+ find()
+ delete()

Component Diagram

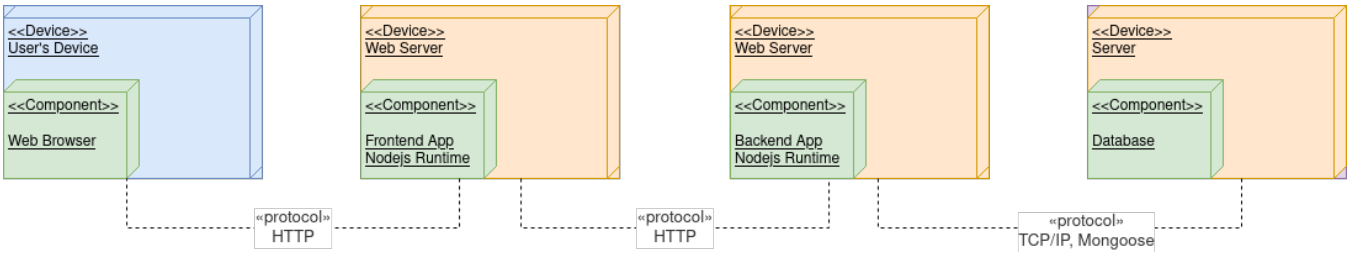

Presentation Layer
«interface»
User Interface
Web Browser
Service Layer
Room Process
Domain Layer
User
Role Choosing
Decision Choosing
Outcome Presentation
Room
Decision Presentation
Data Source Layer
Story
Round
Role
RoleDecision
Use
Question
Decision


Mapper
Use

Use

Database

Deployment Diagram



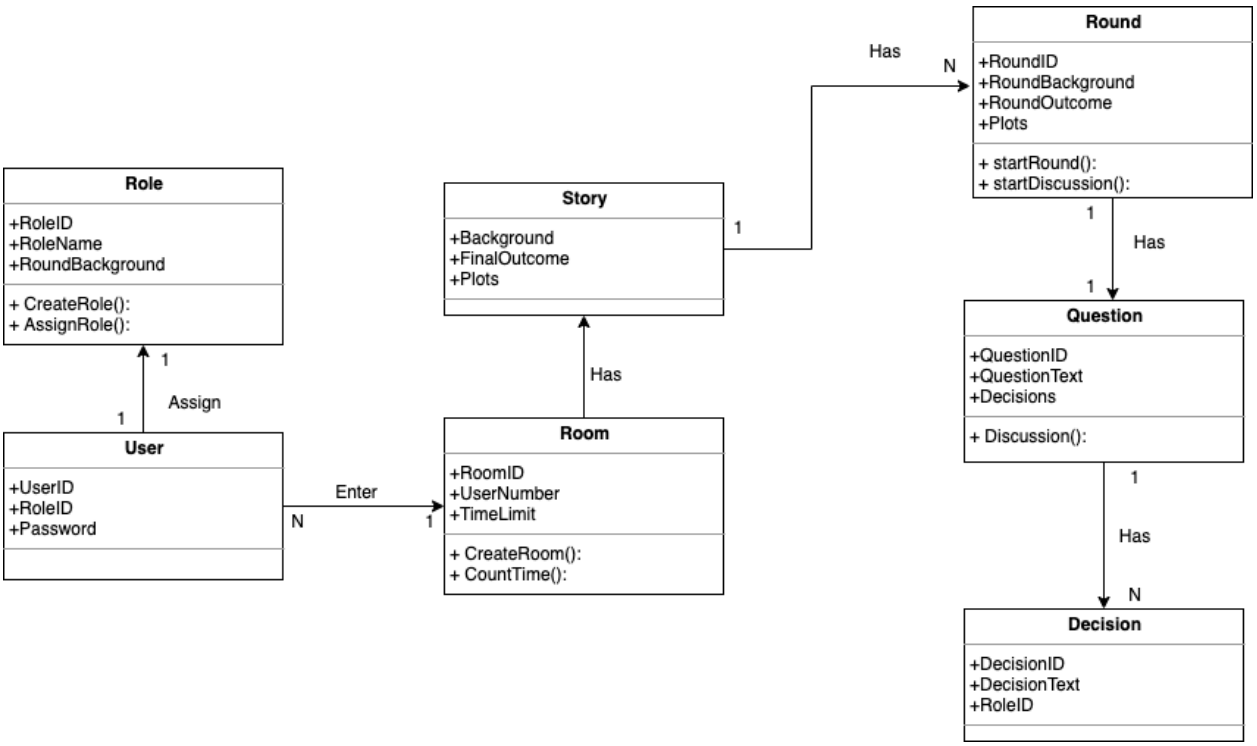
Notes

- 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:%221099996307737903808 18%22%7D#G1B2Dn9RzMut6Gwm_7I-nIBh1hFpn8ghu0



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

1. 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
2. The player/s make the decision, based on the 'consequence' object for that decision, the 'company' object is updated, the decision history is updated
3. 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.

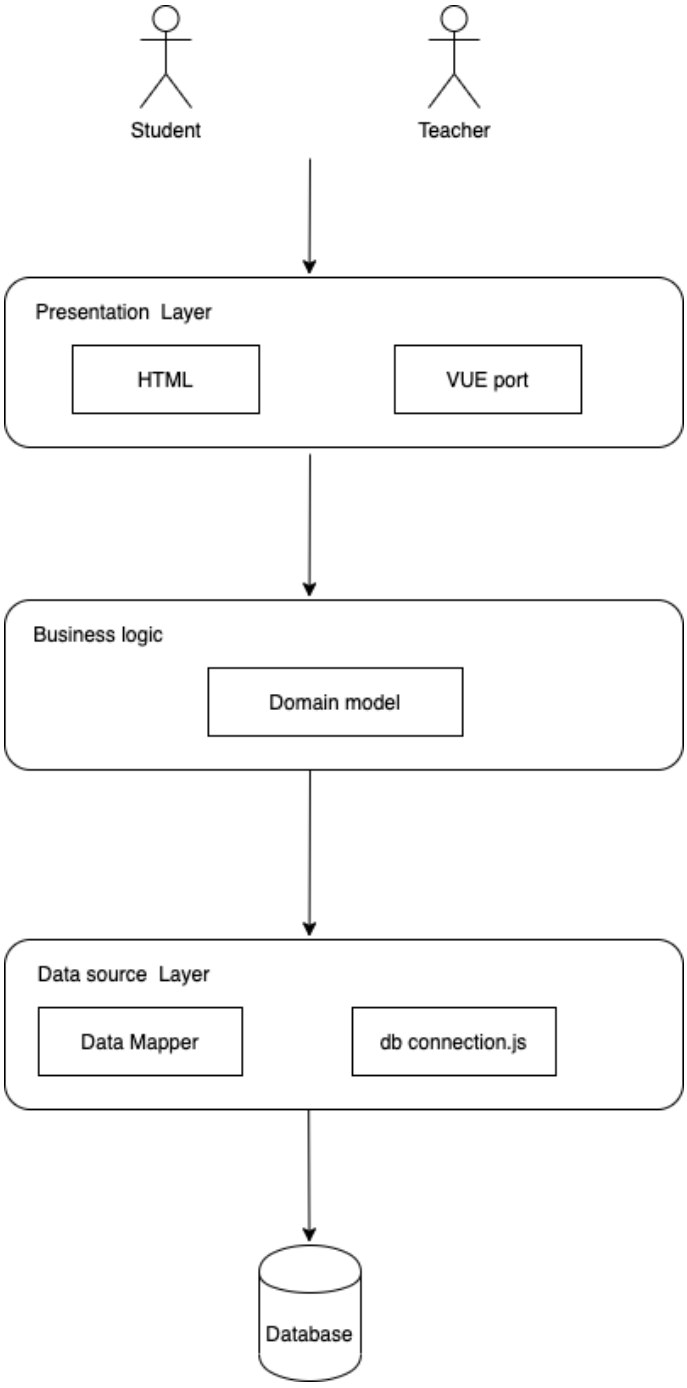
Advantages

- 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



Sequence Diagram

Tool: https://sequencediagram.org/

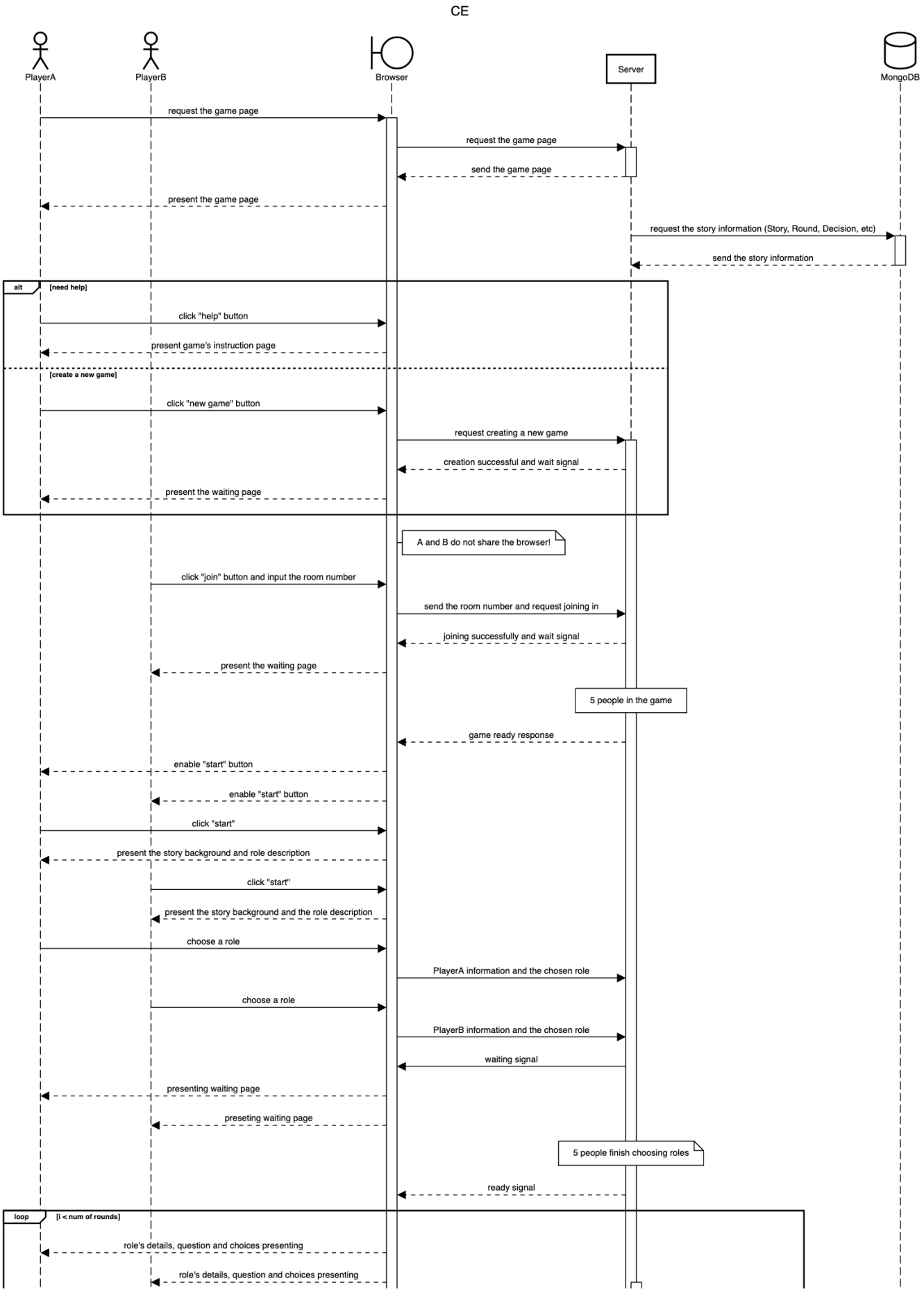
Link of the current diagram and code: CE Sequence Diagram

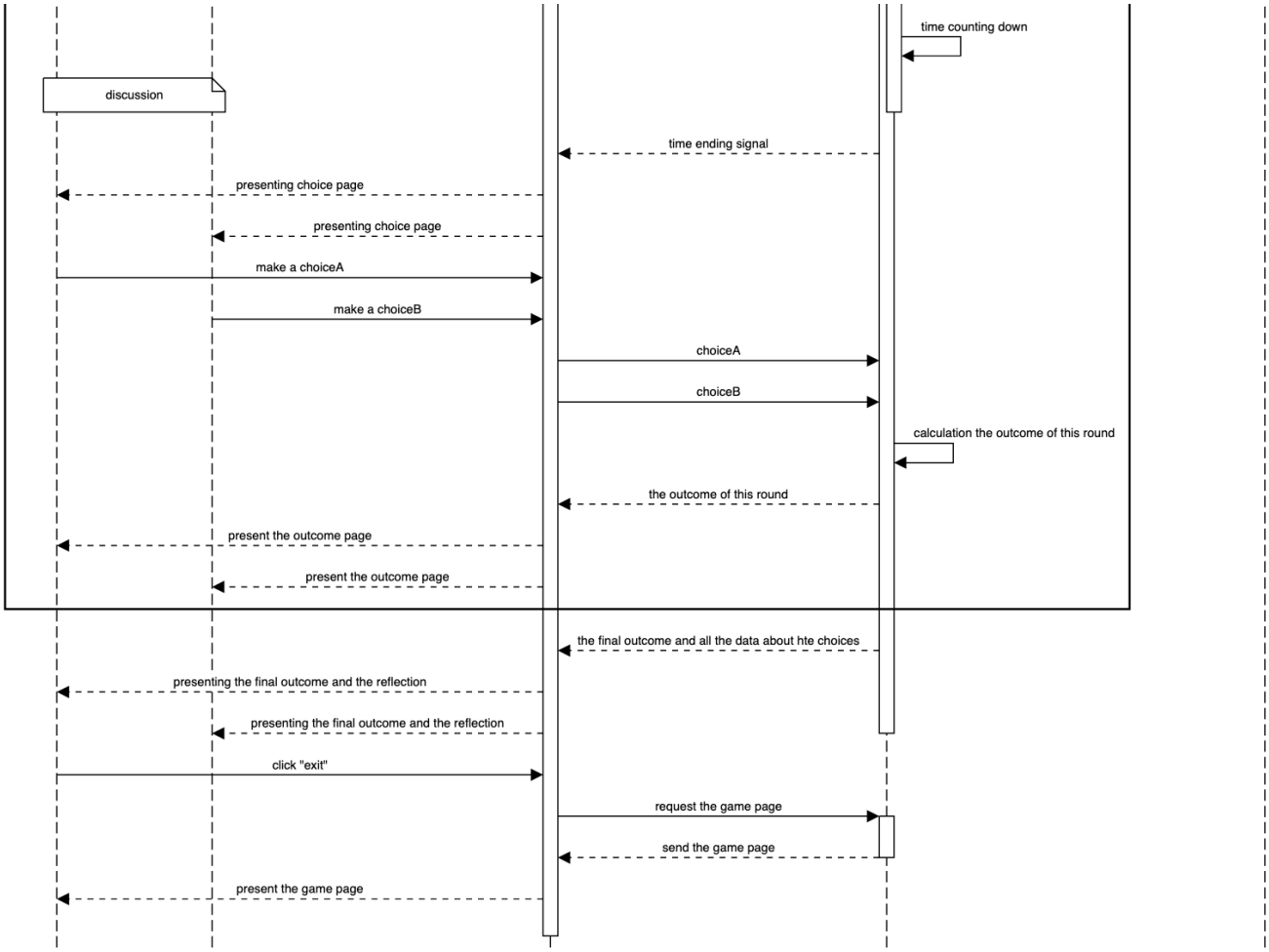
Note:

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 assuption 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.


CE
PlayerA
PlayerB
Browser
Server
MongoDB
request the game page
request the game page
send the game page
present the game page
request the story information (Story, Round, Decision, etc)
send the story information
alt
[need help]
click "help" button
present game's instruction page
[create a new game]
click "new game" button
request creating a new game
creation successful and wait signal
present the waiting page
A and B do not share the browser!
click "join" button and input the room number
send the room number and request joining in
joining successfully and wait signal
present the waiting page
5 people in the game
game ready response
enable "start" button
enable "start" button
click "start"
present the story background and role description
click "start"
present the story background and the role description
choose a role
PlayerA information and the chosen role
choose a role
PlayerB information and the chosen role
waiting signal
presenting waiting page
preseting waiting page
5 people finish choosing roles
ready signal
loop
[i < num of rounds]
role's details, question and choices presenting
role's details, question and choices presenting


time counting down
discussion
time ending signal
presenting choice page
presenting choice page
make a choiceA
make a choiceB
choiceA
choiceB
calculation the outcome of this round
the outcome of this round
present the outcome page
present the outcome page
the final outcome and all the data about hte choices
presenting the final outcome and the reflection
presenting the final outcome and the reflection
click "exit"
request the game page
send the game page
present the game page

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.

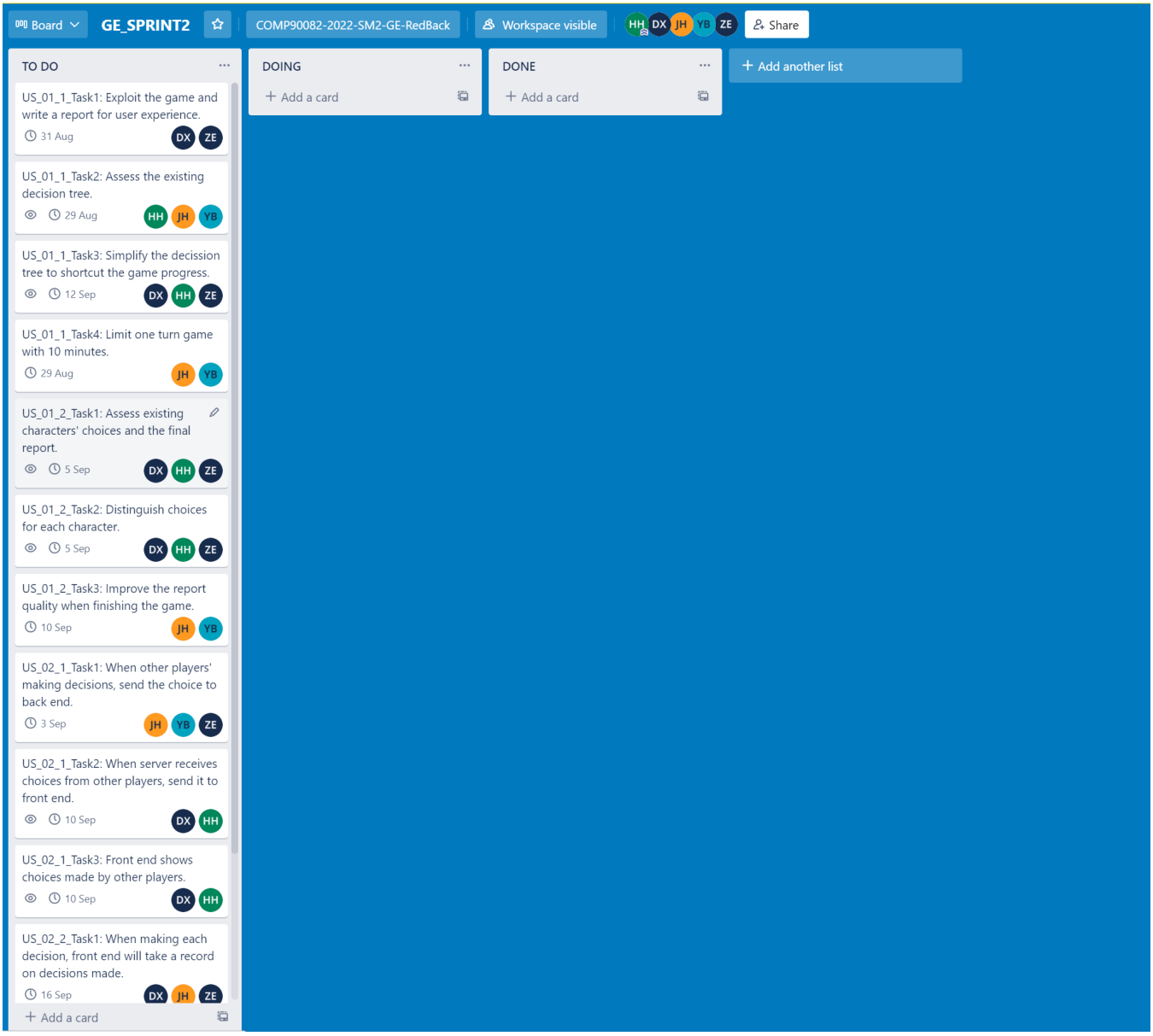
This included initialising and deploying the frameworks used by the game, debug it to make sure it will run successfully on local device, then realize new features.

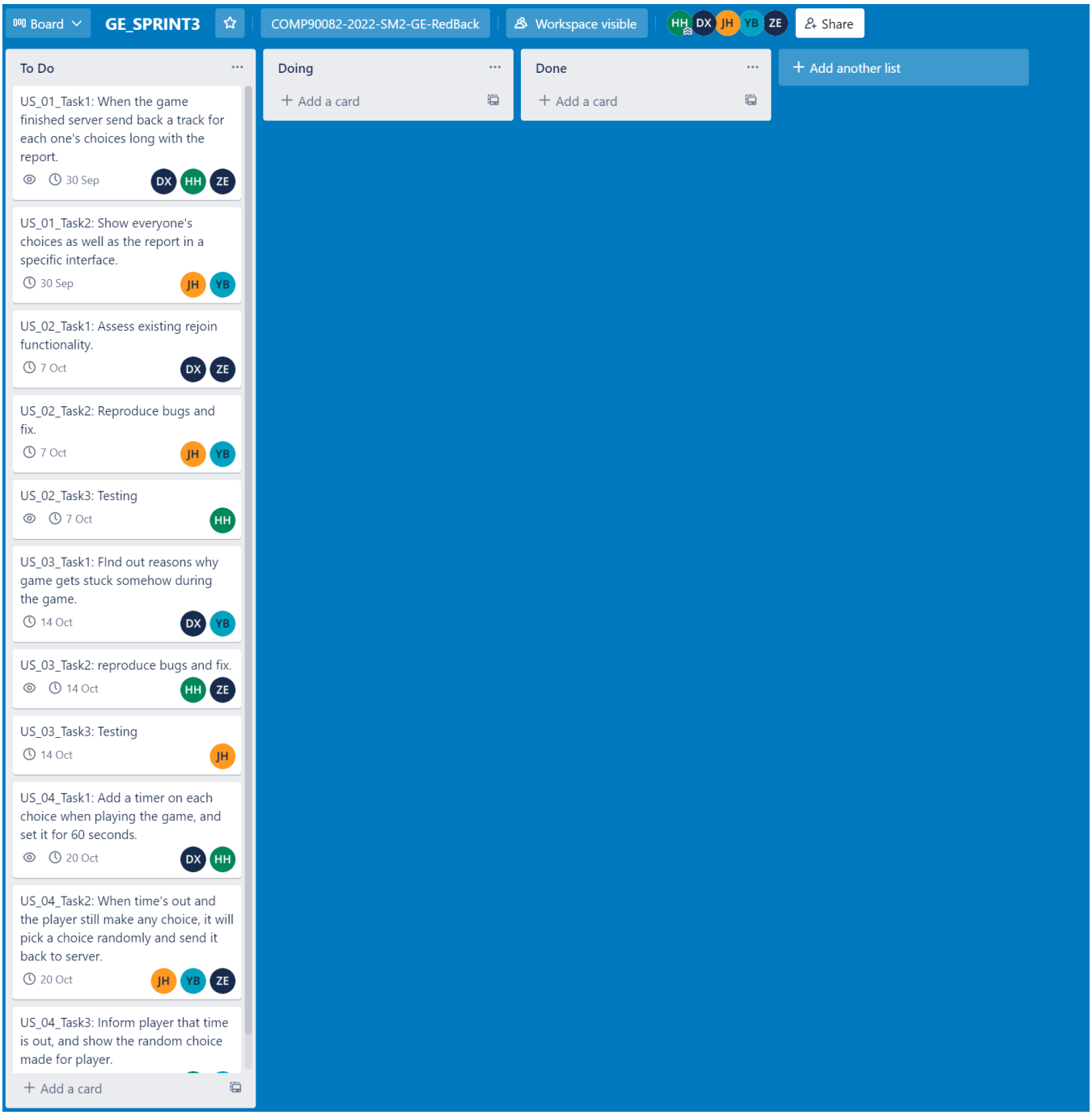
User stories

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 decission 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




Board
GE_SPRINT3
COMP90082-2022-SM2-GE-RedBack
Workspace visible
HH DX JH YB ZE
Share
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
DX HH ZE
US_01_Task2: Show everyone's choices as well as the report in a specific interface.
30 Sep
JH YB
US_02_Task1: Assess existing rejoin functionality.
7 Oct
DX ZE
US_02_Task2: Reproduce bugs and fix.
7 Oct
JH YB
US_02_Task3: Testing
7 Oct
HH
US_03_Task1: FInd out reasons why game gets stuck somehow during the game.
14 Oct
DX YB
US_03_Task2: reproduce bugs and fix.
14 Oct
HH ZE
US_03_Task3: Testing
14 Oct
JH
US_04_Task1: Add a timer on each choice when playing the game, and set it for 60 seconds.
20 Oct
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.
20 Oct
JH YB ZE
US_04_Task3: Inform player that time is out, and show the random choice made for player.
Add a card
Doing
Add a card
Done
Add a card
Add another list

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.

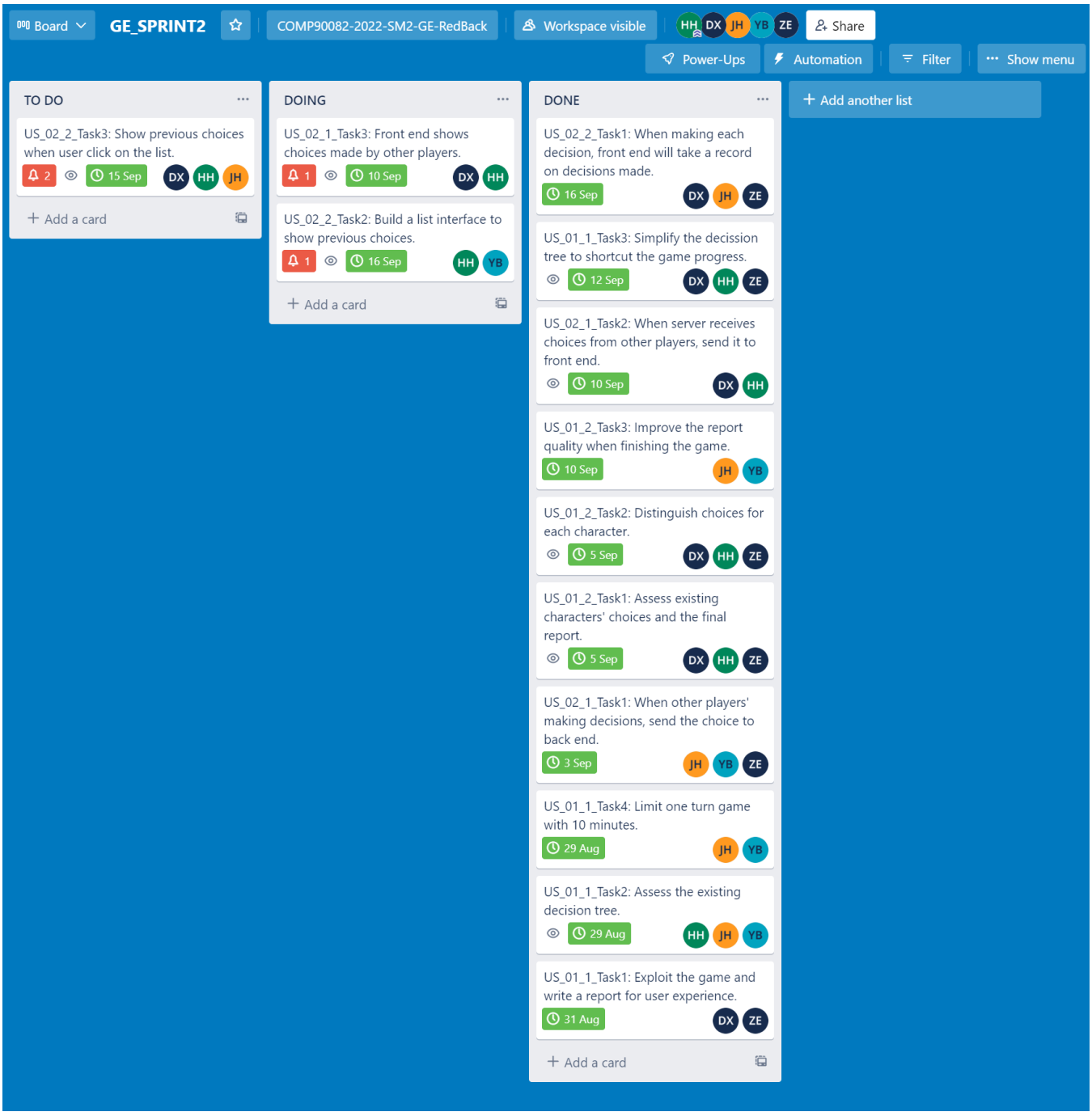
The main problem for this game that prevent it from running successfully is mainly caused by logical bugs. Our team got stuck for a long time during testing and reproducing because the logs didn't show enough information for us to locate the problems. We aim to polish the logs in the future to make debug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

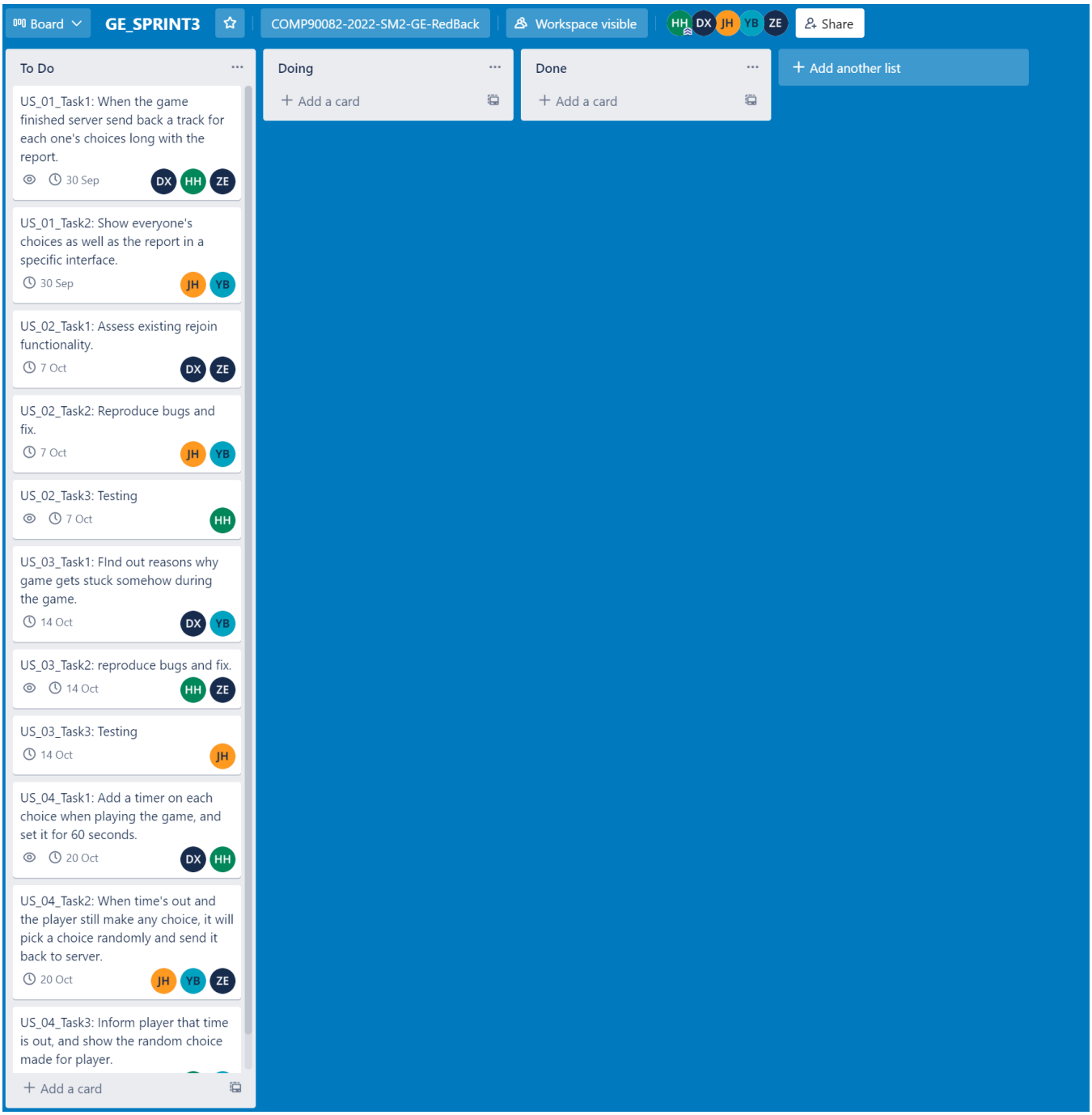
Sprint	User Story	Decomposed Case	Tasks	Point	Priority	Achieved
2	US_01	Simplify the game process	1. Exploit the game and write a report for user experience.	5	Must Have	The new structure of decision tree is designed in Sprint 2. But the implementation will be done in Sprint 3.
			2. Assess the existing decision tree.	9		
			3. Simplify the decission 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		

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

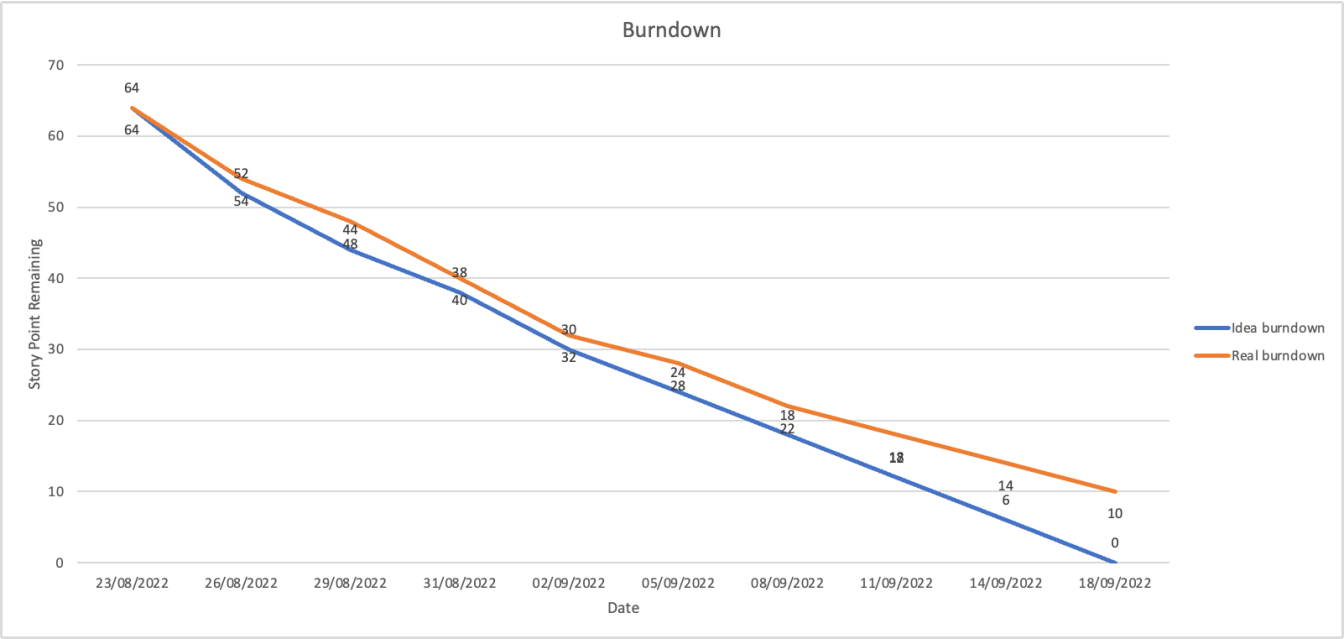
Sprint	User Story	Decomposed Case	Tasks	Point	Priority
3	US_03	To get a overall report that shows decisions made by other users for each questions throughout the game	1. When the game finished server send back a track for each one's choices long with the report.	8	Must Have
			2. Show everyone's choices 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	


Board
GE_SPRINT2
COMP90082-2022-SM2-GE-RedBack
Workspace visible
HH DX JH YB ZE
Share
Power-Ups
Automation
Filter
Show menu
TO DO
US_02_2_Task3: Show previous choices when user click on the list.
2
15 Sep
DX HH JH
Add a card
DOING
US_02_1_Task3: Front end shows choices made by other players.
1
10 Sep
DX HH
US_02_2_Task2: Build a list interface to show previous choices.
1
16 Sep
HH YB
Add a card
DONE
US_02_2_Task1: When making each decision, front end will take a record on decisions made.
16 Sep
DX JH ZE
US_01_1_Task3: Simplify the decission tree to shortcut the game progress.
12 Sep
DX HH ZE
US_02_1_Task2: When server receives choices from other players, send it to front end.
10 Sep
DX HH
US_01_2_Task3: Improve the report quality when finishing the game.
10 Sep
JH YB
US_01_2_Task2: Distinguish choices for each character.
5 Sep
DX HH ZE
US_01_2_Task1: Assess existing characters' choices and the final report.
5 Sep
DX HH ZE
US_02_1_Task1: When other players' making decisions, send the choice to back end.
3 Sep
JH YB ZE
US_01_1_Task4: Limit one turn game with 10 minutes.
29 Aug
JH YB
US_01_1_Task2: Assess the existing decision tree.
29 Aug
HH JH YB
US_01_1_Task1: Exploit the game and write a report for user experience.
31 Aug
DX ZE
Add a card
Add another list


Board
GE_SPRINT3
COMP90082-2022-SM2-GE-RedBack
Workspace visible
HH DX JH YB ZE
Share
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
DX HH ZE
US_01_Task2: Show everyone's choices as well as the report in a specific interface.
30 Sep
JH YB
US_02_Task1: Assess existing rejoin functionality.
7 Oct
DX ZE
US_02_Task2: Reproduce bugs and fix.
7 Oct
JH YB
US_02_Task3: Testing
7 Oct
HH
US_03_Task1: FInd out reasons why game gets stuck somehow during the game.
14 Oct
DX YB
US_03_Task2: reproduce bugs and fix.
14 Oct
HH ZE
US_03_Task3: Testing
14 Oct
JH
US_04_Task1: Add a timer on each choice when playing the game, and set it for 60 seconds.
20 Oct
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.
20 Oct
JH YB ZE
US_04_Task3: Inform player that time is out, and show the random choice made for player.
Add a card
Doing
Add a card
Done
Add a card
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.

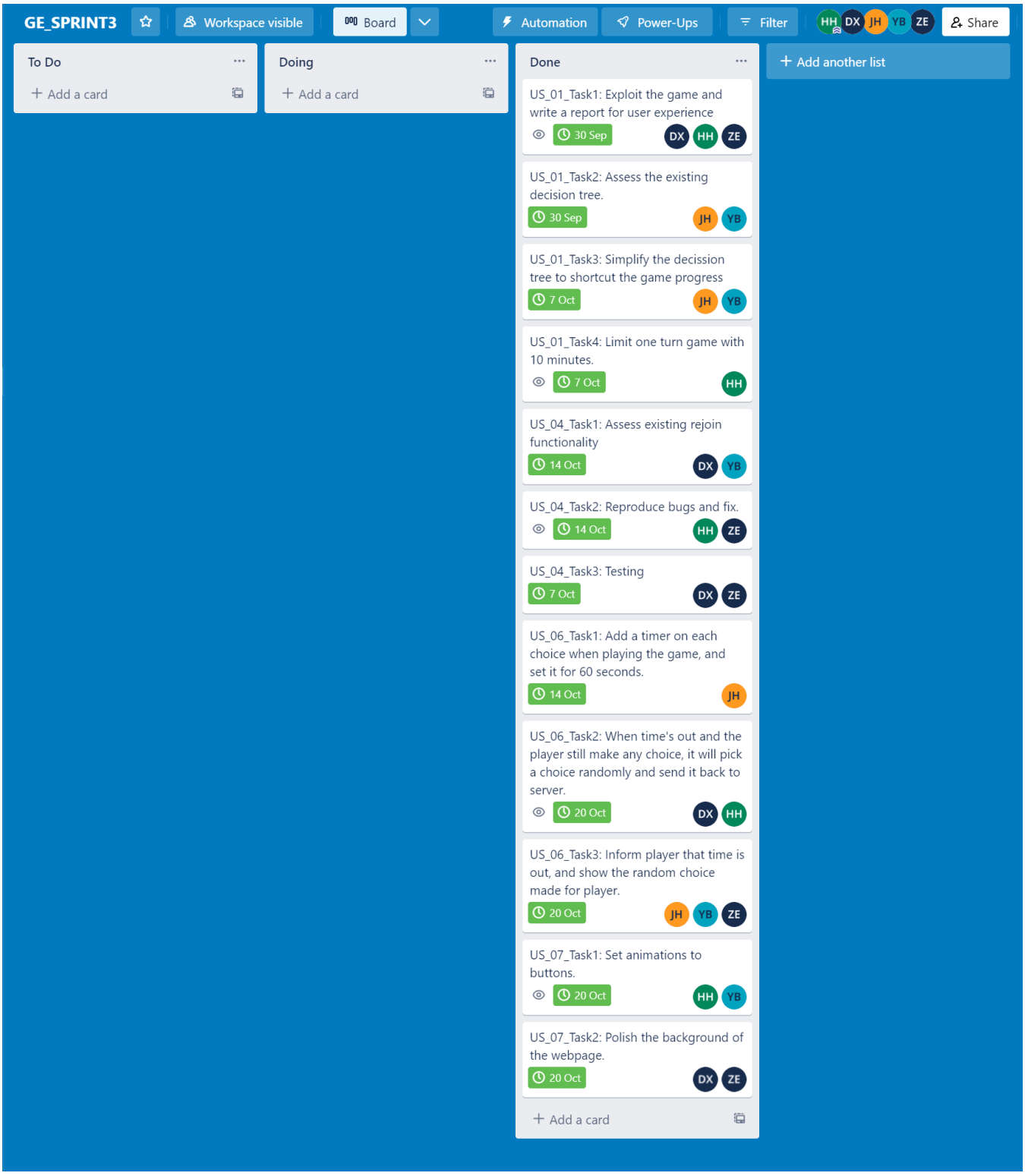
We modified the structure of decision tree based on Sprint 2. However, the workload of the implementation of the decision tree exceeded our expectation.

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

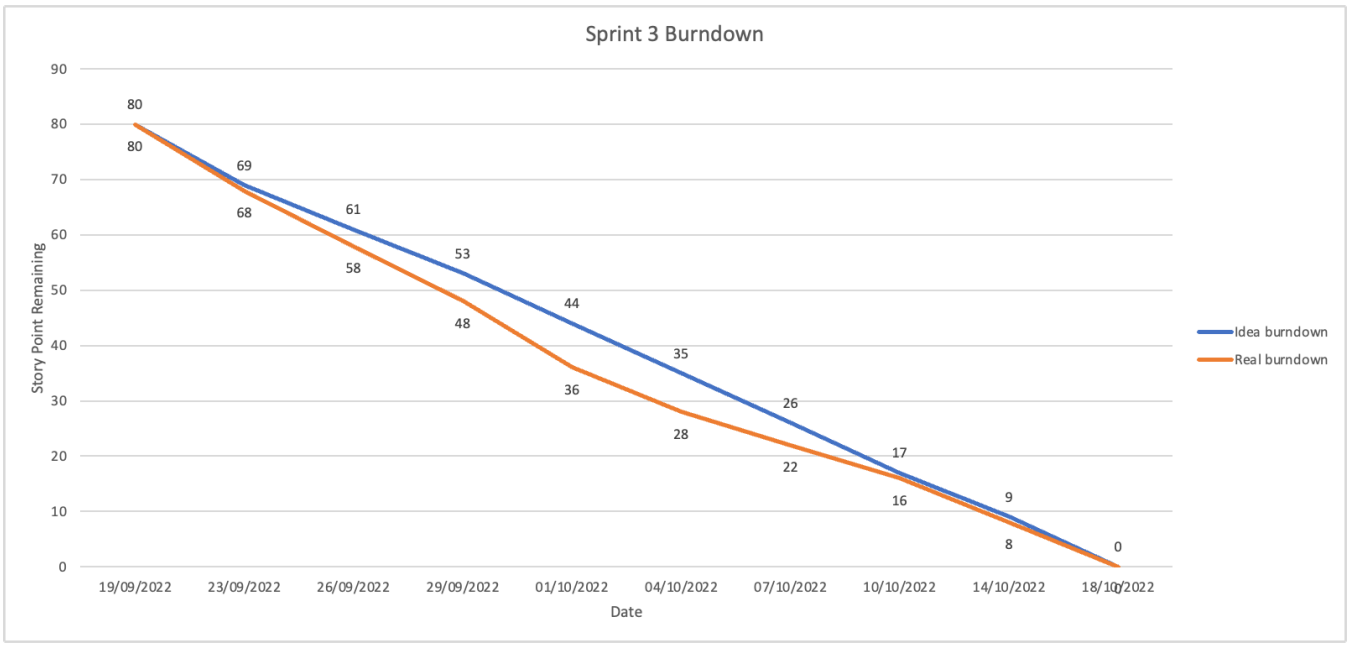
Moreover, we improved the UI to make the game more user-friendly and attractive for players.

Sprint	User Story	Decomposed Case	Tasks	Point	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 decission 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	Deleted.	n/a - Previous choices no longer taken to be a requirement due to the complexity of impementation
			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


GE_SPRINT3
Workspace visible
Board
Automation
Power-Ups
Filter
HH DX JH YB ZE
Share
To Do
Add a card
Doing
Add a card
Done
US_01_Task1: Exploit the game and write a report for user experience
30 Sep
DX HH ZE
US_01_Task2: Assess the existing decision tree.
30 Sep
JH YB
US_01_Task3: Simplify the decission 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
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 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 **Javascipt 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.

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 emerging 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

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`

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 questions 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

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 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://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

Meeting Minutes

Supervisor meeting: once a week

Client meeting: depending on the client

Team casual meeting: to be decided

Note: In this meeting minutes, major meetings are documented in this section.

Client Meetings

Meeting1_09_08_2022

Date: 2022-8-9

Time: 01:00 - 02:00 pm

Online Zoom Meeting

Topics

Have a detailed discussion about the previous project and gather requirements from Simon and Eduardo.

Priority	Topic	Time
1	Project introduction from the client.	10-15mins
2	Questions & clarification	10-15mins
3	Requirements gathering	30mins

Key Points Record

Game Background

1. Promoting conversation
2. Develop from student to student

Scope

Plan to run in tut and workshop

When playing, they will have chat and know what's will lead to crash. The game has several roles. Different people have different perspective. Keep thinking and make the decision. Currently has lots of questions. What's the good number of the question?

Simplify the game

The purpose is to be good in running the game

Two great reasons to do this project

1. Keep this project aliveadd features to this project.
2. Opportunity

Where the choices come from?

People is able to try different roles, and make different decision, that will have cost and benefits, should not be the obvious. Maybe be patience and safety, reach the definition. The decisions can be found already in the confluence.

Will others know your choice?

Not sure. Everyone will get their choices and start discussion. It's acceptable to lie to keep others safety.

The background is the airplane facing the disaster. There will be multiple problems during the flight.

Time limit

Shouldn't be like 50 questions, every turn players will need to discuss. Don't have the magic number right now. The process can't be too long.

Two parts:

First is to do the decision, the second part is to tell why choosing this decision. Players should think about why to make this decision and somehow justify.

Can we add more questions?

The game already has strong cases, let's get the work flows well. We should focus on more in make the schedule fluency.

Should two teams work separately?

Both teams need to develop scopes. For now, we can share the resources to make the game deployed. If the team can make the game run, for example, the environment, the team can share. After that, two team can use different branches. It’s too early to discuss this.

This week we should make this game deployed. Two weeks’ time will meet again. When we have something to show, maybe next Tuesday.

Supervisor Meetings

Meeting01_04/08/2022

Time: 2022-8-4 04:30 pm

Place: Online Zoom Meeting

Supervisor: Mauro Mello Jr

Team: (5/5) Eryi Zhou, Decheng Xu, Haiyu Hao, Yujie Bai, Jiazhe Hou and Team BoxJelly

Item	Topics
1	**Introduction** • Getting to know each other • Rules: • The supervisor helps through the process. Helps to estimate things, gives feedbacks, offers advise. • Meet with supervisor on a weekly basis.
2	**Sponsor (or client)** • Getting to know our sponsor • Communicating with our sponsor (when, how often, how, where) • Etiquette for meetings • cc'ing supervisor in every communication with your sponsor • Documenting meetings with sponsor
3	**Product and the project that will create it** • Product × project — know product well and prepare a better project • The four W's and one H: What is it, When is it needed, Where is it needed, How much… • A neglected W: Why is it needed? • Milestones that should know • Funding for ther project • Getting things done — what needs to be done, who does what, when to do what needs doing, where the doing is done • Keeping track of everything that is happening • Team internal meetings • Weekly meetings with supervisor
4	**Roles** • Project roles • Minimum roles to have as a starting point (e.g., product owner, scrum master) • Collaborate with, support and back each other up! • Communicating with supervisor, the sponsor and the team (as well as with program coordinator and lecturer)
5	**Any other business** • Any other business

Work to Do

1. Review the milestones and deliverables, then work backward to find out what need to do, and send a *summary* on or before 18:00 on Tue 09/05/2022 to supervisor.

2. Determine how the team would like to meet with supervisor, likely after meeting with sponsor.

3. Start discussing what roles team member would like to attempt.

4. Check What Is The Plan-Do-Check-Act (PDCA) Cycle? and PDCA, then google around for more details (lean management, continuous quality improvement, etc.).

5. Adopt this motto (until another one comes along): **The learning is in the doing.**

6. Take it easy, enjoy the learning.

Meeting02_12/08/2022

Time: 2022-2-12 3:30 pm

Place: Online Zoom Meeting

Supervisor: Mauro Mello Jr

Team: (5/5) Eryi Zhou, Decheng Xu, Haiyu Hao, Yujie Bai, Jiazhe Hou and Team BoxJelly

Item	Topics
1	Validate how the project goes: 1. Have deployed the program locally 2. Getting familiar with the source code 3. Planning user stories
2	Problems so far: 1. Online game link offered by supervisor doesn't work 2. All team members are not familiar with JavaScript 3. Game gets stuck some where when running locally
3	Check the check list for Sprint1 and follow the marking criteria to improve it.

Meeting03_19/08/2022

Time: 2022-8-19 3:45 pm

Place: Online Zoom Meeting

Supervisor: Mauro Mello Jr

Team: (4/5) Decheng Xu, Haiyu Hao, Yujie Bai, Jiazhe Hou and Team BoxJelly

Absense Reason: Eryi Zhou is on his flight to Melbourne.

Item	Topics
1	Validate how the project goes: 1. Confluence structured 2. User story planned
2	Problems so far: 1. Clarify both team RedBack and Boxjelly's scopes of delivery for each sprint and the whole project.
3	Works to do before sprint1 ddl 1. Perfect confluence space 2. Review the check list and marking criteria before submission

Meeting04_26/08/2022

Time: 2022-8-26 3:45 pm

Place: Online Zoom Meeting

Supervisor: Mauro Mello Jr

Team: Eryi Zhou, Decheng Xu, Haiyu Hao, Yujie Bai, Jiazhe Hou and Team BoxJelly

Item	Topics
1	Review the submission checklist and deliverables on GitHub
2	What have achieved so far: 1. Conclude the feedback from Sprint 1 and polish the documentation according to the feedback. 2. Group hacky hour to get familiar with the code, play the game together on cloud platform.
3	What need to do in the coming week 1. Divide into two groups to solve code problems and simplify game process.

Meeting05_01/09/2022

Time: 2022-9-2 4:00 pm

Place: Online Zoom Meeting

Supervisor: Mauro Mello Jr

Team: Eryi Zhou, Decheng Xu, Haiyu Hao, Yujie Bai, Jiazhe Hou and Team BoxJelly

Item	Topics
1	What have achieved so far: 1. Fixed the bug of the game. 2. Draw the decision tree.
2	What need to do in the coming week 1. Simplify the decision tree.

Meeting06_09/09/2022

Time: 2022-9-9 4:00 pm

Place: Online Zoom Meeting

Supervisor: Mauro Mello Jr

Team: Eryi Zhou, Decheng Xu, Haiyu Hao, Yujie Bai, Jiazhe Hou and Team BoxJelly

Item	Topics
1	Review the submission checklist and deliverables on GitHub
2	What have achieved so far: 1. Draw the simplified decision tree.
3	What need to do in the coming week 1. Implement the new decision tree in back end.

Meeting07_16/09/2022

Time: 2022-9-16 3:45 pm

Place: Online Zoom Meeting

Supervisor: Mauro Mello Jr

Team: Eryi Zhou, Decheng Xu, Haiyu Hao, Yujie Bai, Jiazhe Hou and Team BoxJelly

Item	Topics
1	Review the submission checklist and deliverables on GitHub
2	Review our feedback on the process of providing 360º feedback • Most of us complete feedback in 5-10 minutes.

Meeting08_21/09/2022

Time: 2022-9-21 14:00 pm

Place: Online Zoom Meeting

Supervisor: Mauro Mello Jr

Team: Eryi Zhou, Decheng Xu, Haiyu Hao, Yujie Bai, Jiazhe Hou and Team BoxJelly

Item	Topics
1	Share our learnings in Sprint2
2	Task planning and allocation in Sprint3.

Meeting09_07/10/2022

Time: 2022-10-12 15:00 pm

Place: Online Zoom Meeting

Supervisor: Mauro Mello Jr

Team: Eryi Zhou, Decheng Xu, Haiyu Hao, Yujie Bai, Jiazhe Hou and Team BoxJelly

Item	Topics
1	Review Sprint3 checklist
2	Review the requirements for the presentation and a draft checklist.
3	Review the requirements for sprint 4
4	Present progress on sprint 3

Meeting10_07/10/2022

Time: 2022-10-14 16:30 pm

Place: Online Zoom Meeting

Supervisor: Mauro Mello Jr

Team: Eryi Zhou, Decheng Xu, Haiyu Hao, Yujie Bai, Jiazhe Hou and Team BoxJelly

Item	Topics
1	Review final presentation *draft* checklist.
2	Present a status update of product and project components for the final presentation.

Team Meeting

Meeting01_03/08/2022

Time: 2022-8-3 8:00 pm

Place: Online Zoom Meeitng

Team: (5/5) Eryi Zhou, Decheng Xu, Haiyu Hao, Yujie Bai, Jiazhe Hou

Item	Topics
1	Team members introduce each othrer
2	Project overview
3	Determine meeting items with supervisor next day

Meeting02_08/08/2022

Time: 2022-8-8 8:00pm

Team: (5/5) Eryi Zhou, Decheng Xu, Haiyu Hao, Yujie Bai, Jiazhe Hou

Item	Topic
1	Exploit the game online via the link Outcome: Failed. The link doesn't work.
2	Determine items talked with sponsor next day
3	View the source code Outcome: Program coded by Vue, which is not familiar with for all team members.

Meeting03_10/08/2022

Time: 2022-8-10 8:00 pm

Place: Online Zoom Meeitng

Team: (5/5) Eryi Zhou, Decheng Xu, Haiyu Hao, Yujie Bai, Jiazhe Hou

Item	Topics
1	List all items to be delivered in Sprint1
2	Allocate work to each team member
3	Share problems coming with so far

Meeting04_15/08/2022

Time: 2022-8-15 9:00pm

Place: Online Zoom Meeitng

Team: (5/5) Eryi Zhou, Decheng Xu, Haiyu Hao, Yujie Bai, Jiazhe Hou

Item	Topics
1	Each team member shares what has done sofar and what needs to do
2	Exploit the game on local device: Outcome: Failed. The game gets stuck after choosing characters

Meeting05_17/08/2022

Time: 2022-8-17 12:00am

Place: Online Zoom Meeitng

Team: (5/5) Eryi Zhou, Decheng Xu, Haiyu Hao, Yujie Bai, Jiazhe Hou

Item	Topics
1	Check the checklist Outcome: GitHub page needs to be structured
2	Run the game on cloud platform Outcome: Failed. The game gets stuck after few questions attempted.
3	Assin work to each team member for sprint1

Meeting06_20/08/2022

Time: 2022-8-20 01:00 pm

Place: Online Zoom Meeitng

Team: (5/5) Eryi Zhou, Decheng Xu, Haiyu Hao, Yujie Bai, Jiazhe Hou

Item	Topics
1	Check the checklist Outcome: GitHub page needs to be structured
2	Run the game on cloud platform Outcome: Failed. The game may be failed because of front end issue.

Meeting07_24/08/2022

Time: 2022-8-24 01:00 pm

Place: Online Zoom Meeitng

Team: (5/5) Eryi Zhou, Decheng Xu, Haiyu Hao, Yujie Bai, Jiazhe Hou

Item	Topics
1	Allocate tasks for sprint 2 Outcome: Eryi, Decheng, and Jiazhe are mainly responsible for debug. Haiyu and Yujie draw the decision tree of the game.
2	Run the game on cloud platform Outcome: Failed. Found the bug that the game always got stuck in the last question.

Meeting08_29/08/2022

Time: 2022-8-29 01:00 pm

Place: Online Zoom Meeitng

Team: (5/5) Eryi Zhou, Decheng Xu, Haiyu Hao, Yujie Bai, Jiazhe Hou

Item	Topics
1	Each team member shares what has done sofar and what needs to do.
2	Review the feedback of Sprint 1.

Meeting09_31/08/2022

Time: 2022-8-31 8:00 pm

Place: Online Zoom Meeitng

Team: (5/5) Eryi Zhou, Decheng Xu, Haiyu Hao, Yujie Bai, Jiazhe Hou

Item	Topics
1	Run the game on cloud platform. Outcome: Successful. Bugs were solved.
2	Share problems coming with so far.

Meeting10_05/09/2022

Time: 2022-9-5 8:00pm

Team: (5/5) Eryi Zhou, Decheng Xu, Haiyu Hao, Yujie Bai, Jiazhe Hou

Item	Topics
1	Each team member shares what has done so far and what needs to be done.
2	Discuss the way to simplify the decision tree.

Meeting11_07/09/2022

Time: 2022-9-7 8:00pm

Team: (5/5) Eryi Zhou, Decheng Xu, Haiyu Hao, Yujie Bai, Jiazhe Hou

Item	Topics
1	Check the Sprint2 checklist Outcome: GitHub page needs to be structured
2	Draw the simplified decision tree.

Meeting12_12/09/2022

Time: 2022-9-12 8:00pm

Team: (5/5) Eryi Zhou, Decheng Xu, Haiyu Hao, Yujie Bai, Jiazhe Hou

Item	Topics
1	Check the Sprint2 checklist Outcome: GitHub page needs to be structured
2	Implement of the decision tree is divided into three parts: questions, options, and functions. This week will focus on the question part.

Meeting13_19/09/2022

Time: 2022-9-19 8:00pm

Team: (5/5) Eryi Zhou, Decheng Xu, Haiyu Hao, Yujie Bai, Jiazhe Hou

Item	Topics
1	Share problems coming with so far
2	This week will focus on the option part. Outcome: Question part of the new decision tree has been modified.

Meeting14_03/10/2022

Time: 2022-10-3 8:00pm

Team: (5/5) Eryi Zhou, Decheng Xu, Haiyu Hao, Yujie Bai, Jiazhe Hou

Item	Topics
1	Share problems coming with so far.
2	This week will focus on the frontend backend interaction of new decision process. Outcome: Question and option part of the new decision tree has been modified.
3	Check the Sprint3 checklist

Meeting15_10/10/2022

Time: 2022-10-10 8:00pm

Team: (5/5) Eryi Zhou, Decheng Xu, Haiyu Hao, Yujie Bai, Jiazhe Hou

Item	Topics
1	Share problems coming with so far.
2	New decision process still has some problems. This week will continue working on that.
3	Jiazhe and Decheng will polish the front-end UI.

Meeting16_12/10/2022

Time: 2022-10-12 8:00pm

Team: (5/5) Eryi Zhou, Decheng Xu, Haiyu Hao, Yujie Bai, Jiazhe Hou

Item	Topics
1	Share problems coming with so far.
2	New decision process still has some problems. This week will continue working on that.
3	Work on the front-end UI.
4	Do unit test for backend.

Meeting17_17/10/2022

Time: 2022-10-17 8:00pm

Team: (5/5) Eryi Zhou, Decheng Xu, Haiyu Hao, Yujie Bai, Jiazhe Hou

Item	Topics
1	Share problems coming with so far.
2	Work on the front-end UI.
3	Do unit test for frontend and backend.
4	Work on the presentation.