Bilkent University

Department of Computer Engineering

Senior Design Project

Analysis and Requirements Report

Project Name: Mutrivia

Team ID: T2316

Group Members: Lara Fenercioğlu, Gökhan Taş, Sebahattin Utku Sezer, Zeynep Büşra Ziyagil, Bedirhan Sakinoğlu

Supervisor: Halil Altay Güvenir

Innovation Expert: Melih Gezer

Nov 13, 2022

This report is submitted to the Department of Computer Engineering of Bilkent University in partial fulfillment of the requirements of the Senior Design Project course CS491/2.

Table of Contents

1. Introduction

Museums are an essential cultural heritage that educates people and improves their points of view. Nevertheless, many people visit museums as if they are on holiday trips, and they do not pay enough attention. People miss tons of information, and even though they read it carefully, it is lost after a time. To prevent these issues, museums have initiated projects using improved technologies such as mobile applications and artificial intelligence. For this purpose, gamification solutions increase the visitor's attention [1].

We plan to develop a mobile application that combines artificial intelligence and gamification to solve problems. We will build an interactive question-based game that can be played on mobile phones. Since generating questions for each object in a museum where, for instance, Rahmi Koç museum has over 5000 objects, is a time-consuming task; therefore, we will propose a natural language processing solution that will automatically generate questions from the text we provide. Then, these questions will be used to test people's knowledge and create a competitive environment.

2. Proposed System

2.1. Overview

Mutrivia will be a mobile application with two different types of clients: Museums and visitors. Museums will buy the application and be registered into the system. They will use it for their purposes, such as increasing the flow of people and sharing their artifacts' information so that we can generate interactive questions using NLP (Natural Language Process). When it comes to the user, they will play a question-based game after they finish a route or room in the museum. They can play either with a group or solo. They will earn a score for each correct answer and, according to the result, will earn some rewards. The museums will decide on these rewards.

During the game, each question will be generated instantaneously. There will be two difficulty levels, as people may not want to play a challenging game. Moreover, some kids and children can be easily bored when they face a tricky question, so we provide easy level choice. On the other hand, there are such artifacts that they do not have enough information about themselves. Also, some artifacts have subjective explanations. These types of artifacts are not included in our quiz parts of the application. Additionally, the information text is in English, and our questions will also be in English. The system will generate questions from the given text and provide four answers, one of them being the correct answer. One of our challenges will be generating false answer choices as distractions. The system must provide answers similar to the real answer; thus, the player should be distracted.

We have already taken sample texts from the Rahmi Koç Museum in Ankara. The sample text looks like the below (correct answers are highlighted):

“This is one of the first flags ordered by Mustafa Kemal Atatürk for the celebration of the declaration of the Turkish Republic on the 29th October 1923. It was bequeathed to Nilüfer Çavuşoğlu by her grandfather Zekai Apaydın, one of the first deputies of the Turkish National Assembly and erstwhile Minister of Defence.”

Sample multiple-choice questions and answers would be the following:
1- *“On which date was the Turkish Republic declared?”*
29 October 1923
30 October 1923
27 October 1923
28 October 1923

2- “Who bequeathed the flag to Nilüfer Çavuşoğlu?”
Nilüfer Çavuşoğlu's grandfather
Atatürk
Zekai Apaydın
Turkish National Assembly

Since we are planning to develop this application for global usage, the language of the application will be English. In the future, we plan to make a Turkish version of the quiz.

2.2. Functional Requirements

2.2.1. Client Side Mobile Application

- The application must allow users to create a party and become a host.
- The application must show a unique id to each party's host, letting other users join the party by using that id.
- The application must let users join a party by entering unique ids.
- The application must let users pick a nickname when joining a party and select a game level.
- The application must have the option to let the user play alone.
- The application must show questions and answer options to the user.
- The application must let the user select an answer option, show the correct answer and show how many points the user gets from that question.
- The application must show the user his/her score.
- The application must show the winner when the game is over.
- The application must provide an all-time leaderboard that shows top scorer users ordered by their score in a party.

2.2.2. Server Side

- The application should keep the information about the potential hosts and players, such as a nickname and score.
- The application must be able to generate questions from texts for each party or solo player.
- The application must calculate the number of points users get from their answers.
- The application must order the users of a party according to their score and decide the winner when the game is over.
- The application must generate a unique id for each party.
- The application must check entered party ids and assign users to parties according to the id entered.

2.3. Nonfunctional Requirements

2.3.1. Performance

We will produce each question in real-time, and it shouldn’t take more than 0.5 seconds for a smooth user experience. We will store only the required texts, but each time a user plays the game, a new question will be generated; thus, producing fastly is crucial so that the user doesn't wait too long. Also, it should be scalable because multiple museums will be enrolled in the system, and all requests should be handled without a bottleneck. To make the process as fast as possible, we will use server-side to generate questions and calculate users' points, as mobile phones are much slower at doing such calculations.

2.3.2. Usability

The mobile application will be used by many people and people of many ages. Therefore it should be user-friendly. Since we aim to make museums more entertaining, the application should not frustrate users. Also, it should look attractive and likable since it is a game. In short, a simple, easily understandable, and appealing interface will be provided.

The user interface should be simple, with no distracting themes. As the main aim of this project is to increase the interaction of people who visit museums with the exhibited objects, complex and detailed user interfaces may distract users from museum objects to only the application.

2.3.3. Extensibility

We will begin with a couple of museums. Therefore, the application should be adaptable to other museums that may be included in the future. In addition, we may add new functionalities in the future, such as different language options, different levels of difficulty, and different question types.

2.3.4. Rationality

The questions the application would generate using natural language processing must be accurate to the corresponding object description text. The questions should not be out of context from the objects’ descriptions. Also, the answer choices should be reasonable with the question, and the wrong answers should be compatible with the correct answer so that it can distract the player.

2.3.5. Portability

This application is specialized for smart mobile devices only. There will not be any web or desktop version of this project. This project will support both Android and iOS operating systems.

2.3.6. Security

All the data collected from museums will be kept securely. No third-party user is going to be accessing this private data.

2.3.7. Reliability

The application must not have extended server downtime to give users an uninterrupted experience. It also must not be laggy and have delays. As the users using the application will be friendly and competitive while trying to answer the questions, the sessions and users' local experience must not be interrupted for long or frequently.

2.4. Pseudo Requirements

- MySQL will be used as a relational database.
- Git and GitHub will be used as a version control system and issue tracking.
- The application will support both iOS and Android operating systems.
- Flutter will be used for the front end of the mobile application.
- Python will be used for NLP service.
- Microservice architecture will be used.
- Google APIs and several Python libraries will be used for NLP.
- Java Spring Boot will be used for the backend and APIs.
- Rest API will be used.
- AWS will be used for the server.
- Apache Kafka will be used as a message queue to enable communication between microservices
- Asana will be used for job distribution.

2.5. System High-Level Architecture

Below, you can see the high-level architecture of our project which includes and shows pseudo requirements.

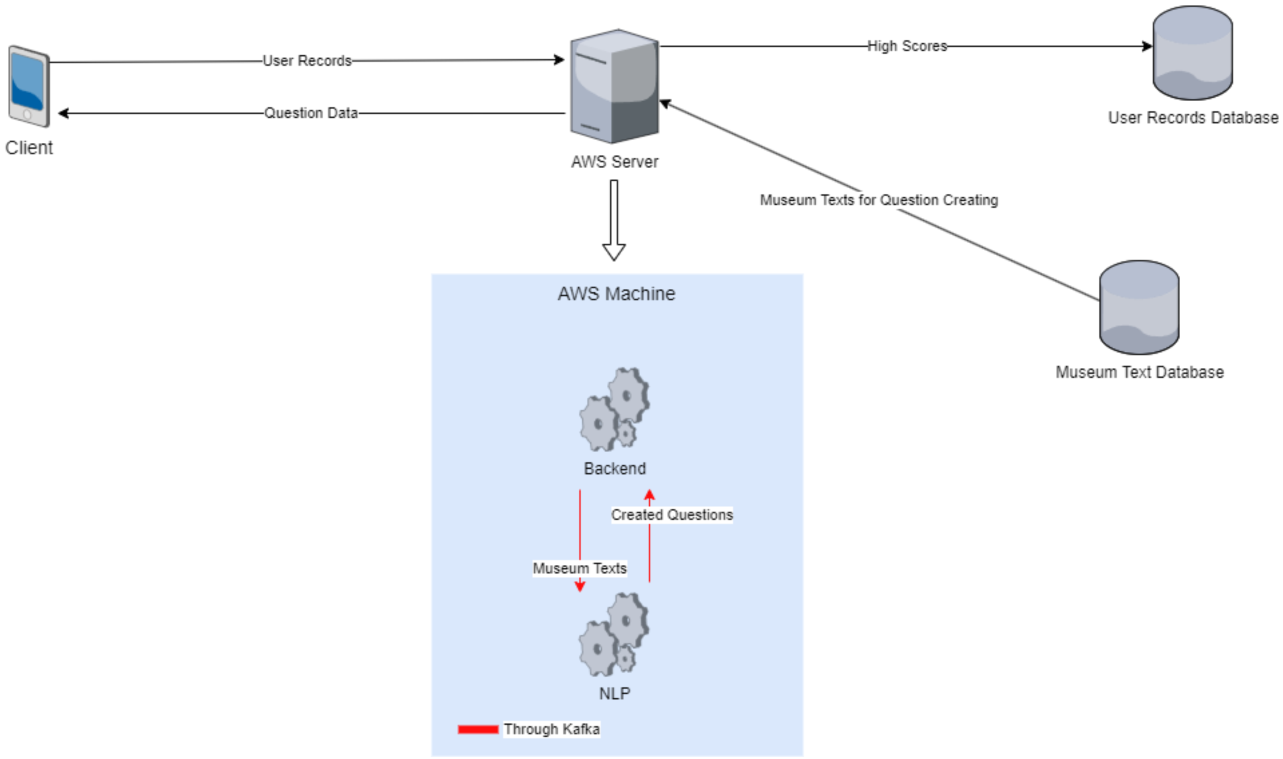

Figure 1: Mutrivia High-Level Architecture

2.6. System Models

2.6.1. Scenarios

Scenario 1	Join A Session
Participating Actor	Visitor
Entry Condition	Visitor is on the main menu
Exit Condition	- Visitor clicks on leave session or - Host ends the quiz
Main Flow of Events	1. The user will open the application 2. Visitor clicks join session 3. The visitor enters session ID 4. The visitor enters a nickname 5. The session will be started. 6. Questions will be shown one by one, and the user will answer. 7. The user will see the scoreboard and continue playing if the quiz is not finished. 8. The user will see the final scoreboard if the quiz is finished.

Table 1: “Join A Session” Scenario

Scenario 2	Create Session
Participating Actor	Host
Entry Condition	Host is on the main menu
Exit Condition	- Host clicks end session or - Session is finished
Main Flow of Events	1. The user will open the application. 2. The user will choose to create a session. 3. The session id will be generated and shown to the user. 4. The user will choose whether to play or watch the session. 5. If the “watch” is selected

	scoreboard will be shown. 6. If play is selected, the user will enter a nickname. 7. Questions will be shown one by one, and the user will answer. 8. Questions will end, and the session scoreboard will be shown to all. 9. Scoreboard leaders will be notified.

Table 2: “Create Session” Scenario

Scenario 3	Play Solo
Participating Actor	A Visitor
Entry Condition	Visitor is on the main menu
Exit Condition	Quiz is finished
Main Flow of Events	1. The user will open the application. 2. The user will enter his/her nickname. 3. The session will be started. 4. Questions will be shown individually, and the user will answer. 5. The user will see the leaderboard if the quiz is finished.

Table 3: “Play Solo” Scenario

Scenario 4	View Leaderboard
Participating Actor	All Users
Entry Condition	The user is on the main menu
Exit Condition	The user gets back to the main menu
Main Flow of Events	1. The user can choose to view the leaderboard from the main menu. 2. The user views the leaderboard.

Table 4: “View Leaderboard” Scenario

2.6.2. Use Case Model

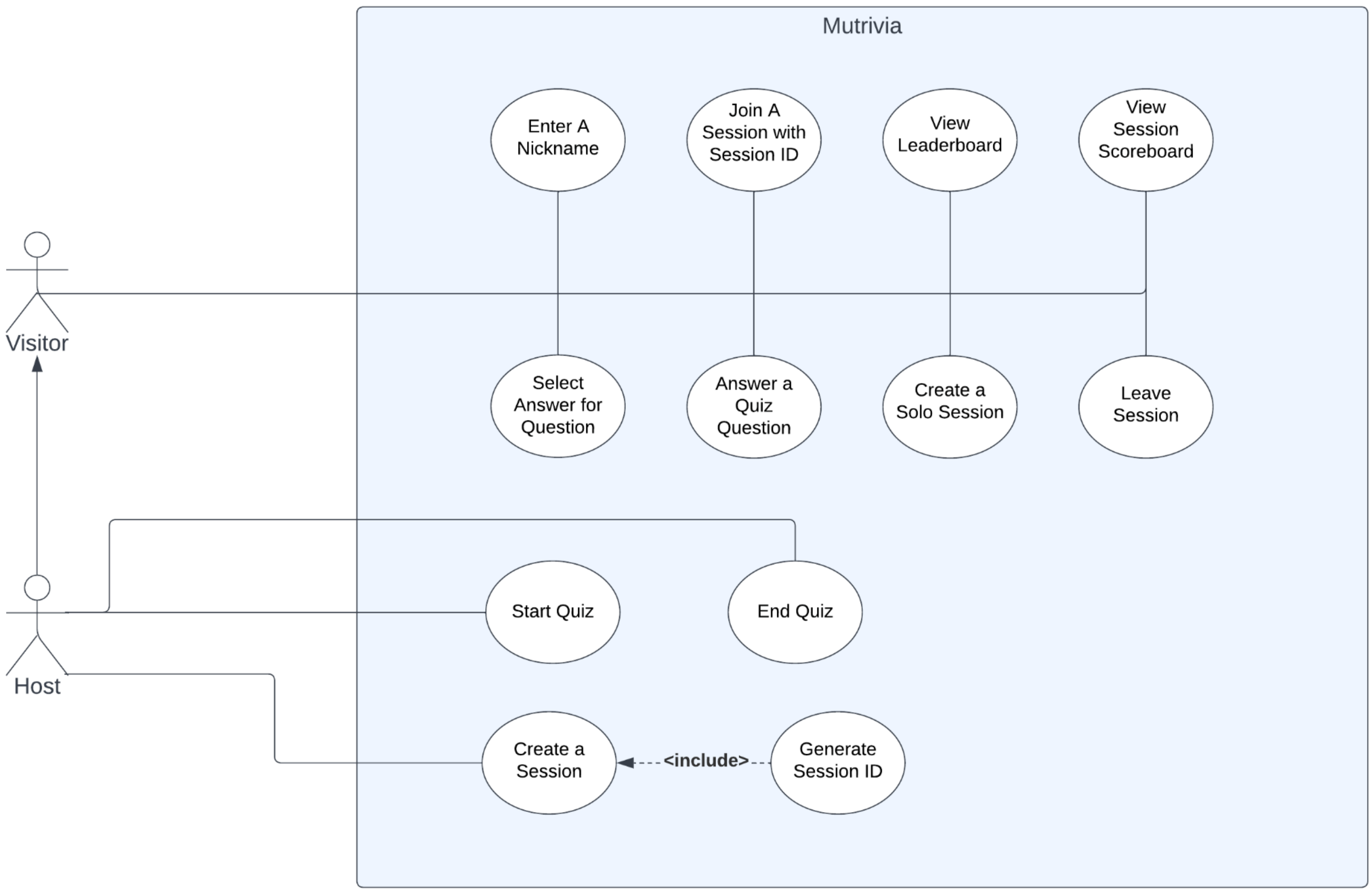

Figure 2: Use Case Diagram of Mutrivia

2.6.3. Object and Class Model

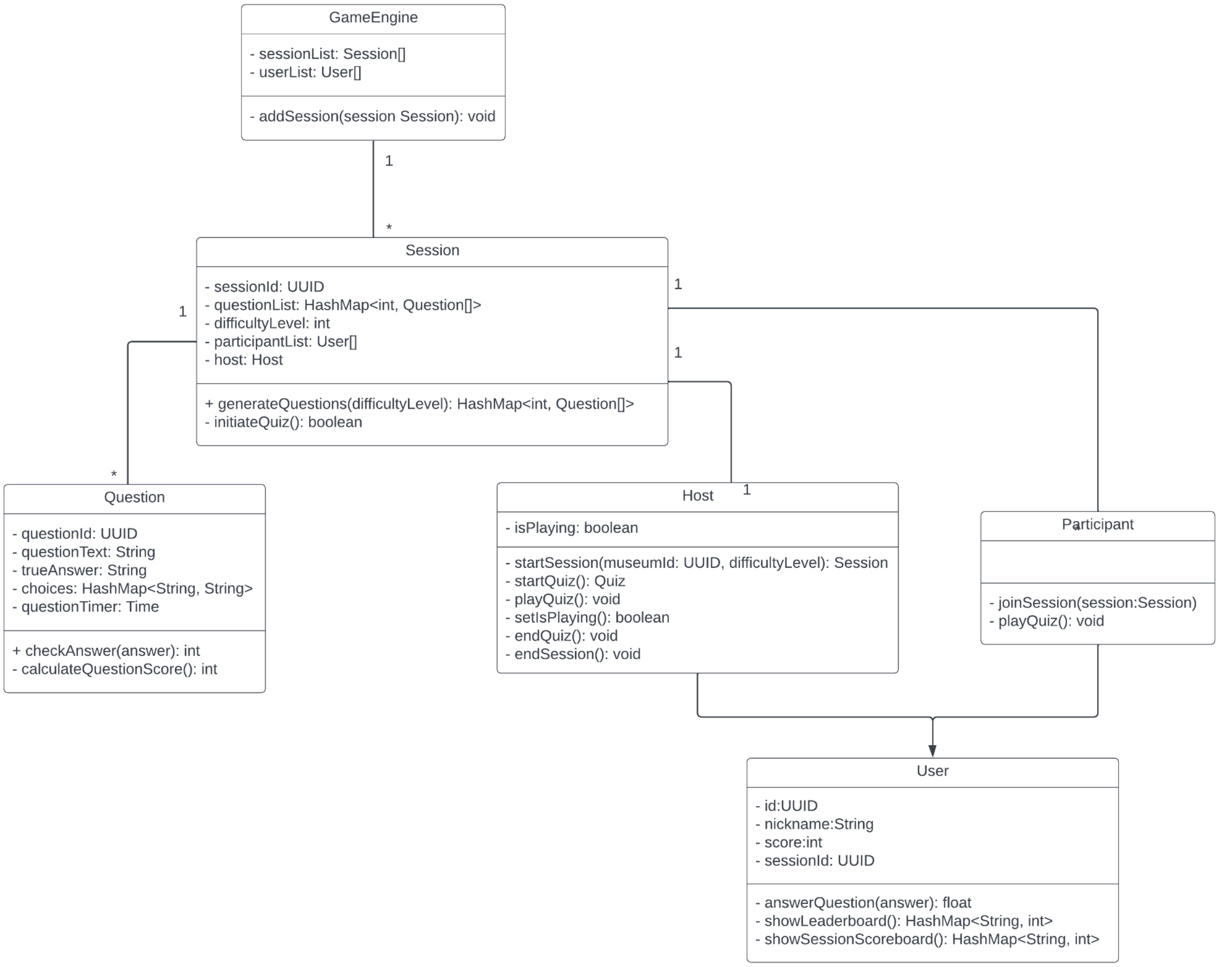
GameEngine
- sessionList: Session[]
- userList: User[]
- addSession(session Session): void
1
*
Session
- sessionId: UUID
- questionList: HashMap<int, Question[]>
- difficultyLevel: int
- participantList: User[]
- host: Host
+ generateQuestions(difficultyLevel): HashMap<int, Question[]>
- initiateQuiz(): boolean
1
1
1
*
Question
- questionId: UUID
- questionText: String
- trueAnswer: String
- choices: HashMap<String, String>
- questionTimer: Time
+ checkAnswer(answer): int
- calculateQuestionScore(): int
Host
1
- isPlaying: boolean
- startSession(museumId: UUID, difficultyLevel): Session
- startQuiz(): Quiz
- playQuiz(): void
- setIsPlaying(): boolean
- endQuiz(): void
- endSession(): void
Participant
- joinSession(session:Session)
- playQuiz(): void
User
- id:UUID
- nickname:String
- score:int
- sessionId: UUID
- answerQuestion(answer): float
- showLeaderboard(): HashMap<String, int>
- showSessionScoreboard(): HashMap<String, int>

Figure 3: Class Diagram of Mutrivia

2.6.4. Dynamic Models

2.6.4.1. Activity Diagrams

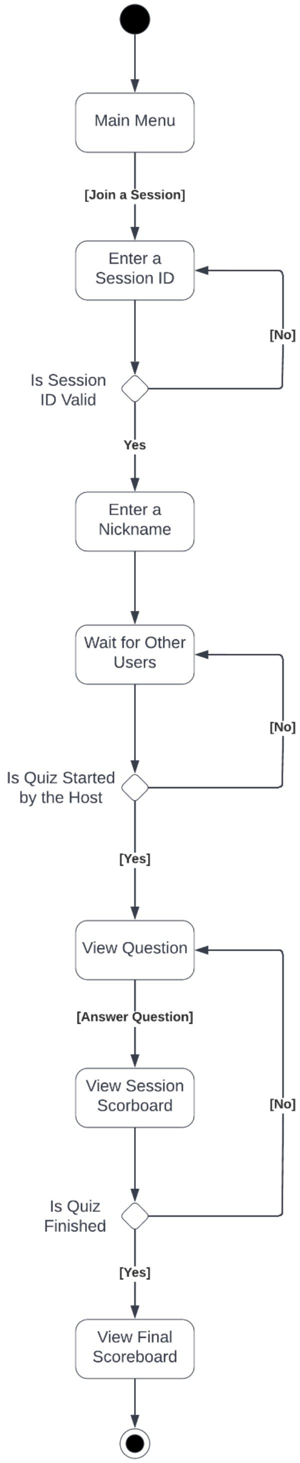

Figure 4: Activity Diagram of playing a group game

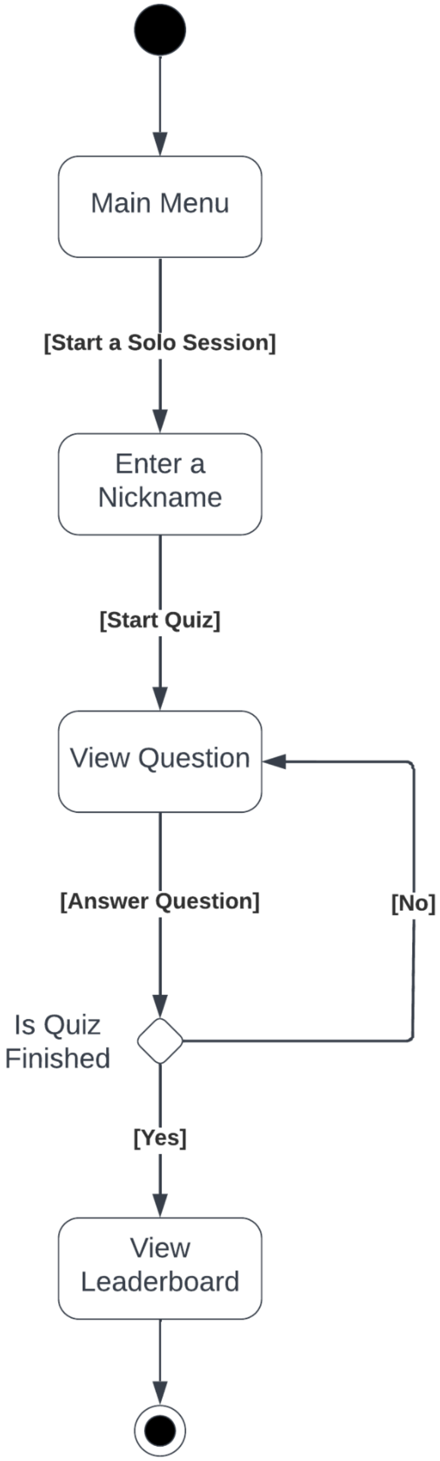

Figure 5: Activity Diagram of playing a solo game

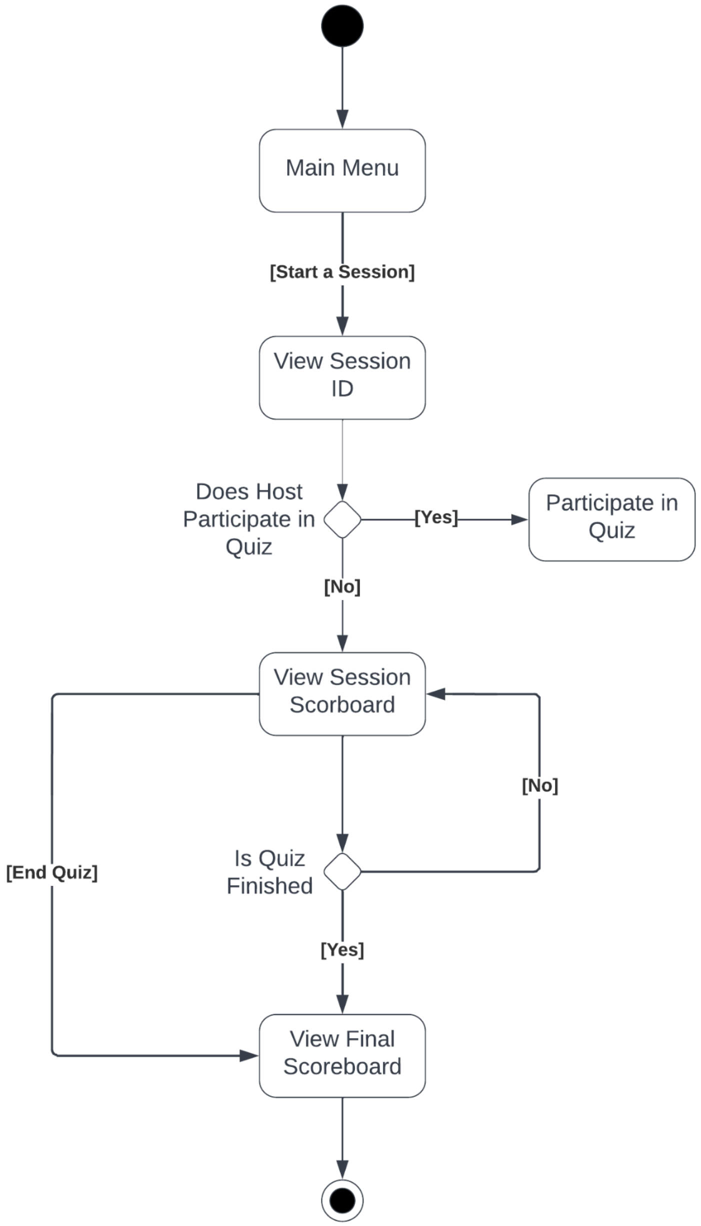

Figure 6: Activity Diagram of hosting a game session

2.6.4.2. Sequence Diagrams

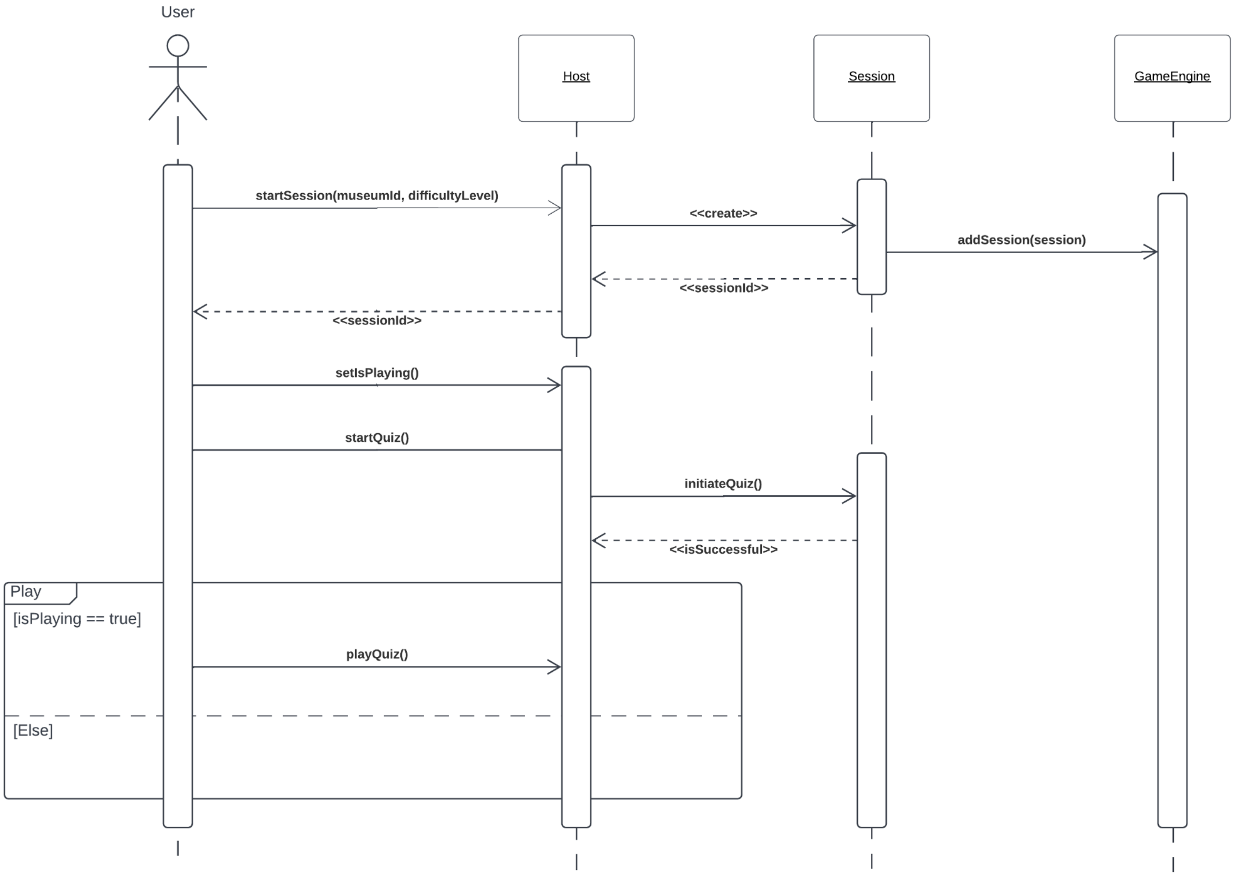

Figure 7: Sequence Diagram of starting a session

2.6.5. User Interface - Navigational Paths and Screen Mock-ups

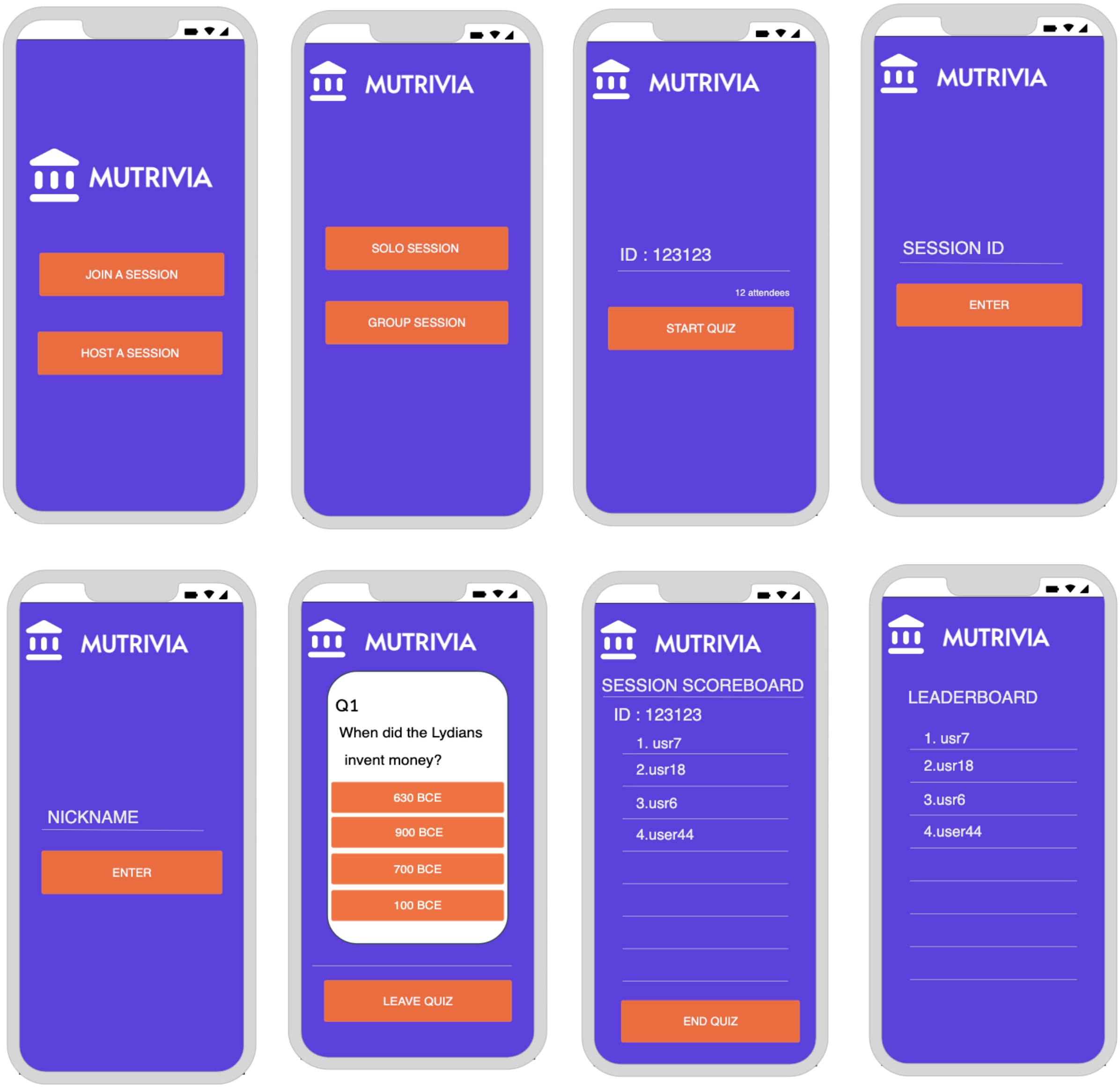

Figure 8: User Interfaces of Mutrivia

3. Other Analysis Elements

3.1. Consideration of Various Factors in Engineering Design

As part of our analysis, we consider how our system will be constrained or address the needs of factors like public health, safety, and welfare, as well as global, cultural, social, environmental, and economic factors.

3.1.1. Public Health Considerations

Mutrivia is an application aiming to enhance the museum trip experience of visitors, making it more fun and informative. There is no direct effect of Mutrivia on public health. However, the pandemic might have a negative impact since Mutrivia can increase the flow of people in museums.

3.1.2. Public Safety Considerations

Mutrivia is an application aiming to enhance the museum trip experience of visitors, making it more fun and informative. There is no direct effect of Mutrivia on public safety.

3.1.3. Public Welfare Considerations

Mutrivia is an application aiming to enhance the museum trip experience of visitors, making it more fun and informative. There is no direct effect of Mutrivia on public welfare.

3.1.4. Global Considerations

The target audience for Mutrivia can be everyone around the globe. Since every museum can support Mutrivia at no additional cost, everyone can use Mutrivia in the supported museums. The application will be implemented in English to be accessible by any museum. The data of museums will be collected in English to be compatible globally.

3.1.5. Cultural Considerations

The app aims to increase visitor interaction in museums by including gamification applications in the tour. The app experience will enhance the visitor’s engagement, and the user will learn more from experience due to the engagement level increment. Through this process, users' cultural engagements will be augmented. From a cultural background view, since the application will be in English, commonly used in museums worldwide and the most inclusive language from that perspective, there will not be discrimination issues. Users' personal beliefs or preferences will not be collected or used in any way.

3.1.6. Social Considerations

In the app, the only information asked is the username, and it does not have to be the real name of the user; they all will be considered nicknames. From a social perspective, since no personal data of any user is saved in the platform, there are no concerns for the sake of the user's privacy.

3.1.7. Environmental Considerations

Mutrivia has no environmental considerations, such as reducing its carbon footprint. So, the application has no direct or indirect effects that can affect environmental issues.

3.1.8. Economic Considerations

The application is expected to be free for users. An annual subscription must be bought for a museum to be added to Mutrivia for a year. To make the application free for users, development costs must be minimized. Some free alternatives, such as adding ads to applications, can be considered if needed.

	Effect Level	Effect
Public Health	2	Mutrivia may increase the number of visitors during a pandemic.
Public Safety	0	No effect
Public Welfare	0	No effect
Global Factors	5	English is used as it is a global language. The addition of a museum should

		only require data in English.
Cultural Factors	9	No discrimination will be made in terms of language and questions as the same algorithm will be used for every museum
Social Factors	3	Mutrivia does not require personal data.
Environmental Factors	0	No effect
Economic Factors	5	Costs must be minimized. Free alternative ways may be considered.

Table 5: Table of factors and effects

3.2. Risks and Alternatives

In this section, the possible risks and alternative ways (B plan) to handle these risks will be discussed.

3.2.1. Time Management

Mutrivia has a predetermined and rigid schedule, even though there is a short amount of time to finish the project. The project's architecture and planning should be constructed with extreme attention. After selecting the fundamental technical systems, we will prioritize key features and functionalities to prevent failure. The project's priorities will determine the timeframe for which the schedule will be created.

3.2.2. Implementation

To generate questions immediately, NLP(Natural Language Processing) will be used. Since it is an AI-based technology, implementation of NLP will be a challenging area since it is not a technology the team is familiar with. To eliminate the risk of failure, courses will be studied, and experiments using NLP will be conducted.

3.2.3. Work Distribution

Five computer science undergraduates are involved in the project, indicating that five individuals are available to fulfill the project's criteria. On the other hand, there may be instances when the five students cannot meet due to time constraints caused by their heavy workloads. Due to the organization being required to assemble all of the project's components, it does not appear that breaking the task into sections and doing them independently is a particularly effective strategy. The project will be carried out step by step to resolve this issue, with brief gatherings and discussions on the next steps. There is also the possibility that some group members will need to become more familiar with certain project-related concepts and technology. The planned systems and technologies will be specified immediately to reduce the time lost during the implementation phase. On this basis, the members will have the opportunity to construct a foundation of information.

Risk	Likelihood	Effect
Time management	Medium	Failure to complete functionalities
Implementation	Low	Failure to implement some features
Work Distribution	Low	Failure to equally distribute tasks and implement features

Table 6: Table of risks, their likelihoods and effects

3.3. Project Plan

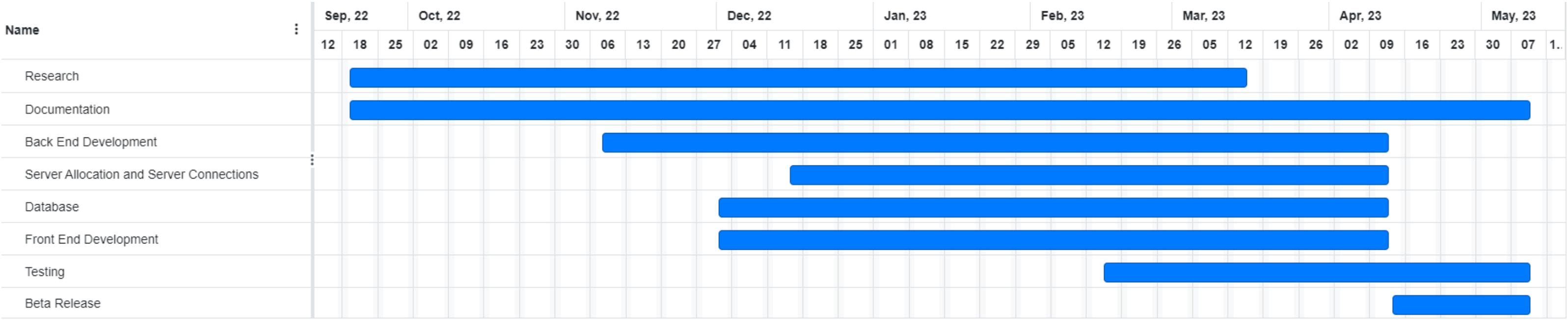

Figure 9: Project Plan Gantt Chart (https://imgur.com/a/aAO1tm3)

WP (Work Package)	WP Title	Leader	Members	Deadline
WP 1	Research	Lara Fenercioğlu	All Members	Mar. 15, 2023
WP 2	Documentation	Utku Sezer	All Members	May 10, 2023
WP 3	Back End Development	Bedirhan Sakinoğlu	Utku Sezer, Gökhan Taş	Apr. 12, 2023
WP 4	Server Allocation and Server Connections	Utku Sezer	Gökhan Taş, Bedirhan Sakinoğlu	Apr. 12, 2023
WP 5	Database Implementation	Lara Fenercioğlu	Gökhan Taş, Zeynep Ziyagil	Apr. 12, 2023
WP 6	Front End Development	Gökhan Taş	Bedirhan Sakinoğlu, Lara Fenercioğlu, Zeynep Ziyagil	Apr. 12, 2023
WP 7	Testing	Zeynep Ziyagil	Lara Fenercioğlu	May. 10, 2023
WP 8	Project Demo	Bedirhan Sakinoğlu	All Members	Dec. 25, 2022
WP 9	Beta Release	Gökhan Taş	All Members	May. 10, 2023

Table 7: Work package distribution

WP 1: Research
Start Date: Sep. 19, 2022 | End Date: Mar. 15, 2023
Leader: Lara Fenercioğlu Members Involved: Gökhan Taş, Bedirhan Sakinoğlu, Utku Sezer, Zeynep Ziyagil
Objectives: Try to understand the problem domain which is Natural Language Processing (NLP).
Tasks: Task 1.1: Reading articles related to NLP Task 1.2: Researching similar projects done Task 1.3: Setting up requirements Task 1.4: Testing before implementation
Deliverables: Deliverable 1.1: Packages and requirements for NLP

Table 8: Research work package

WP 2: Documentation
Start Date: Sep. 19, 2022 | End Date: May. 10, 2023
Leader: Utku Sezer Members Involved: Gökhan Taş, Bedirhan Sakinoğlu, Lara Fenercioğlu, Zeynep Ziyagil
Objectives: To make the developers be aware of what the system will propose and make the stakeholders understand and comprehend the project.
Tasks: Task 2.1: Creating project website Task 2.2: Creating diagrams and mock models Task 2.3: Getting feedback from the project advisor Task 2.4: Writing the Project Specifications Report Task 2.5: Writing the Analysis and Requirement Report Task 2.6: Writing the Final Report
Deliverables: Deliverable 2.1: Project Specifications Report Deliverable 2.2: Analysis and Requirement Report Deliverable 2.3: Final Report

Table 9: Documentation work package

WP 3: Back End Development
Start Date: Nov. 8, 2022 | End Date: Apr. 12, 2023
Leader: Bedirhan Sakinoğlu Members Involved: Utku Sezer, Gökhan Taş
Objectives: Implementing back end related functionalities.
Tasks: Task 3.1: Setting up framework environment Task 3.2: Creating core classes and functions Task 3.3: Connecting with front end Task 3.4: Connecting with database and server
Deliverables: Deliverable 3.1: Back end of Mutrivia

Table 10: Back End Development work package

WP 4: Server Allocation and Server Connections
Start Date: Dec. 15, 2022 | End Date: Apr. 12, 2023
Leader: Utku Sezer Members Involved: Bedirhan Sakinoğlu, Gökhan Taş
Objectives: Finding appropriate server, allocating it, and deploying the code for natural language processing calculations.
Tasks: Task 4.1: Allocating appropriate server Task 4.2: Setting up server environment Task 4.3: Deploying code for server Task 4.4: Creating connections with server Task 4.5: Testing server
Deliverables: Deliverable 4.1: Server

Table 11: Server Allocation and Server Connections work package

WP 5: Database Implementation
Start Date: Sep. 19, 2022 | End Date: Mar. 15, 2023
Leader: Lara Fenercioğlu Members Involved: Gökhan Taş, Zeynep Ziyagil
Objectives: Delivering relationship of entities in the database.
Tasks: Task 5.1: Creating the database Task 5.2: Specifying the functionalities Task 5.3: Designing database with entity relationship diagram
Deliverables: Deliverable 5.1: Database implementation document Deliverable 5.2: Database design document

Table 12: Database Implementation work package

WP 6: Front End Development
Start Date: Dec. 1, 2022 | End Date: Apr. 12, 2023
Leader: Gökhan Taş Members Involved: Bedirhan Sakinoğlu, Lara Fenercioğlu, Zeynep Ziyagil
Objectives: User experience research, group design of the displays, creation, and implementation of the interface using a chosen language.
Tasks: Task 6.1: Researching user experience Task 6.2: Creating mock-ups Task 6.3: Implementing the user interfaces with defined languages
Deliverables: Deliverable 6.1: Design mockups Deliverable 6.2: User interface in both Android and iOS

Table 13: Front End Development work package

WP 7: Testing
Start Date: Feb. 15, 2023 | End Date: May 10, 2023
Leader: Zeynep Ziyagil Members Involved: Lara Fenercioğlu

Objectives: Ensure the functionality of all modules and integrate their functionality into the to-be-tested version.
Tasks: Task 7.1: Logging the flow Task 7.1: Writing test codes Task 7.2: Analyzing the bugs and errors Task 7.3: Fixing errors Task 7.4: Usability testing
Deliverables: Deliverable 7.1: Log files Deliverable 7.2: Bug fixed environment

Table 14: Testing work package

WP 8: Project Demo
Start Date: Nov. 15, 2022 | End Date: Dec. 25, 2022
Leader: Bedirhan Sakinoğlu Members Involved: Gökhan Taş, Bedirhan Sakinoğlu, Lara Fenercioğlu, Zeynep Ziyagil
Objectives: Presenting the end product to the stakeholders.
Tasks: Task 8.1: Presentation preparation Task 8.2: Rehearsal
Deliverables: Deliverable 8.1: Project Demo Deliverable 8.2: Project Presentation

Table 15: Project Demo work package

WP 9: Beta Release
Start Date: Apr. 13, 2023 | End Date: May 10, 2023
Leader: Gökhan Taş Members Involved: Utku Sezer, Bedirhan Sakinoğlu, Lara Fenercioğlu, Zeynep Ziyagil
Objectives: To make the merchandise compatible with multiple mobile platforms
Tasks: Task 9.1: Creating user manual

Task 9.2: Agreements with museums Task 9.3: Testing on-site Task 9.4: Creating mobile application executable Task 9.4: Publishing on mobile application stores
Deliverables: Deliverable 9.1: User manual Deliverable 9.2: Application

Table 16: Beta Release work package

3.4. Ensuring Proper Teamwork

During the development of Mutrivia, we need to keep track of the progress of the development and make a good task distribution among our team. We wanted to make sure that everyone participated in the project equally. It is not easy to see what a group member has done, how tasks are going, and the project's overall status without using some applications. Not using the main application to communicate the project development may lead to information loss and communication problems. Also, we wanted to use a versioning algorithm to make development easier.

- **Jira**: We utilize Jira basically to keep track of tasks. Jira enables us to plan all the necessary tasks and have an overall view of the project. The difficulty of tasks can be determined and assigned to team members. Also, it lets us see the status of each task. In short, it helps us to keep track of the development process.

- **GitHub**: We use GitHub mainly for version control. It helps us add functionality to the project without disrupting others' work, enabling us to work on diverse functionalities of the project as a team. Besides, it lets us see the participation of team members through their commitment.

3.5. Ethics and Professional Responsibilities

Construction of the Mutrivia application requires team members to work in a structured and disciplined manner. In addition to the workload of Mutrivia, each team member has a certainly demanding workload, and they must conduct themselves properly. For the project to be completed on time, every member must satisfy the weekly requirements meticulously.

The data collected from museums will not be shared outside the gamification area.

The application's professional obligations include the resultant code. There is a chance that our team will have to fix numerous bugs toward the project's conclusion. As a professional duty, group members are expected to follow coding practices, comment on the code, and produce clean code to resolve defects quickly after the testing phase.

3.6. Planning for New Knowledge and Learning Strategies

Natural Language Processing will be utilized for generating questions; therefore, team members need to learn about AI and NLP. Some team members are already familiar with these technologies; others use literature reviews and online resources such as videos, online courses, and tutorials to learn about these subjects. In addition, we will use Flutter for mobile applications. Some members are familiar with Flutter, and others will use online resources and hands-on experience to learn it. We are also planning to use the Spring Boot framework for the backend; however, it is a new technology for most of the team. We will learn about this framework through online resources and tutorials and by inspecting similar projects made by Spring Boot.

4. Glossary

- **Natural Language Processing (NLP):** NLP is a branch of artificial intelligence concerned with allowing computers to understand the text and spoken words in the same way humans can [2].
- **Gamification:** Gamification is a method that is a set of activities and processes to solve problems using or applying game elements' characteristics. It improves engagement with the application and is used for different purposes, such as education and entertainment [3].
- **REST API**: Application programming interface (API) is a set of rules defining how applications or devices can connect to and communicate. A REST API is an API that conforms to the design principles of the REST, or representational state transfer architectural style. For this reason, REST APIs are sometimes referred to as RESTful APIs [4].

5. References

[1] A. López-Martínez, Á. Carrera, and C. A. Iglesias, “Empowering museum experiences applying gamification techniques based on Linked Data and smart objects,” *Applied Sciences*, vol. 10, no. 16, p. 5419, 2020.

[2] IBM Cloud Education, “What is natural language processing?,” *IBM*, 02-Jul-2020. [Online]. Available: https://www.ibm.com/cloud/learn/natural-language-processing. [Accessed: 28-Oct-2022].

[3] D. Z. Fitz-Walter, “What is gamification? education, Business & Marketing (2021 examples),” *What is Gamification? Education, Business & Marketing (2021 Examples)*, 2022. [Online]. Available: https://www.gamify.com/what-is-gamification. [Accessed: 12-Nov-2022].

[4] IBM Cloud Education, “Rest-apis,” *IBM*, 06-Apr-2021. [Online]. Available: https://www.ibm.com/cloud/learn/rest-apis. [Accessed: 12-Nov-2022].