Software Requirements Specification

READ THIS FIRST

0. DO NOT START WORKING ON THIS ASSIGNMENT BEFORE READING THE ASSIGNED READING FROM THE TEXTBOOK.

1. Develop the SRS document for your project.

2. When you see words enclosed in "< >" it means that you need to fill this with a proper value, for example, <college name> → Hood College

3. Under each heading there is a paragraph (enclosed in "< >") that provides directions about what should be filled out included under this heading for your project. These directions should not be present when you submit your SRS document, remove them.

4. This is a general template. This means that there might be sections which may not be applicable to the type of software application/system of your project. Do not remove the headings that are not applicable, only indicate that they are N/A. If upon review or iteration the sections are relevant, you will be able to fill them in.

5. Do not ask "what do I put under section X?" This is the point of the assignment, to figure things out and write a comprehensive specification for your project so a software architect can read the requirements and begin designing the software architecture.

6. Treat this as a work deliverable. This means YOU NEVER SUBMIT THE FIRST DRAFT, you spell-check the document and team members review it independently for content inconsistencies along with grammar, syntax, etc.

6. Teams in the past have found it useful to use a shared platform to collaborate on the document and/or track changes. Whatever you do, save the work often.

8. Remove this text-box before submitting your SRS for your project.

for

<Jeopardy style web game - ALEX>

Version 1.0 approved

Prepared by <

Krishna Priya Mula

Anoushka Kondoj

Sai Ram Kondali

Rushikiran Gurijala

Malak Almutairi >

<TEAM E>

<09/27/2023>

Table of Contents

Revision History

Name	Date	Reason For Changes	Version

1. Introduction

1.1 Purpose

This Software Requirements Specification (SRS) document details the requirements for

"Development Codename ALEX", Version 1.0, an online multiplayer Jeopardy-style game. The scope of this SRS encompasses the entire system, detailing the application's intended functionality, features, constraints, and interactions. This document does not focus on any single subsystem but provides a holistic view of the entire application to ensure a shared understanding among stakeholders, developers, testers, and users.

1.2 Document Conventions

To promote readability and uniformity, the following typographic rules are used in this Software.

Requirements Specification (SRS) document:

• **Bold text** is used to draw attention to system features, key phrases, or headlines.

• For clarity, *italicized text* highlights specific words or phrases.

• Monospaced Text: Used to display technical jargon, file names, or little pieces of code.

• Lists or groups of related items are denoted by bullet points.

• "Quotation Marks": These are used to denote user feedback, system feedback, or straight quotations from sources.

Unless otherwise specified, it is anticipated that the detailed needs for each higher-level

requirement would inherit the higher-level requirements' priorities. Without being specifically stated, each requirement will have its own priority, ensuring that the development team is aware of its significance and the sequence in which it should be implemented.

These conventions are used to make the document easier to read and to provide rapid visual cues for significant content. For successful material comprehension, readers are urged to get familiar with certain conventions.

1.3 Intended Audience and Reading Suggestions

The following readers are among those for whom this Software Requirements Specification (SRS) document is intended:

• Developers: Those in charge of planning and creating the application.

• Project Managers: Project managers are in charge of monitoring the project's development and making sure milestones are met.

• Marketing personnel: Experts who want to comprehend the aspects of the product for outreach and promotion.

• Users: End users curious about the capabilities of the program.

• Testers: People in charge of confirming the functionality of the application in relation to the specifications.

Document Structure:

The SRS is designed to present information in a logical order, beginning with a general overview and moving on to specifics:

• **Introduction**: General overview of the document's purpose and conventions

• **Overall description**: Gives a bird's-eye view of the product, its features, and its working environment in its overall description.

• **System Features**: Explores the system's intricate functionalities.

• **External Interface Requirements:** Describes interfaces in terms of user and system interactions.

• **Non-functional Requirements**: Performance, security, and other non-functional requirements are listed here.

• **Other Requirements**: List any other additional considerations or requirements

• **Appendices:** Glossaries, analytical models, and other supporting data

Reading Suggestions

It is advised to start with the "Introduction" and "Overall Description" parts for a comprehensive understanding. After that: The "System Features", "External Interface Requirements", and "Other Nonfunctional Requirements" should be the developers' and testers' main areas of attention. Project managers may want to read the entire document, with "System Features" and "Other Requirements" getting special attention. The terms "Overall Description" and "System Features" will

be useful to marketing staff. The "Overall Description" and "User Interfaces" sections of the "External Interface Requirements" section can provide users with information. To guarantee thorough documentation, documentation writers should go over the entire document. Readers can efficiently understand the product's scope, functions, and requirements by adhering to the recommended order.

1.4 Project Scope

The "Development Codename ALEX" initiative aims to create a Computer Science-focused. online multiplayer Jeopardy-style game. The goal of this game is to be a full platform that offers its customers an educational tool in addition to a competitive quiz atmosphere.

Purpose:

This software's main objective is to build an interactive space where users may test them

understanding of computer science. It is a platform for learning rather than merely a game that combines fun and knowledge. The objective of the game is to encourage players to learn more about Computer Science while having fun with friendly competition.

Benefits:

• Enhances education by giving people a forum to evaluate and increase their understanding of computer science.

• Collaborative learning: Promotes group interaction and a sense of belonging.

• Flexibility: Can be used locally or through a web browser.

• Environment that is interactive: Enables dynamic interactions with game management

and real-time question changes.

Features:

• Authentic Gameplay: The game will give various categories of questions with various

degrees of difficulty, and it will strictly adhere to the regulations of the official Jeopardy Quiz Show.

• Multiple authentication options, including the usual username-password combination and optional facial recognition, for enhanced security.

• Advanced Logging: Extensive recording of each player's action with the option to rerun

the game.

• The ability to create games, send invitations, and control game sessions is shared by all

users.

• Administrator Controls: Administrators have a wide range of controls, including the

ability to batch upload questions using CSV and start and end game sessions.

• Adaptation to Corporate Objectives:

The larger goal of boosting computer science education is aligned with this game. By providing a platform that mixes education and pleasure, it not only raises awareness of the topic but also promotes lifelong learning. The game can be included into lesson plans, used as an extra resource in the classroom or used as a stage for tech competitions and events.

Future Outlook:

The game "Development Codename ALEX" has the ability to change, even though this SRS only covers the game's original release. In later incarnations, new question types, improved game mechanics, or even broader subject areas outside of computer science may be included.

1.5 References

1. Web Content Accessibility Guidelines (WCAG):

• Title: Web Content Accessibility Guidelines (WCAG) 2.1

• Author: W3C Web Accessibility Initiative (WAI)

• Version: 2.1

• Date: June 2018

• Source: https://www.w3.org/TR/WCAG21/

2. General Data Protection Regulation (GDPR):

• Title: Regulation (EU) 2016/679 of the European Parliament and of the Council

• Author: European Union

• Version: OJ L 119, 4.5.2016

• Date: May 2016

• Source: https://eur-lex.europa.eu/legal-content/EN/TXT/?uri=celex%3A32016R0679

3. California Consumer Privacy Act (CCPA):

Title: California Consumer Privacy Act of 2018

• Author: California State Legislature

• Date: June 2018

• Source:

https://leginfo.legislature.ca.gov/faces/codes_displayText.xhtml?division=3.&chapter=20

.&lawCode=CIV&title=1.81.5

4. Children's Online Privacy Protection Rule (COPPA):

• Title: Children's Online Privacy Protection Rule ("COPPA")

• Author: United States Federal Trade Commission

• Date: April 2000

• Source: https://www.ftc.gov/enforcement/rules/rulemaking-regulatory-reform

proceedings/childrens-online-privacy-protection-rule

5. Jeopardy Game Original Design:

• Title: Jeopardy! – An American Television Game Show

• Author: Merv Griffin

• Date: March 1964

• Source: https://www.jeopardy.com/

6. User Interface Design Principles:

• Title: User Interface Design Basics

• Author: US Department of Health & Human Services

• Date: Updated March 2021

• Source: https://www.usability.gov/what-and-why/user-interface-design.html

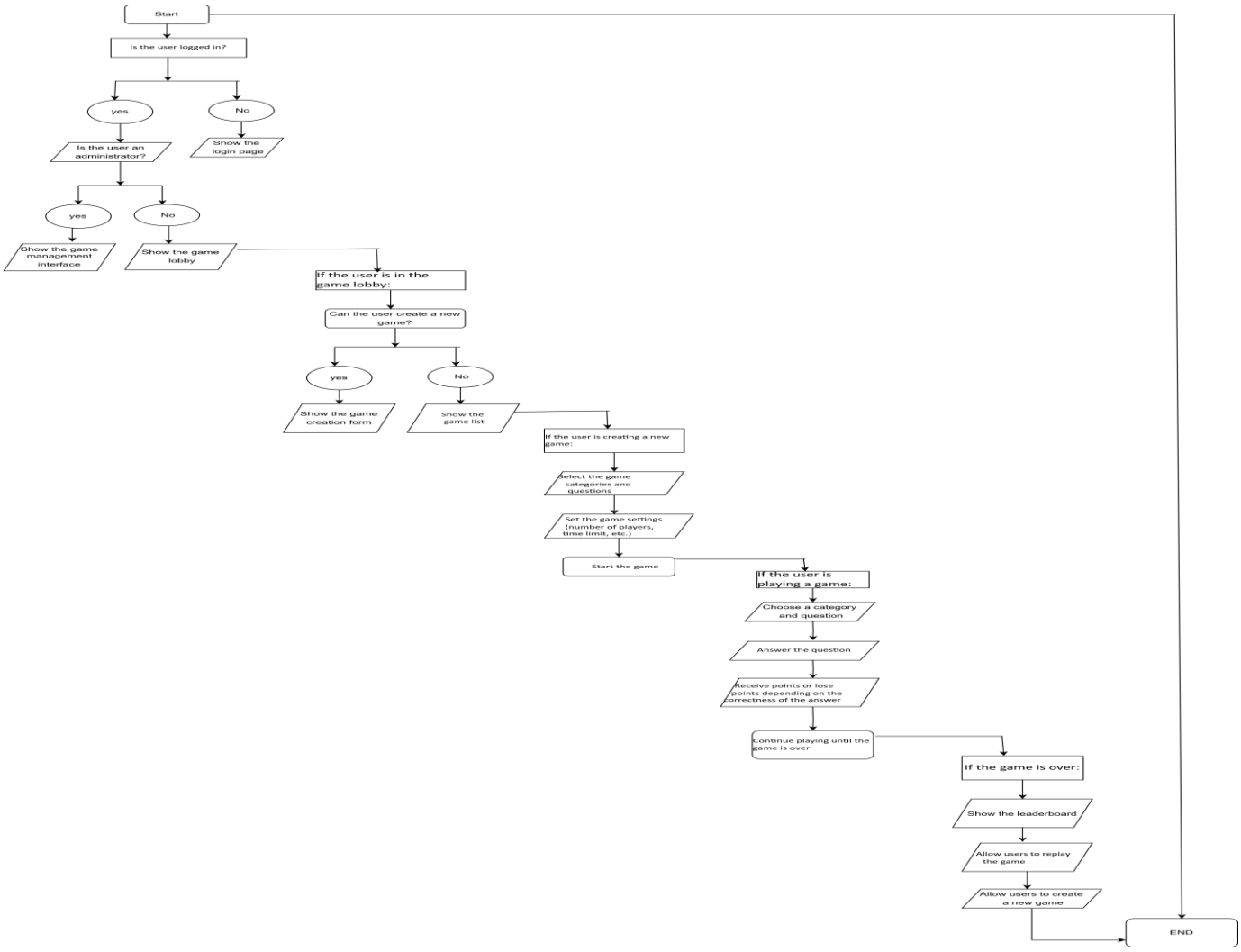
2. Overall Description

2.1 Product Perspective

The Jeopardy software is a stand-alone tool meant to simulate the popular trivia game show "Jeopardy!" for amusement, educational, or training reasons. This software is not a component of a bigger system or a member of a product family. It is, instead, a stand-alone program designed to provide users with an interactive and enjoyable experience reminiscent of a classic television game show. The purpose of this software is to capture the essence of the presentation and let users to enjoy it from the comfort of their own computers or devices. While inspired by the television show, this software is not affiliated with the official "Jeopardy!" brand and is only for amusement and education.

2.2 Product Features

The online, multiplayer CS Jeopardy-style Game web application will allow users to play the game alone or with others. It will have a user-friendly, intuitive, and appealing web-based interface and will completely conform with the official Jeopardy Quiz Show guidelines. It will also provide detailed logging of every player's activity during the game, allow users to retrieve and view a replay of the game, and allow administrators to start and stop game sessions, visualize the state of the users and gameplay, and manually add, edit, and remove categories and questions.



2.3 User Classes and Characteristics

Casual Players:

- **Characteristics**: Casual players are individuals who occasionally play Jeopardy for entertainment and may have varying levels of familiarity with the game. They do not require in-depth knowledge or technical expertise.
- **Frequency of Use**: Infrequent or sporadic usage.
- **Relevant Requirements**: Intuitive user interface, easy navigation, and straightforward gameplay instructions.

Enthusiast Players:

- **Characteristics:** Enthusiast players are aficionados of the "Jeopardy!" game show and may be more familiar with its rules and conventions. They are looking for a more immersive experience.
- **Frequency of Use**: Regular or frequent players.
- **Relevant Requirements:** Advanced gameplay options, customizable game settings, and the ability to simulate the show's format closely.

Competitive Players:

- **Characteristics:** Competitive players engage in Jeopardy competitions with friends or in organized events. They prioritize features that support competitive play.
- **Frequency of Use:** Depends on competition schedules and personal preferences.
- **Relevant Requirements**: Multiplayer modes, leaderboards, and real-time scoring.

Distinguishing User Classes:

The favored user classes for the Jeopardy software depend on the software's primary use case. **For example:**

- For a consumer-focused Jeopardy game app, casual and enthusiast players are favored user classes.
- For an educational Jeopardy software used in schools, educators and students would be prioritized.
- In a corporate training context, administrators and employees in training would be the primary user classes.
- The prioritization of user classes should align with the software's goals and target audience. However, it's essential to ensure that the software provides a positive experience for all user classes and meets their specific needs when applicable.

2.4 Operating Environment

The operating environment for the Jeopardy game software encompasses the hardware and software components necessary for the software to function correctly and provide a seamless user experience. Here are the key aspects of the operating environment:

Hardware Platform:

- The Jeopardy game software is designed to run on standard computing hardware.
- It should be compatible with various hardware configurations, including desktop computers, laptops, and mobile devices (smartphones and tablets).

Operating System (OS):

- The software should be compatible with multiple operating systems to reach a broad user base. Key target operating systems may include:
- Microsoft Windows (e.g., Windows 10 and later)
- macOS (e.g., macOS 10.12 and later)
- Linux distributions (e.g., Ubuntu, Fedora)
- Mobile platforms (e.g., iOS for Apple devices, Android for Android devices)

Web-Based Version:

- If the Jeopardy game is web-based, it should be compatible with a range of web browsers, such as Google Chrome, Mozilla Firefox, Microsoft Edge, Safari, and others.
- Ensure compatibility with both desktop and mobile browsers to accommodate different users' preferences and devices.

Software Dependencies:

- The Jeopardy game software may have specific software dependencies, such as runtime libraries, frameworks, or scripting languages (e.g., JavaScript).
- It should clearly specify any required software components and versions in its installation or setup instructions to ensure seamless installation and execution.

Network Connectivity:

- Depending on the software's features, it may require internet connectivity for various purposes, such as downloading question sets, sharing game results, or enabling multiplayer modes.
- The software should gracefully handle scenarios with limited or no internet access, providing an offline or single-player mode when necessary.

Security Considerations:

- If the software collects user data or involves user accounts, it should adhere to security best practices and may require encryption for data transmission and storage.
- For web-based versions, secure socket layer (SSL) encryption may be necessary for secure communication over the internet.

Screen Resolution and Display:

- The software should be designed to adapt to various screen resolutions and display sizes to ensure an optimal user experience across different devices.

Accessibility:

Consider accessibility standards and guidelines to ensure that the software is usable by individuals with disabilities. This includes features like keyboard navigation, screen reader compatibility, and accessible user interfaces.

By considering these aspects of the operating environment, the Jeopardy game software can be made versatile and accessible to a wide range of users across different platforms and devices while ensuring a smooth and enjoyable gaming experience.

2.5 Design and Implementation Constraints

1. **Intellectual Property and Copyright:**

Developers must adhere to copyright laws and intellectual property rights, ensuring that game content, including questions and answers, does not infringe on existing copyrights.

2. **Hardware Limitations:**

The software should be optimized to run on a wide range of hardware configurations, including devices with limited processing power, memory, and graphics capabilities.

3. **Operating System Compatibility:**

The need to support multiple operating systems and versions can limit the choice of development tools and libraries. Developers must ensure cross-platform compatibility.

4. **Web Browser Compatibility (for Web-Based Versions):**

If the game is web-based, developers must ensure compatibility with various web browsers, which may have differing levels of support for web technologies.

5. **Software Dependencies:**

Depending on the chosen technology stack, developers may be limited by the availability and compatibility of specific libraries, frameworks, or scripting languages.

6. **Regulatory and Compliance Requirements:**

If the game involves financial transactions, user data collection, or other sensitive activities, developers must comply with relevant regulations (e.g., GDPR, HIPAA) and security standards.

7. **Security Considerations:**

Developers must prioritize security by implementing encryption, user authentication, and secure data storage. Security constraints may limit certain development options.

8. **Design and Accessibility Standards:**

Adherence to design conventions and accessibility standards may limit creative choices but is essential to ensure usability and inclusivity.

9. **Database and Content Management:**

Developers may need to work with specific databases or content management systems for storage and management. These systems should align with organizational requirements.

2.6 User Documentation

Providing comprehensive user documentation is essential to ensure that users can effectively use the Jeopardy game software. Here are the user documentation components that will be delivered along with the software:

1. **User Manual:**
- A user manual in digital or printable format that provides detailed instructions on how to install, configure, and use the Jeopardy game software.
- It includes step-by-step guides for various user scenarios, from starting a new game to customizing settings and accessing advanced features.
- The user manual should be organized logically and include a table of contents for easy navigation.

2. **On-Screen Help or Tooltips:**
- Context-sensitive on-screen help and tooltips within the software's user interface to provide immediate assistance to users as they interact with the application.
- These tooltips should explain the purpose and usage of various buttons, icons, and menu options.

3. **Tutorials:**
- Interactive tutorials or walkthroughs within the software to guide users through their first experience with the game.
- Tutorials can help users understand the rules, controls, and objectives of the game through a hands-on learning approach.

4. **FAQ (Frequently Asked Questions):**
- A document or section of the user manual that addresses common user queries and issues.
- The FAQ provides quick answers to questions related to gameplay, technical troubleshooting, and common scenarios.

5. **Troubleshooting Guide:**
- A document that helps users diagnose and resolve common technical issues they may encounter while running the software.
- It should include step-by-step instructions for identifying and resolving problems.

6. **Gameplay Strategies and Tips:**
- Optional documentation that offers gameplay strategies, tips, and suggestions for users who want to improve their performance in the game.
- This can enhance the user's overall experience and engagement with the software.

7. **Accessibility Guide (if applicable):**
- Information on how to use accessibility features, such as screen reader support or keyboard shortcuts, for users with disabilities.

8. **Glossary of Terms:**
- A glossary that defines and explains terms, acronyms, and jargon related to the game and software.

9. **Legal and Privacy Information:**

- Disclosures about user data handling and privacy practices, as well as any legal disclaimers or terms of use.

User Documentation Delivery Formats and Standards:

- The delivery formats and standards for the user documentation may vary depending on the target audience and distribution channels. Common formats and standards include:

PDF Documents: User manuals and guides may be provided in PDF format, which allows for easy printing and offline access.

Online Help: On-screen help and tooltips are integrated into the software's user interface and provide real-time assistance.

Interactive Tutorials: Tutorials within the software can be interactive and include animations or guided simulations.

HTML or Web-Based Documentation: Some documentation components, especially FAQs, may be provided as web-based pages for easy access through a browser.

Plain Text or Markdown: Documentation may also be provided in plain text or Markdown format for simplicity and compatibility.

Compliance with Accessibility Standards: If the software aims to be accessible, user documentation should adhere to accessibility standards like WCAG to ensure that it's usable by individuals with disabilities.

Localization: Consideration for localization to provide user documentation in multiple languages, especially if the software has a global user base.

The choice of formats and standards should align with the software's distribution model and accessibility requirements to ensure that users can easily access and utilize the documentation as needed.

2.7 Assumptions and Dependencies

Availability of Question Data:

- **Assumption**: It is assumed that a reliable source of questions and answers for the Jeopardy game will be available for integration into the software.

- **Impact**: If obtaining question data becomes problematic or costly, it may affect the availability and variety of content in the game.

Third-Party APIs or Services:

- **Assumption**: The SRS assumes that any third-party APIs or services used for features like question updates, multiplayer functionality, or analytics will remain accessible and compatible.
- **Impact**: If third-party services change or become unavailable, it could impact the software's functionality and require adjustments or alternative solutions.

Stability of Target Operating Systems:

- **Assumption:** The SRS assumes that the target operating systems (e.g., Windows, macOS, Android, iOS) will remain stable and compatible with the software.
- **Impact:** Changes in operating system behavior or updates could lead to compatibility issues, necessitating updates to the software.

Web Browser Compatibility:

- **Assumption:** For web-based versions of the game, it is assumed that major web browsers will continue to support web technologies used in the software.
- **Impact**: Changes or discontinuation of support for specific web technologies could affect the software's performance and usability.

Data Privacy Regulations:

- **Assumption**: It is assumed that the software will comply with existing data privacy regulations (e.g., GDPR, CCPA) at the time of development and deployment.
- **Impact**: Changes in privacy regulations may require updates to data handling and user consent mechanisms.

Server Infrastructure (if applicable):

- **Assumption**: If the software relies on server infrastructure for multiplayer modes or content updates, it is assumed that the server infrastructure will remain available and functional.
- **Impact**: Server outages or changes could disrupt online features and gameplay.

External Collaborations:

- **Assumption**: If the development team collaborates with external content creators, partners, or organizations for question sets, it is assumed that these collaborations will proceed as planned.
- **Impact:** Delays or changes in external collaborations could affect the availability of content and features.

User Preferences and Feedback:

- **Assumption:** User preferences and feedback collected during the software's development will align with the assumptions made in the SRS.
- **Impact:** If user preferences and feedback diverge significantly, it may require adjustments to the software's design and features.

3. System Features

3.1 System Feature

3.1.1 Description and Priority

User registration allows users to create accounts on the platform. It has high priority as it is essential for users to access the platform.

3.1.2 Stimulus/Response Sequences

. Users access the registration page.

. User gives information such as email, password, and name.

. The system processes the details of the user.

. System sends confirmation email to user.

.user clicks on the link sent by system.

. system confirms user registration.

.user gets register successful email.

. The system provides facial biometric page.

3.1.3 Functional Requirements

REQ-1: System must provide user friendly interface page with fields such as name, email address,

passwords.

REQ-2: System must validate the email address provided by user is unique and not used address and

password must meet security requirements such as length.

REQ-3: System must securely store user information using hash symbols for password. Upon successful

completion of registration, the system must send verification email to the user.

REQ-4: User should click on the email to confirm the activation of account.

REQ-5: System should send confirmation email on successful activation of account.

REQ-6: User must provide facial pictures for login, so system can store the data and use while user log in

to account.

REQ-6: User must be able to request forgotten password when they forget password.

3.2 User authentication:

3.2.1 Description and priority:

User authentication is the most significant feature that verifies the identity of the user during login into

game. It is a high priority as it ensures security process.

3.2.2 Response sequences/Stimulus:

. User accesses login information page.

. User enters his credentials such as email, username, and password.

. The system checks user credentials if they are valid. The system enters the user into the game.

. Users can also login into the page using face recognition.

. The system should provide facial biometric interface for login into account.

3.2.3 Functional requirements:

REQ-1: The system must provide a login page containing username, password.

REQ-2: The system must validate the provided information is registered in the system.

REQ-3: Before temporary logout system must provide limited number of logins attempts for user.

REQ-4: If the user credentials are valid and matched with details in system, system must grant the

access.

REQ-5: The system should provide facial recognition option, while login to system.

REQ-6: Users should have a secure logout option from the platform.

3.3 Gameplay functioning

3.3.1 Description and priority:

Gameplay functioning means the core rules of the game such as how the game works. This feature has

high priority as it is a major function of the game.

3.3.2 Response sequence/Stimulus:

. Users can start a new game or can join in an existing game.

. The system displays the game board with categories of games.

. Players take turns selecting a category and clues.

. Players respond to clues in the form of questions.

. The system checks the response of players and update scores.

. System displays the final scores and winner on leader board.

3.3.3 Functional requirements:

REQ-1: System must provide leader board with clues, categories by level of difficulty and value of points.

REQ-2: Players must able start a new game or join in existing game.**REQ-3:** System must maintain player scores, questions, timers, and user should take turns while

selecting categories and clues.

REQ-4: System must validate the players answers and deduct the points based on

correcness or incorrect.

REQ-5: System should implement the timer to each question.

REQ-6: System should display scores of players and display winners.

3.4 In game communication:

3.4.1 Description and priority:

The chat in game feature enhances the user experience while playing game and it allows players to

communicate with each other during the game. It has medium priority because it is not essential for all the games.

3.4.2 Response sequence/Stimulus:

. Players can access chat features in game play.

. Text messages can be sent to other players.

. Players can mute others and the system should send time stamps for messages.

. Message feature should support basic features of texting such as bold, italic etc.

3.4.3 Functional requirements:

REQ-1: System must provide chat interface during the game to players.

REQ-2: System must display time stamps to message to indicate the time of chat.

REQ-3: Chat messages should have character limits to prevent spam and fraudulent messages.

REQ-4: The chat interface must be user friendly and clear.

REQ-5: Basic emojis such as emotions like laughter and crying should be supported.

3.5 Game management:

3.5.1 Description and priority:

Game managements deals with administrative control functions needed to organize games in the

system. This feature has medium priority. Because smooth game operation is not as critical as core

gameplay.

3.5.2 Response sequences/Stimulus:

. Only admins can access the game management interface.. Admin can create new game by defining game parameters and settings.

. Admin can pause or resume the game.

. Admin can invite new players to join the game.

3.5.3: Functional requirements:

REQ-1: System must provide an admin user for game management.

REQ-2: Admins should be able to create new games by specifying rules and parameters of game.

REQ-3: Admins must be able to stop or resume or end the game.

REQ-4: The system should provide an option for quit or exit game.

REQ-5: System should support multiple game sessions and handle players capacity.

3.6 Error handling:

3.6.1 Description and priority:

Error handling is an important feature that manages unexpected errors and provides feed-back to users,

and it is highly prioritized as it impacts user experience and system.

3.6.2 Response sequences/Stimulus:

. User selecting Invalid responses to clues results errors.

. System encounters database connection issues or server problems.

. Errors may come during communication between players in multiplayer.

3.6.3: Functional requirements:

REQ-1: System must display invalid message if player provides wrong input.

REQ-2: The system must handle disconnections and time outs gracefully, without any data loss of the

game.

REQ-3: Error recovery methods should be implemented to restore the game to normal state after

technical errors.

REQ-4: A mechanism should be introduced to report cheating in multiplayer games, with consequences.

REQ-5: System must follow security measures in place to protect against cheating and unauthorized access.

REQ-6: Make sure error handling methods and security measures comply with data protection and

privacy regulation.

3.7 Accessibility:

3.7.1 Description and priority:

Accessibility ensures the platform is usable by people with disabilities, making it used for a wide range of

audience. It has high priority to promote the platform/game.

3.7.2 Response sequence/Stimulus:

. Game should be designed to keyboard navigable.

. All elements of game labeled are with names and images.

. All texts should be clear and understandable.

3.7.3 Functional requirements:

REQ-1: Make sure all the portions of game are navigable using screen readers for visual impaired players.

REQ-2: Keyboard navigation should be implemented to allow users to interact with the platform.

REQ-3: System should offer options like audio settings, and visual affects enabling and disabling.

REQ-4: Test should be done with people with disabilities and adjust based on their reviews and feedback.

3.8 Compatibility:

3.8.1 Description and priority:

Compatibility ensures that a game is compatible with various devices and platforms. It has high priority as it directly impacts the reach of the game to users.

3.8.2 Response sequence/Stimulus:

. Users can access games on their comfortable device such as web, mobile.

. System loads and adapts to the device, by providing great experience to users.

3.8.3 Functional requirements:

REQ-1: Game should be accessible on various platforms, including browser and mobile.

REQ-2: It must adapt to various sizes of screens and resolutions to ensure a consistent experience.

REQ-3: User should be able to login with account to access game from different devices.

REQ-4: The development team should stay informed of updates to decrease compatibility issues.

4. External Interface Requirements

4.1 User Interfaces

The application will have two main user interfaces:

- Player interface: This interface will allow users to play the game, either alone or with others. It will include a game board, question and answer displays, and controls for buzzing in and answering questions.
- Administrator interface: This interface will allow administrators to manage the game, including creating and editing categories and questions, starting and stopping game sessions, and viewing game logs.

The user interfaces should be designed to be user-friendly, intuitive, and appealing. They should also be fully accessible to users with disabilities.

Sample screen images:

- Player interface:

    1. Home screen: The home screen should allow users to create a new game, join an existing game, or log in to their account.
    2. Game screen: The game screen should display the current question, answers, and scores for all players. It should also allow players to buzz in and answer questions.

3. Category selection screen: This screen will allow players to select a category of questions to play.
4. Question screen: This screen will display the current question and answer choices.
5. Leaderboard: The leaderboard should display the top players and their scores.
6. Scoreboard screen: This screen will show the current scores of all players.
7. Profile page: The profile page should allow users to view their game history, statistics, and account settings.

- Administrator interface:

- Category management screen: This screen will allow administrators to create, edit, and delete categories of questions.
- Question management screen: This screen will allow administrators to create, edit, and delete questions.
- Game session management screen: This screen will allow administrators to start, stop, and view game sessions.
- Game log screen: This screen will allow administrators to view game logs, including detailed information about each player's activity during the game.

GUI standards and product family style guides:

The user interfaces should be designed in accordance with the following GUI standards and product family style guides:

- Google Material Design
- CS Jeopardy web application style guide (to be developed)

Screen layout constraints:

The user interfaces should be designed to be responsive, meaning that they should look good and function well on devices of all sizes.

Standard buttons and functions:

The following standard buttons and functions should appear on every screen:

- Help: This button should open a help dialog box that provides information about how to use the application.
- Exit: This button should close the application.

Keyboard shortcuts:

The following keyboard shortcuts should be supported:

- Spacebar: Buzz in to answer a question.
- Enter: Submit an answer to a question.

- Tab: Cycle through the answer choices.
- Left/right arrows: Navigate between the different screens.

Error message display standards:

Error messages should be clear, concise, and actionable. They should also be displayed in a prominent location on the screen.

4.2 Hardware Interfaces

The application will be a web application, so it will not require any specific hardware interfaces. However, it should be designed to be compatible with a wide range of devices and browsers.

- Specify the supported device types, such as desktop computers, laptops, tablets, and smartphones.
- Describe the nature of data and control interactions between the software and hardware components, especially for face recognition.
- Mention any specific hardware requirements for face recognition (e.g., webcams or cameras).

4.3 Software Interfaces

The application will use the following software components:

- Database: The application will store its data in a database. The specific database type to be used will be determined during the design phase.
- Web server: The application will be hosted on a web server. The specific web server type to be used will be determined during the design phase.
- Programming language: The application will be developed using a programming language such as Python, Sql, or JavaScript. The specific programming language to be used will be determined during the design phase.

The application will need to communicate with the database and web server in order to store and retrieve data. The specific communication protocols that will be used will be determined during the design phase.

4.4 Communications Interfaces

The application will use the following communications interfaces:

- Web browser: The application will be accessed by users through a web browser.
- Network server: The application will communicate with the database and web server through a network connection.

The application will need to support the following communication protocols:

- HTTP to communicate between the client and server.

- Web sockets to provide real-time communication between players.

The application will also need to implement appropriate security measures to protect user data. This may include using encryption and authentication mechanisms.

The application will also need to use a number of third-party communications services, such as a CAPTCHA service and a face recognition service.

Additional Requirements:

The ALEX web application should also meet the following requirements:

- Security: The program should be safe and prevent unauthorized access to user data.
- Scalability: The application should be scalable to support a large number of users and games simultaneously.
- Maintainability: The application should be well-documented and easy to maintain.

Conclusion:

The ALEX web application is a complex project with a number of challenging requirements. However, by following the requirements outlined in this document, the software engineering team can develop a successful and enjoyable game for users.

5. Other Nonfunctional Requirements

5.1 Performance Requirement

- Response Time: Under normal server load conditions, the application should reply to user activities (e.g., selecting questions, answering, etc.) within 2 seconds on average.
- Scalability: The system should be able to manage at least 1,000 concurrent users without noticeable performance deterioration.
- delay: To maintain a seamless and synchronous gameplay experience, network delay between participants in a multiplayer game should be minimized.
- Database Performance: On average, database searches for retrieving game data, player profiles, and logs should take 1 second.
- Bandwidth Efficiency: Reduce data transfer between the server and client to conserve bandwidth and shorten player loading times.

5.2 Safety Requirements

- Data Integrity: Ensure the integrity of user data and game progress in the event of server crashes or unexpected disruptions to prevent data corruption or loss.

- User Privacy: Take steps to preserve user privacy, such as storing personal information securely, adhering to data protection legislation, and encrypting sensitive data.

- Error Handling: Ensure that robust error handling methods are in place to avoid crashes and mitigate any safety concerns caused by unexpected system failures.

- Backup and Recovery: Backup game data and lo*gs on a regular basis to assist recovery in the event of system failures or data loss occurrences.*

5.3 Security Requirements

- User Authentication: Implement safe user authentication processes such as username and password validation, CAPTCHA functionality, and face recognition-based authentication.

- Access Control: Enforce role-based access control to prevent unauthorized access to administrative operations and sensitive player data.

- Data Encryption: Encrypt all data transmission between the client and server to prevent eavesdropping and data tampering.

- Security Auditing: Implement auditing functionality to track and log all security-related events, such as failed login attempts, access to sensitive data, and illegal actions.

- Compliance: Ensure compliance with appropriate security standards and legislation, such as GDPR or HIPAA, where applicable.

5.4 Software Quality Attributes

- Usability: Prioritize user-friendliness and an intuitive UI to guarantee that gamers can easily navigate the program and enjoy a flawless gaming experience.

- Maintainability: Create the program with clean, well-documented code to make future upgrades, bug repairs, and feature enhancements easier.

- Reliability: Reduce system downtime and crashes to guarantee that users can rely on the program for constant and uninterrupted gameplay.

- 1.4.4 Scalability: Design the program to easily allow future increase in user base and features.

- 1.4.5 Interoperability: Ensure that the program can communicate with multiple web browsers and operating systems.

- 1.4.6 Portability: Optimize the program for cross-platform compatibility, allowing users to access it from a variety of devices and browsers.

- 1.4.7 Testability: Create a robust test suite for the program to discover and address any issues that arise during development and maintenance.

These nonfunctional requirements are critical for producing an online multiplayer CS Jeopardy-style game web application that meets the demands and expectations of both players and administrators. They should direct the development and testing procedures to ensure the project's success.

6. Other Requirements

There are serval other requirements that are required for developing software such as:

6.1 Database Requirements:

. Database management system: Specify the choice of DBMS such as MySQL, MongoDB.

. Specify the data schema including relationships, and attributes for storing the questions of game and

other game data.

. Backup and Recovery methods to make sure data integrity.

6.2 Legal Requirements:

. Address any requirements of license used by platform questions and answers, make sure all content used in the platform compiles with copyright.

. Ensure data protection regulations such as GDPR, CCPA.

6.3 Security Requirements:

. Define how users will be authorized to modify and access the data of game.

. Describe standards of encryption for securing data storage, if any sensitive information is present.

6.4 Internationalization Requirements:

. Multilanguage support: Define if application requires to support multiple languages.

. Ensure international format settings for dates, times, and currency if required.

6.5 Documentation and Training:

. Ensure the creation of help guides and manuals to guide users in using the game.

. Create materials for training for support staff and administrators, if needed.

6.6 Testing and quality assurance:

. Ensure specific testcases to validate functional performance and security of game.

. Set quality metrics such as benchmarks and tolerances levels.**6.7 Deployment and hosting:**

. Define the specific hosting platform, whether it is on premises or cloud based such as AWS,

Azure, Google Cloud.

. Figure out the steps for deploying the application particularly for servers.

6.8 Scalability and performance:

. Describe how the system handles increased load, in terms of multiple games as well as concurrent user.

Appendix A: Glossary

ALEX: The codename for the online, multiplayer CS Jeopardy-style Game web application in development.

CS Jeopardy: A Jeopardy-style trivia game in which teams of CS students compete.

Jeopardy Quiz Show: A popular American television game show in which competitors are given general knowledge hints in the form of answers and must respond with questions.

GUI stands for Graphical User Interface.

CAPTCHA: A form of challenge-response test used to detect whether a user is human or not.

Authentication based on facial recognition: An authentication mechanism that uses facial recognition technology to validate the user's identity.

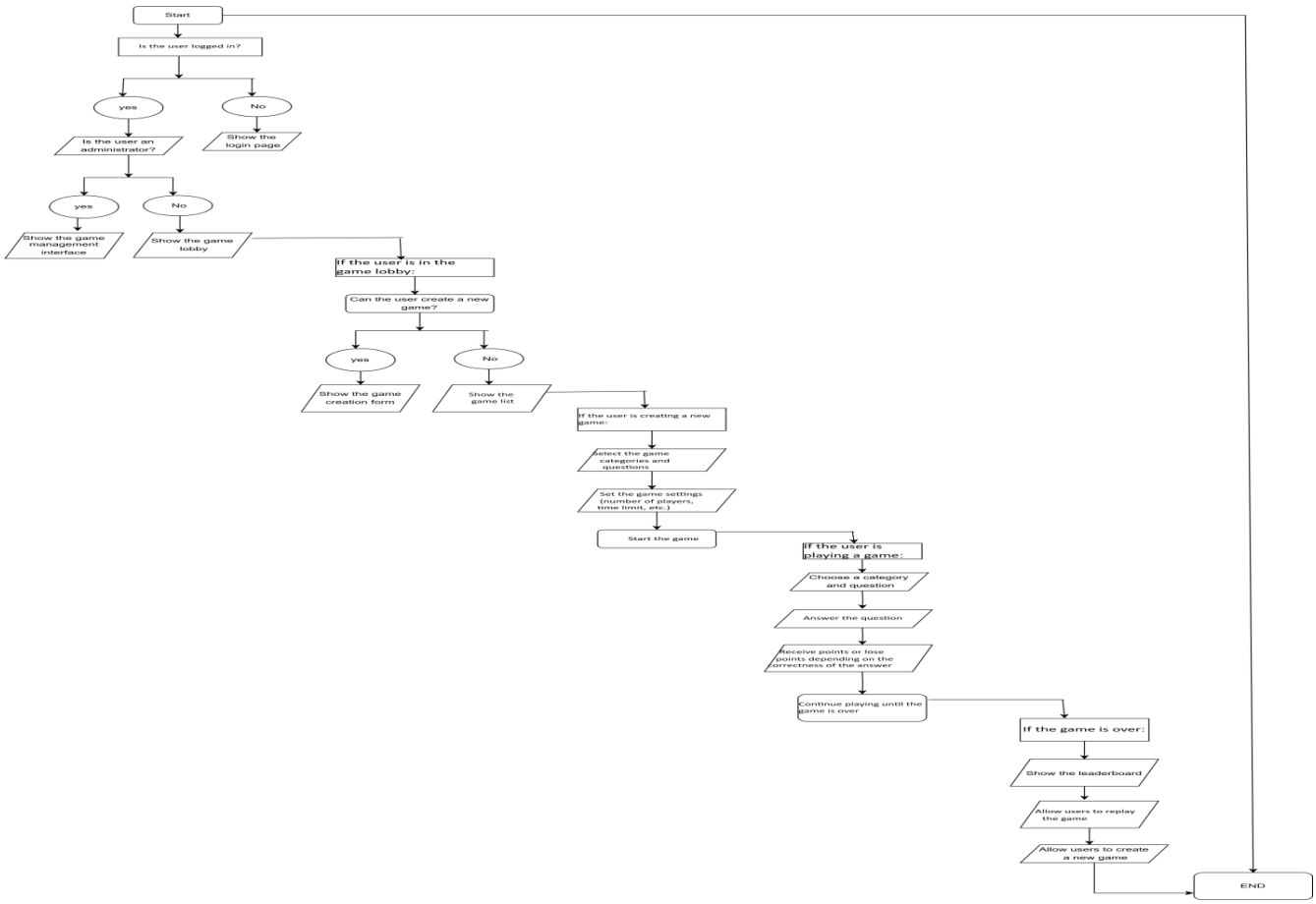
Replay: A game recording that can be viewed afterwards.

Invitation board: A board where users can post game invitations.

Administrators: Users with privileged access to the system.

Appendix B: Analysis Models

Data flow diagram: A diagram that depicts how data moves through a system.



Appendix C: Issues List

Some potential challenges that could emerge during the creation of the ALEX application are as follows:

How will the game's single-player and multiplayer modes be balanced?

How will people with disabilities be able to play the game?

How will the game be protected against cheating?

How will the game scale to accommodate a huge number of users?

These are just a handful of the many concerns that must be addressed throughout the creation of the ALEX application. It is critical to thoroughly examine the requirements and create a comprehensive plan to meet all potential concerns.