

# Requirement Documentation

## Capital Chain - A Word Chain Game Which Teaches the Audience Capitals in North America

### Team Members

Maneet Kaur Chahal

Kareena Sen

Simran Kaur Kullar

Hailey Pong

Victor Yung Wai Tan

## 1. Main Page

### 1.1 Table of Contents

- [Capital Chain - A Word Chain Game Which Teaches the Audience Capitals in North America](#)
  - [Team Members](#)
- [1. Main Page](#)
  - [1.1 Table of Contents](#)
  - [1.2 Tabular Revision History](#)
- [2. Introduction](#)
  - [2.1 Overview](#)
    - [2.1.1 Game Description](#)
    - [2.1.2 Game's Objective](#)
    - [2.1.3 Game's Features](#)
    - [2.1.4 Game's System Requirements](#)
  - [2.2. Objectives](#)
  - [2.2.5 References](#)
- [2.3 Domain Analysis](#)
- [2.4. Functional Requirements](#)
  - [2.4.1 Mandatory Functionalities](#)
  - [2.4.2 Additional Functionalities](#)
  - [2.4.3 Scenario Models](#)
    - [2.4.4 Actors](#)
    - [2.4.5 Use Case Templates](#)
    - [2.4.6 Activity Diagrams](#)
    - [2.4.7 Use Case Diagrams](#)
- [2.5 Non-Functional Requirements](#)
- [2.6. Summary](#)
  - [2.6.1 Table of Terms](#)

### 1.2 Tabular Revision History

Version	Date	Author(s)	Summary of Changes
---------	------	-----------	--------------------

<p>Introduction 1.0</p> <p>Domain Analysis 1.0</p>	<p>Wednesday, January 24th, 2024 - Tuesday, January 30th, 2024</p>	<p>Overview</p> <ul style="list-style-type: none"> <li>• Simran</li> </ul> <p>Objectives</p> <ul style="list-style-type: none"> <li>• Hailey</li> </ul> <p>Domain Analysis</p> <ul style="list-style-type: none"> <li>• Kareena</li> <li>• Simran</li> <li>• Maneet</li> <li>• Hailey</li> <li>• Victor</li> </ul>	<ul style="list-style-type: none"> <li>• Created the project's rough draft</li> <li>• Worked on overview: game's description, objective, features, system requirements</li> <li>• Created first draft of Domain Analysis: Project in specific subdomain and category within this software domain, and instructions component of functional requirements portion</li> </ul>
<p>Introduction 2.0</p> <p>Domain Analysis 2.0</p> <p>Project in a specific subdomain and category within this software domain portion in domain analysis</p> <p>Functional Requirements 1.0</p>	<p>Wednesday, January 31st, 2024 - Tuesday, February 6th, 2024</p>	<p>Functional Requirements Instructions</p> <ul style="list-style-type: none"> <li>• Victor</li> <li>• Hailey</li> <li>• Simran</li> <li>• Maneet</li> </ul> <p>Actors</p> <ul style="list-style-type: none"> <li>• Hailey</li> </ul> <p>Activity Diagrams</p> <ul style="list-style-type: none"> <li>• Simran</li> </ul> <p>Use Case Table</p> <ul style="list-style-type: none"> <li>• Maneet</li> <li>• Kareena</li> </ul> <p>Confluence Transfer</p> <ul style="list-style-type: none"> <li>• Hailey</li> <li>• Simran</li> <li>• Kareena</li> </ul>	<ul style="list-style-type: none"> <li>• Worked on Game Overview <ul style="list-style-type: none"> <li>◦ Objectives</li> <li>◦ Features</li> <li>◦ System requirements</li> </ul> </li> <li>• Functional Requirements: <ul style="list-style-type: none"> <li>◦ Instructions</li> <li>◦ Scenario Models: <ul style="list-style-type: none"> <li>▪ Actors</li> <li>▪ Use Cases Table: <ul style="list-style-type: none"> <li>• Save/Load</li> <li>• High Scores</li> </ul> </li> <li>▪ Activity Diagram: <ul style="list-style-type: none"> <li>• Save/Load</li> </ul> </li> </ul> </li> </ul> </li> <li>• Created Confluence Requirement Documentation page and transferred contents from Google Docs</li> </ul>

Introduction 3.0 Domain Analysis 2.0 Functional Requirements 2.0 Non-Functional Requirements 1.0 Summary 1.0	Wednesday, February 7th, 2024 - Monday, February 12, 2024	<div>Domain<ul style="list-style-type: none"><li>Hailey</li></ul></div> <div>References<ul style="list-style-type: none"><li>Maneet</li></ul></div> <div>Use Case Table<ul style="list-style-type: none"><li>Maneet</li><li>Kareena</li></ul></div> <div>Activity Diagram<ul style="list-style-type: none"><li>Simran</li><li>Maneet</li><li>Kareena</li><li>Victor</li></ul></div> <div>Use case diagrams<ul style="list-style-type: none"><li>Victor</li></ul></div> <div>Non-Functional Requirements<ul style="list-style-type: none"><li>Hailey</li><li>Simran</li><li>Kareena</li></ul></div> <div>Summary<ul style="list-style-type: none"><li>Hailey</li><li>Kareena</li><li>Maneet</li><li>Simran</li></ul></div>	<ul style="list-style-type: none"><li>Introduction<ul style="list-style-type: none"><li>References<ul style="list-style-type: none"><li>Added references for definitions used in Table of Terms<ul style="list-style-type: none"><li>CSV, GUI, JSON, TSV, UI, UX, XML (Maneet)</li></ul></li><li>Added the Software Requirements Document from OWL (Maneet)</li></ul></li><li>Added more description on domain analysis</li><li>Functional Requirements:<ul style="list-style-type: none"><li>Scenario Models:<ul style="list-style-type: none"><li>Use Case Table:<ul style="list-style-type: none"><li>Multiple Players (Maneet)</li><li>Debug/Level selection (Maneet)</li></ul></li><li>Activity Diagram<ul style="list-style-type: none"><li>High scores (Simran)</li><li>Instructions/Tutorials (Simran)</li><li>Debug Mode/level (Simran)</li><li>Multiple Players (Maneet)</li><li>User Chooses Difficulty Level (Kareena)</li><li>User Submits Capital Guess (Victor)</li><li>Advancing To The Next Level (Victor)</li></ul></li></ul></li><li>Added Use case diagram</li></ul></li><li>Non-Functional Requirements:<ul style="list-style-type: none"><li>Rephrased requirements from assignment specification</li></ul></li><li>Summary<ul style="list-style-type: none"><li>Table of Terms</li></ul></li></ul></li></ul>
--	---	---	---

Table 1.0. Tabular Revision History

## 2. Introduction

### 2.1 Overview

#### 2.1.1 Game Description

This educational game aims to enhance your understanding of capital cities in North America, extending beyond your local familiarity and encouraging geographical exploration. By participating in this engaging activity, you'll not only reinforce your knowledge of well-known capitals but also discover and learn about new and possibly unfamiliar ones. The game is designed to expand your understanding of North American capital cities by establishing a chain where the current city you input begins with the last letter of the preceding city. Through interactive gameplay, you can actively connect capitals, promoting a deeper engagement with the diverse geography of North America. It serves as an enjoyable way to expand your geographical knowledge while challenging your memory and encouraging discovery of distant capitals.

#### 2.1.2 Game's Objective

Our goal is to enhance geographical education by transforming what can be a tedious and confusing process into an engaging and accessible experience. The traditional route of memorization is replaced with an interactive and enjoyable approach that leverages a word chain format to connect capital cities. Rather than relying solely on memorization, this game encourages critical thinking as users form connections between the previous and current capital in the word chain. To make the learning process even more enjoyable and educational, the game provides hints and information about each capital, offering insights into their history and culture.

This innovative educational tool not only facilitates the learning of capital cities but also enriches the experience by imparting interesting facts. By creating an interactive journey that combines entertainment with education, users can easily absorb information about the capitals in a way that makes them memorable. Recognizing the importance of geographical learning in our society, the game serves as an effective and fun way to enhance geographical awareness. It blends education with entertainment, offering a fun and unconventional approach to learning.

#### 2.1.3 Game's Features

Our game introduces a fresh take on the classical word chain concept, focusing exclusively on the capitals of North America. The interactive application incorporates several innovative features to enhance the learning experience. Each round, users receive a piece of information connected to a specific capital. Using this information and the last letter of the previous capital, users guess the current capital. For instance, if the hint mentions a capital known for musical entertainment and the last letter of the previous capital is 'n,' the correct guess would be "Nashville."

This application is designed to be versatile, adapting seamlessly to various devices such as phones and laptops. As users successfully guess two capitals, they progress to a higher level, with an increase in difficulty and a shorter timer. Failing to guess within the time prompts a restart with a new capital and information. Users actively engage with the game, inputting their guesses and advancing to the next level, while the application saves their highest score.

The game serves as a dynamic implementation of geographical learning, making the process engaging and interactive. By combining the educational content with the word chain game concept, users not only test their capital knowledge but also form connections with the cultural and historical aspects of each capital. This unique learning culture fosters a deeper understanding of the geographical landscape of North America in an enjoyable and educational manner.

### 2.1.4 Game's System Requirements

- a. Gameplay and Interface
  - i. Core mechanic revolves around linking capitals using the last letter of the previous capital with the first letter of the current capital.
  - ii. Display facts or clues about the capital city for users to guess the correct capital in the chain.
  - iii. User-friendly interface ensuring an engaging and smooth experience for viewing capital information and inputting names.
- b. Key Highlights
  - i. Interesting facts about each capital are integrated into the game, promoting a simultaneous learning experience.
  - ii. Discover unique cultural aspects of each capital, fostering cultural awareness while playing the word chain game.
  - iii. Explore events and historical facts associated with each capital, connecting users with the historical and cultural significance of the capitals.
- c. Users, Application Developers, and Accessibility
  - i. Help section/tutorial providing instructions for users on how to play the game and utilize its features effectively.
  - ii. Developers can regularly update or add new content about capitals to ensure the game stays current and engaging.
- d. Extra Functionality
  - i. Each round features a countdown timer, challenging users to recall and input their guess before the timer expires, testing their knowledge and ability to think under time constraints.
  - ii. Faster correct guesses result in more points for users.
  - iii. Tracking and displaying the highest score achieved by the user.
  - iv. Compatibility with various devices such as phones and laptops.
  - v. Multiplayer feature enabling multiple users to play together and compete.

## 2.2. Objectives

Overall Project Goal:

- Create an immersive and educational word chain game that not only enhances users' knowledge of North American capitals but also offers a satisfying and challenging experience.
- Encourage a positive user experience by combining learning with interactive gameplay, focusing on detailed specifications, incorporating time constraints, and celebrating users' success through a dynamic scoring system.

Project Objectives:

- Learn Capital Cities in North America
- Interpreting and Following Detailed Specifications
- Working Under Time Constraints
- Achievement through a Score Table

## 2.2.5 References

- Computer Hope. (2023, November 12). What is a GUI (graphical user interface)?. Computer Hope. <https://www.computerhope.com/jargon/g/gui.htm>
- Feedink.com. (n.d.). Feedlink. Feedink. <https://feedink.com/en/plik-tsv/>
- Introducing json. JSON. (n.d.). <https://www.json.org/json-en.html>
- Servos, D. (2024). CS2212 Group Project Specification - Version 1.0 - Winter Session 2024. OWL. <https://owl.uwo.ca/access/content/group/aa2311c9-4cef-497f-8d9c-b502023be21c/project/CS2212%20Group%20Project%20Specification.pdf>
- W3Schools. (n.d.). Introduction to XML. XML Introduction. [https://www.w3schools.com/xml/xml\\_what\\_is.asp](https://www.w3schools.com/xml/xml_what_is.asp)

## 2.3 Domain Analysis

The project involves creating an interactive educational application focused on learning North American state capitals through a word chain game format. Users input capital names sequentially, starting with the last letter of the previous one, within a given time frame. The application serves as a capital database, enhancing users' knowledge and easing navigation through various capital cities. Challenges include accurate user input (correct answers and correct spelling), avoiding repetition, time constraints, and creating an educational interface, especially for elementary school students. Proactively addressing these challenges during development will result in a more engaging and user-friendly learning experience.

The target audience includes anyone interested in learning about capital cities, with a particular emphasis on younger age groups, especially elementary school students. The game aims to be a valuable tool for educational institutions, specifically elementary schools, providing an engaging and educational platform for teaching capital cities. Falling under the educational software domain, the project integrates the word chain concept with geographical education, focusing on North American capitals. This game not only captivates users with the word chain but also imparts knowledge about capital cities. The system interprets user-inputted words or capital cities, capturing the last letter of the entered capital to smoothly advance the word chain. The domain dynamically progresses with the user's advancements, ensuring an effective and tailored learning experience.

As for common issues in this domain, ensuring Data Accuracy and Completeness is critical for providing reliable and accurate information about U.S. state capitals. User Engagement and Retention are challenges that will be addressed through the implementation of motivational elements like points, leaderboards, and rewards. Adjusting the difficulty level based on user performance ensures that the learning experience remains challenging yet attainable. Some additional features our game aims to conquer are supporting an offline mode for users, broadening our educational game to children from diverse backgrounds (including English and French language modes), and progress tracking by including the user's number of correct answers and the time taken to complete each level. Our domain understanding provides a strategic foundation for developing an educational game that aligns seamlessly with elementary education goals. This clarity acts as a guideline, helping coordinate efforts and allocate resources effectively. With a well-defined goal, we can plan the project, breaking it into manageable tasks and milestones, and improving development efficiency.

## 2.4. Functional Requirements

### 2.4.1 Mandatory Functionalities

- 1. **User Interface:**
  - Partially mouse-based GUI for easy navigation and user input.
  - Responsive interface conveying the current game state and responding to user inputs.
  - Keyboard shortcuts for common actions, enhancing accessibility (e.g., "esc" for the menu).
  - Visual and auditory feedback for user actions.
- 2. **Main Menu:**
  - Display game title, graphics, developers, team number, and creation term information.
  - Options: Start a new game, load a previously saved game, access instructions/tutorials, view high scores, or exit.
- 3. **Instructions/Tutorials:**
  - Text-based with visual representations for easy understanding.
  - Accessible through the main menu.
  - Tailored for the target audience, including educational institutions and those interested in learning about capital cities.
- 4. **Game Play:**
  - Save/Load Game State: Allows users to save and load game progress.
  - Progression Mechanics: Players advance through levels or stages based on educational objectives or scores.
  - Responds to Player Inputs: Executes in-game actions based on user selections and interactions.
- 5. **High Score Table:**
  - Automatically updates with player scores.
  - Displays player names or initials and corresponding scores.
  - Maintains a leaderboard, adjusting rankings as new high scores are achieved.
- 6. **Multiple Players:**
  - Supports multiple players, recording their names, initials, usernames, or emails.
  - Allow collaboration and competition among users.
- 7. **Instructor Dashboard:**
  - Provides a mode for instructors to access and display metrics and datasets.
- 8. **Debug Mode/Level Selection:**
  - Allows for level selection with a password or pattern for debugging.
  - Not visible to normal players.
- 9. **Housekeeping/Error Handling:**
  - Adapts UI elements to different scales.
  - Implements error handling mechanisms for a smooth user experience.

### 2.4.2 Additional Functionalities

- 1. **Time Limit:** The system must incorporate a time limit set at N seconds for users to solve problems.
- 2. **Point Deduction:** Incorrect user inputs should result in points being deducted from the overall score.
- 3. **Skip Option:** Users can choose to skip a question if they do not know the answer, but this action will lead to point deduction.
- 4. **Star Reward:** Stars are earned when users reach a specific point threshold, serving as a reward mechanism for achieving a certain level of performance.

### 2.4.3 Scenario Models

#### 2.4.4 Actors

There are three types of actors in this project:

- a. User
- b. Instructor
- c. Administrator

Description	Users enter data (eg. login) or respond to given questions to fulfill achievement requirements.
Aliases	Player(s)
Inherits	None
Actor Type	Person

<b>Active/Passive</b>	Active
-----------------------	--------

Table 2.0 Actor User

<b>Description</b>	<b>Instructor retrieves user data.</b>
<b>Aliases</b>	Educator, teacher
<b>Inherits</b>	User
<b>Actor Type</b>	Person
<b>Active/Passive</b>	Passive

Table 3.0 Actor Instructor

<b>Description</b>	<b>Administrator actively perform tasks such as managing users, configuring system settings, and overseeing the operation for the software.</b>
<b>Aliases</b>	Programmer, developer
<b>Inherits</b>	None
<b>Actor Type</b>	Person
<b>Active /Passive</b>	Active

Table 4.0 Actor Administrator

## 2.4.5 Use Case Templates

<b>Use Case:</b>	<b>Save/Load Game</b>
<b>Primary Actor:</b>	User
<b>Secondary Actor:</b>	None
<b>Goal in Context:</b>	To save the current game or load previously saved progress.
<b>Preconditions:</b>	The user is logged in to be able to save.
<b>Trigger:</b>	<ul style="list-style-type: none"> <li>◦ When the game is launched the user can load previous progress.</li> <li>◦ When a stage is complete, the user has the option to save progress.</li> </ul>
<b>Scenario:</b>	<ul style="list-style-type: none"> <li>a. The user inputs their username and password to log in.</li> <li>b. The user has an option to load previously saved games.</li> <li>c. If the previous save is loaded, the user will be able to continue working on stages from where they last saved.</li> <li>d. Once the stage is complete, the user can choose to save and continue or save and exit.</li> </ul>
<b>Alternatives:</b>	<ul style="list-style-type: none"> <li>a. If the user is new, then they are asked to create a username &amp; password (account) first.</li> <li>b. They will then be prompted with the same layout as previously described.</li> </ul>
<b>Exceptions:</b>	<ul style="list-style-type: none"> <li>a. An account does not exist with the username and/or password provided</li> <li>b. There is no game to load if there is no previously saved progress</li> <li>c. User is given 2 tries to log into account and then account will be locked</li> </ul>
<b>Priority:</b>	Essential, must be implemented

Table 5.0 Save/Load Game Use case

<b>Use Case:</b>	<b>High Scores</b>
<b>Primary Actor:</b>	User
<b>Secondary Actor:</b>	Instructor, Admin
<b>Goal in Context:</b>	To be able to add high scores and view a listing/table of highest scores in a certain top percentage.
<b>Preconditions:</b>	<ul style="list-style-type: none"> <li>◦ The user has an account to save the score</li> </ul>
<b>Trigger:</b>	The user's score surpasses the minimum score to be on leaderboards
<b>Scenario:</b>	<ul style="list-style-type: none"> <li>a. The user finishes a stage.</li> <li>b. They achieve a score which surpasses at least the lowest score on the leaderboard.</li> <li>c. The user will be alerted that they have made leaderboards.</li> <li>d. User selects the leaderboard button/option and can see a list of all users in the top percent and where they stand on the leaderboard.</li> </ul>
<b>Alternatives:</b>	<ul style="list-style-type: none"> <li>a. The user can also view high scores from main menu without being on the leaderboards</li> </ul>
<b>Exceptions:</b>	<ul style="list-style-type: none"> <li>a. If the user is not logged in they cannot view where they stand on the leaderboards and will have to find their name on their own</li> <li>b. If there is an error calculating score it could not appear on the leaderboards</li> <li>c. User must make account in order to record score so that later on the score is visible</li> </ul>
<b>Priority:</b>	Medium, however it is required.

Table 6.0 High Scores Use Case

<b>Use Case:</b>	<b>Multiple Players</b>
<b>Primary Actor:</b>	Admin/System
<b>Secondary Actor:</b>	Instructors
<b>Goal in Context:</b>	To allow the game to support multiple players/users with data stored
<b>Preconditions:</b>	<ul style="list-style-type: none"> <li>◦ The user must have an account</li> </ul>
<b>Trigger:</b>	An account is created
<b>Scenario:</b>	<ul style="list-style-type: none"> <li>a. A user creates an account</li> <li>b. The information on their account login is stored</li> <li>c. The user can create another account and will have no problem logging in and out</li> </ul>
<b>Alternatives:</b>	<ul style="list-style-type: none"> <li>a. The user's game data will also be stored</li> </ul>
<b>Exceptions:</b>	<ul style="list-style-type: none"> <li>a. User is not logged in</li> <li>b. The data between users gets mistranslated</li> </ul>
<b>Priority:</b>	Medium, however it is required

Table 7.0 Multiple Players Use Case

<b>Use Case:</b>	<b>Debug Mode/Level Selection</b>
<b>Primary Actor:</b>	Admin
<b>Secondary Actor:</b>	None
<b>Goal in Context:</b>	To provide access which allows for level selection if password/pattern is entered for debugging/testing
<b>Preconditions:</b>	None
<b>Trigger:</b>	The admin enters the password/pattern to view the debug mode/level selection
<b>Scenario:</b>	<ul style="list-style-type: none"> <li>a. The admin logs in/enters the password for debug mode/level selection</li> <li>b. They are provided with access to all levels without any progress needed</li> <li>c. Any level can be played by the admin to test the system</li> </ul>
<b>Alternatives:</b>	<ul style="list-style-type: none"> <li>a. The admin is also able to view a list of all answers for game</li> </ul>
<b>Exceptions:</b>	<ul style="list-style-type: none"> <li>a. User is not logged in and cannot view friends' scores</li> </ul>
<b>Priority:</b>	Medium, however it is required

Table 8.0 Debug Mode/Level Selection Use Case

<b>Use Case:</b>	<b>Instructions/Tutorial</b>
<b>Primary Actor:</b>	User
<b>Secondary Actor:</b>	Instructor, Admin
<b>Goal in Context:</b>	To allow users to view instructions/tutorials on how to play the game
<b>Preconditions:</b>	None
<b>Trigger:</b>	The user presses the menu option button to direct themselves to the instructions page.
<b>Scenario:</b>	<ul style="list-style-type: none"> <li>a. The user loads the application.</li> <li>b. The user selects the instructions button the main menu.</li> <li>c. The user is able to view written instructions on how to play as well as some images for more clarification.</li> <li>d. The user can then press the play button which will be shown at the end of the instructions page (as well as the main menu page).</li> </ul>
<b>Alternatives:</b>	<ul style="list-style-type: none"> <li>a. If the user wishes to return to main page they can also select the back button.</li> </ul>
<b>Exceptions:</b>	<ul style="list-style-type: none"> <li>a. There is an error loading the page or images/content on the page.</li> </ul>
<b>Priority:</b>	Medium, however it is required.

Table 9.0 Instructions/Tutorial Use Case

<b>Use Case:</b>	<b>Chooses Difficulty Level</b>
<b>Primary Actor:</b>	User
<b>Secondary Actor:</b>	System



<b>Goal in Context:</b>	The player can choose an appropriate difficulty level according to their preference.
<b>Preconditions:</b>	<ul style="list-style-type: none"> <li>◦ The game has launched and is ready to accept user input</li> <li>◦ The user is on the display screen that enables the setup to choose a certain difficulty level</li> </ul>
<b>Trigger:</b>	The user decides to begin a new game or fix the game settings.
<b>Scenario:</b>	<ul style="list-style-type: none"> <li>a. The user accesses the setup display screen.</li> <li>b. The user selects the button to modify the difficulty level.</li> <li>c. The system displays varying difficulty levels (Easy, Medium, Hard).</li> <li>d. The user can select the difficulty level from the given list.</li> <li>e. The system acknowledges the user's choice and updates the game settings.</li> </ul>
<b>Alternatives:</b>	<ul style="list-style-type: none"> <li>a. If the user does not want to change the difficulty level, they can leave the display screen without making any game changes.</li> </ul>
<b>Exceptions:</b>	<ul style="list-style-type: none"> <li>a. The system undergoes an error when updating the difficulty level.</li> <li>b. The user input is not valid.</li> </ul>
<b>Priority:</b>	Medium, however, is required.

Table 10.0 Choose Difficulty Level Use Case

<b>Use Case:</b>	<b>User Submits Capital Guess</b>
<b>Primary Actor:</b>	User
<b>Secondary Actor:</b>	System
<b>Goal in Context:</b>	The player enters the capital name to start the game or based on the previous capital name entry.
<b>Preconditions:</b>	<ul style="list-style-type: none"> <li>◦ The game is active and displays the page to begin entering a North American capital name or the previously inserted capital name.</li> <li>◦ The user understands the rules of the game, knowing that each capital input after the previous must begin with the last letter of the previous capital city.</li> </ul>
<b>Trigger:</b>	The user decides to submit their guess for the capital city.
<b>Scenario:</b>	<ul style="list-style-type: none"> <li>a. The user enters the display page for the game based on their chosen difficulty.</li> <li>b. The user begins the game by entering a North American capital.</li> <li>c. The user thinks about the next possible entry based on the previous capital entry.</li> <li>d. The user enters their guess in the system.</li> <li>e. The system evaluates the user's entered guess.</li> <li>f. If the guess is correct, the system acknowledges the correct answer and proceeds to let the user enter another capital starting with the last letter of their previous entry. The user can keep entering capitals until either: <ul style="list-style-type: none"> <li>i. The user gets an entry incorrect (the system saves the score)</li> <li>ii. The time limit runs out and terminates the game (the system saves the score)</li> </ul> </li> <li>g. If the user is incorrect, the system terminates and saves the user's score.</li> </ul>
<b>Alternatives:</b>	<ul style="list-style-type: none"> <li>a. The user does not decide to submit a guess and exits the capital chain game.</li> </ul>
<b>Exceptions:</b>	<ul style="list-style-type: none"> <li>a. The user submits an invalid input.</li> <li>b. The system undergoes a systematic error while evaluating the user's guess.</li> </ul>
<b>Priority:</b>	High

Table 11.0 User Submits Capital Guess Use Case

Use Case:	Advancing to the Next Level
Primary Actor:	User
Secondary Actor:	System
Goal in Context:	The player advances to the next difficulty level (for example: easy to medium mode) after accomplishing the previous level.
Preconditions:	<ul style="list-style-type: none"> <li>○ The game is active and has successfully cleared the previous level.</li> <li>○ The user has accumulated the appropriate score to progress to the next game level.</li> <li>○ The design of the game enables multiple game levels.</li> </ul>
Trigger:	The user decides to submit their guess for the capital city.
Scenario:	<ol style="list-style-type: none"> <li>The user is in an active game and has successfully cleared the previous level based on the acquired score.</li> <li>The system evaluates the user's performance and determines whether the user has met the requirements for advancing to the next level.</li> <li>The system displays a message to the user, notifying them that they have cleared the level, and can unlock the next level.</li> <li>The user acknowledges the message and can choose whether or not to proceed to the next level.</li> <li>The system updates the game state to transition to the next desired level.</li> <li>The system presents the user with the new level's challenges, including the shorter time limit and needed score to advance.</li> <li>The user can view and exit the message to continue to the new level of the game.</li> </ol>
Alternatives:	<ol style="list-style-type: none"> <li>The user does not decide to advance to a new level, so they can opt to replay the current level or exit the game completely.</li> <li>If the user has not met the needed requirements to proceed to the next level, the system redirects them to the current level and enables the user to reattempt the level.</li> </ol>
Exceptions:	<ol style="list-style-type: none"> <li>The user experiences any technical issues that prevent them from proceeding.</li> <li>The system undergoes a systematic error when proceeding to the next level.</li> <li>The user accidentally declines the option to advance to the next level.</li> <li>The user tries to manually advance to the next level without meeting any given requirements.</li> </ol>
Priority:	Medium

Table 12.0 Advancing to the Next Level

## 2.4.6 Activity Diagrams

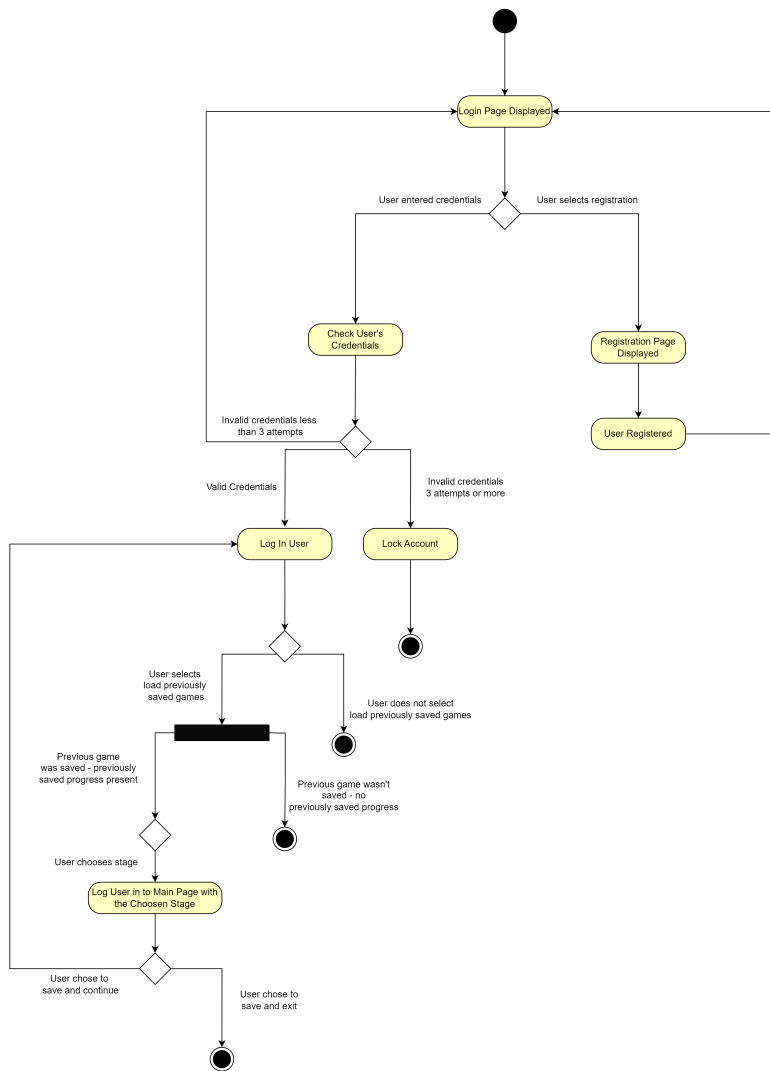


Figure 1.0 Save/load Game Activity Diagram

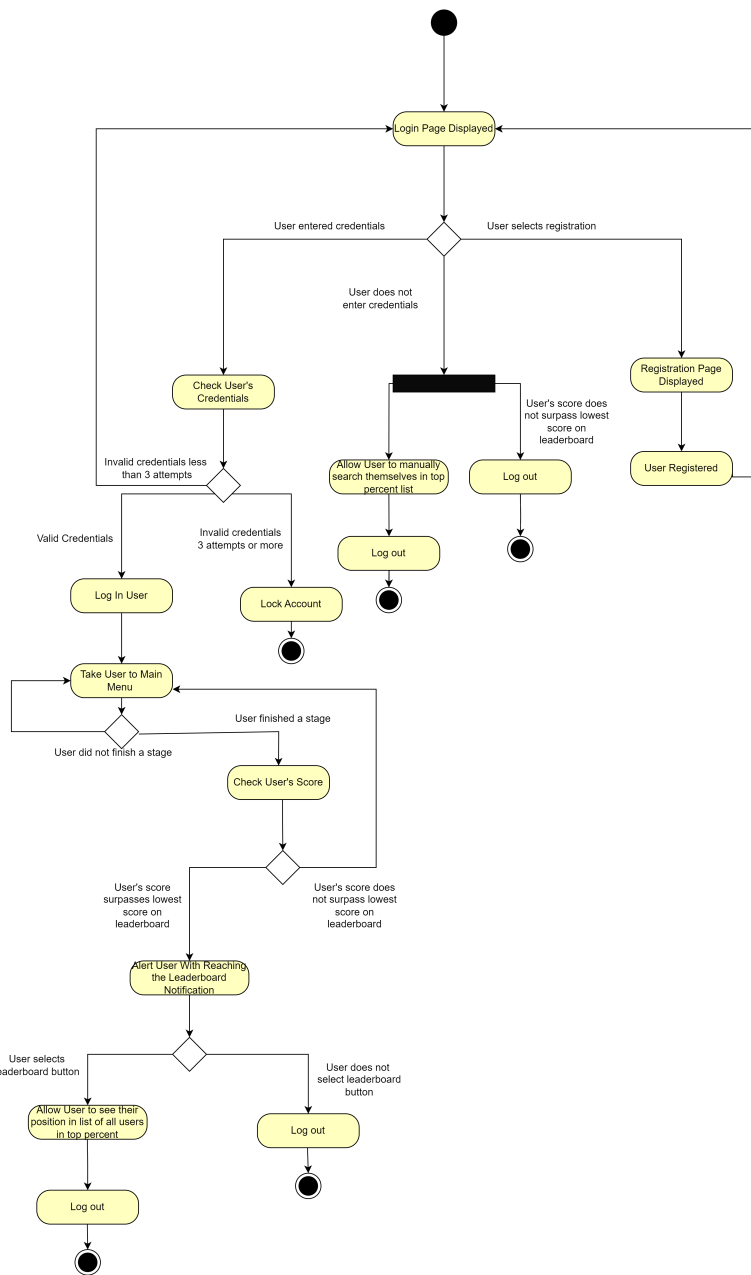


Figure 2.0 High Scores Activity Diagram

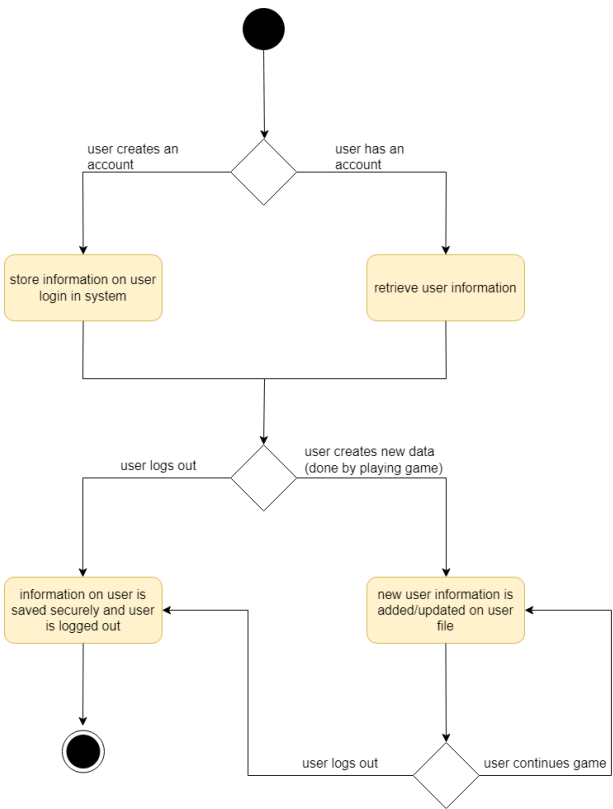


Figure 3.0 Multiple Players/Multiplayer Activity Diagram

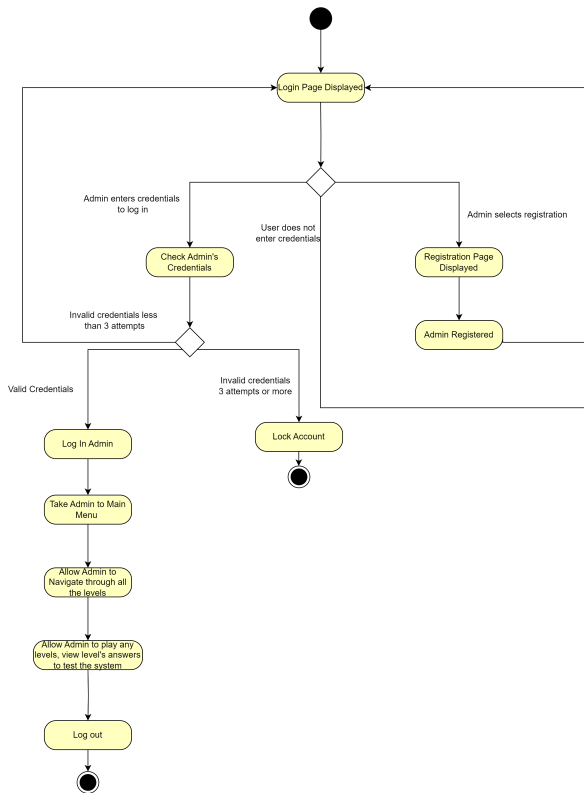


Figure 4.0 Debug Mode/ Level Selection Activity Diagram

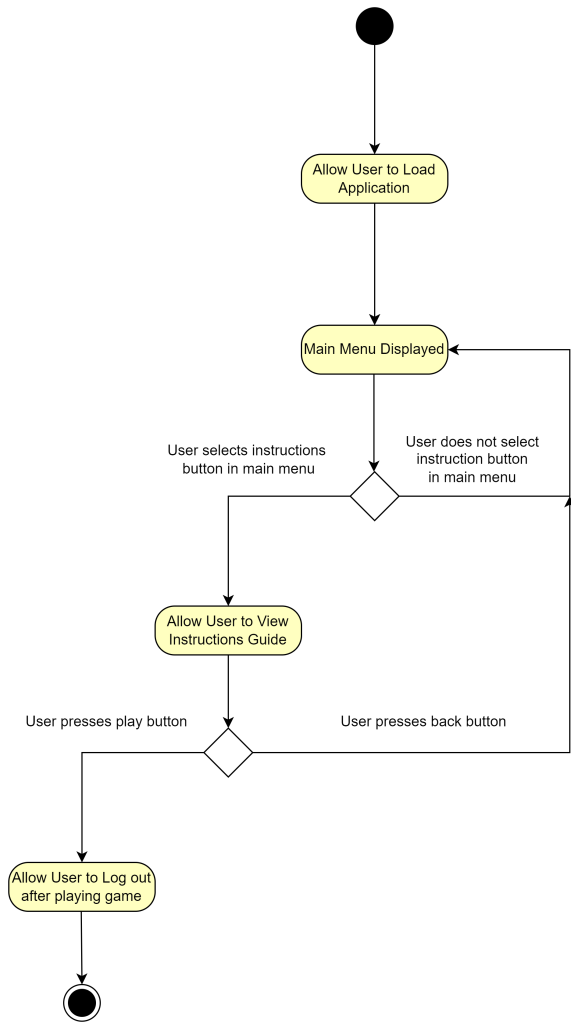


Figure 5.0 Instructions/ Tutorials

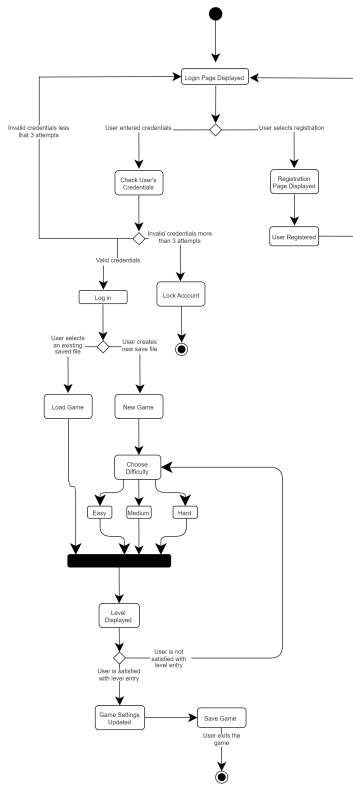


Figure 6.0 User Chooses Difficulty Level Activity Diagram

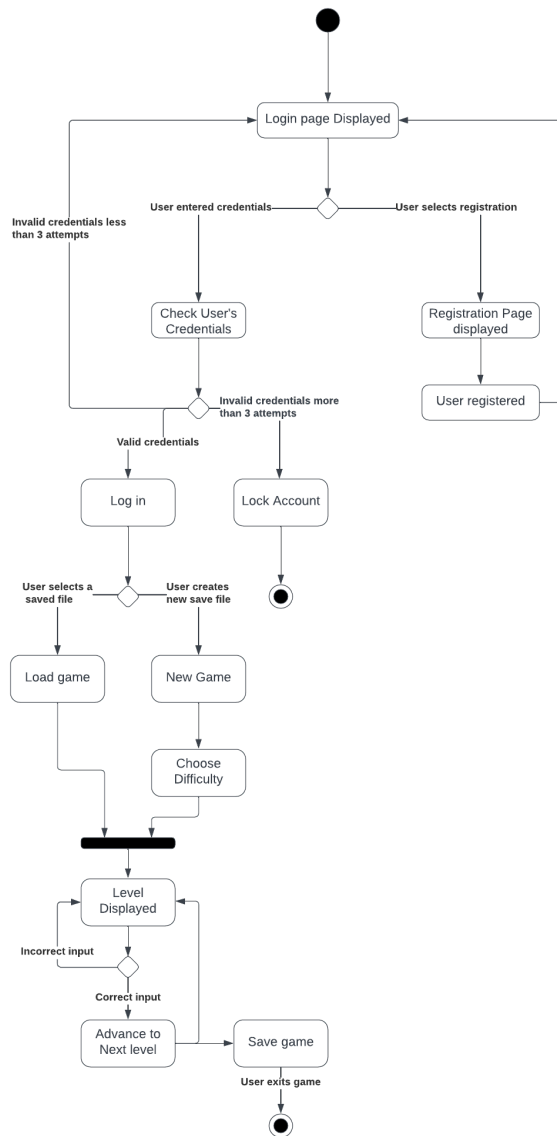


Figure 7.0 User Submits Capital Guess and advances to the next level

## 2.4.7 Use Case Diagrams



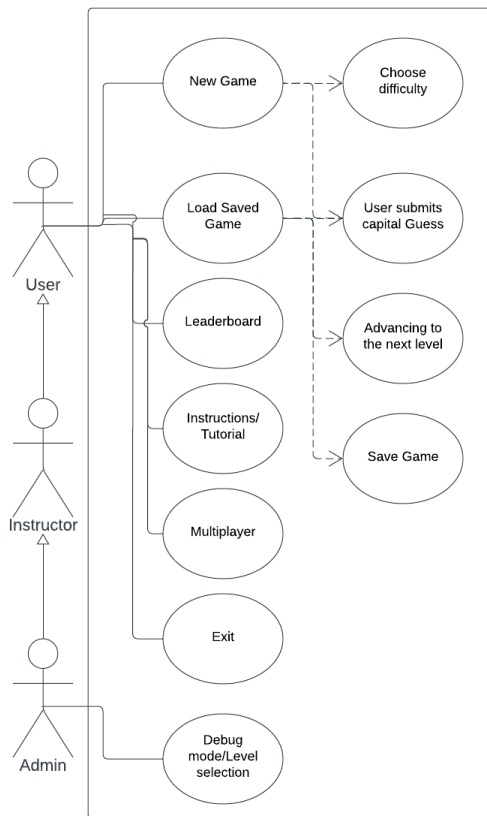


Figure 8.0 Use Case Diagram of Interactions

## 2.5 Non-Functional Requirements

1. Developed in Java 19 or newer; Python allowed with team agreement.
2. Utilize an object-oriented approach and incorporate design patterns.
3. Educational content must be accurate, relevant, up to date, and engaging for the target age group.
4. Include a Graphical User Interface (GUI) following UX best practices.
5. Store all data locally using formats like JSON, XML, CSV, or TSV; avoid internet dependency.
6. Avoid third-party libraries that are not freely available; provide clear instructions for running the application.
7. Use Bitbucket Git repository for code storage, Confluence for design work, and Jira for task tracking.
8. Comment code using Javadoc (or equivalent for Python); cite sources for non-team code.
9. Code must be well formatted with proper indentation and name calling
10. Unit test code with JUnit 5 (or equivalent for Python); GUI-only actions exempt.
11. Follow agreed-upon coding conventions and styles consistently.
12. Execute on Windows 10 with standard Java installation; ensure all team members can compile and run.
13. Self-contained application; no unauthorized file modification outside the installation directory.
14. Visible response to every user action; provide useful error messages.
15. Keep the project file size under 1 gigabyte.
16. Ensure efficient use of computing resources; maintain responsive interface during gameplay.
17. Consider accessibility in UI design, allowing tasks with both keyboard and mouse; logical tab order.
18. Code should be maintainable and reusable.
19. Use English for all content, documentation, and communications among team members.
20. Adhere to sound software engineering principles in the application design.
21. User experience (UX) should be optimized by gathering any feedback, and continuously improving user interfaces for better useability, intuitiveness, and overall user satisfaction.
22. Improve and ensure game security by checking that the user data is securely stored and encrypted to prevent any unauthorized access.
23. Clean, organized user interface.

## 2.6. Summary

Our project, Capital Chain, will educate the users on the geology of North American cities as a fun and engaging game. Instructors will be able to see their student's progress and view where their students stand. Students/users will be able to compete against each other to get on leaderboards. The score will be presented to engage the users and give a sense of achievement. This document provides our objectives and goals, and a blueprint for our project to help us guide through the process. The document will be referred to throughout the entirety of the project. Stakeholders involved will also be able to get an excellent sense of the process of the project.

## 2.6.1 Table of Terms

Terms	Definitions
CSV	Short for Comma Separated Values, which is a type of file that is used to store plain text that is separated by commas.
GUI	Graphical User Interface, an interface that includes images, icons, and other types of graphics that users can interact with for information.
JSON	A text format/file is used to store temporary data.
TSV	Tab Separated Values, records are stored in columns and rows, which are separated by the tab character.
UI	User Interface
UX	User Experience
XML	Extensible Markup Language is a software and hardware tool used to store and transport data
Actor	A role that the user (or external system) plays in the use case
Activity Diagram	<ul style="list-style-type: none"><li>• Provides a graphical representation of the flow of interaction.</li><li>• Represent how a system reacts to internal events.</li></ul>
Use case	Describe typical interactions between the users of a system and the system it's self

Table 13.0 Table of Terms