Software Requirements Specification

for

Water Droplet Game

Version 1.0 approved

Prepared by Katie, Terry & Spencer

University of Utah

April 17, 2023

Table of Contents

Revision History

Name	Date	Reason For Changes	Version

1. Introduction

1.1 Purpose

This SRS covers the development of an application to play the “water droplet catching game”. We will be using the QT framework to facilitate user interaction and gameplay. The initial release will focus on two main things:

1. User Authentication System
2. Game Development (Water Droplet Catching Game)

1.2 Document Conventions

This SRS document for the water droplet game project follows simple conventions for clarity:

1. Font Styles:
    - Bold Text: Used for section headings and key terms.
    - Italicized Text: Used for emphasis.
    - Regular Text: Standard content.
2. Priority Levels:
    - Each requirement statement is labeled with its priority level (P1, P2, P3) to indicate its importance:
        - P1: Critical for basic gameplay functionality.
        - P2: Enhances user experience or provides additional features.
        - P3: Nice-to-have for future iterations.
3. Formatting Conventions:
    - Functional and non-functional requirements are presented in a structured format.
    - UML diagrams are included for clarity.

These conventions ensure that stakeholders can easily navigate and understand the requirements outlined in this document for the water droplet game project.

1.3 Intended Audience and Reading Suggestions

This SRS document is tailored for students in a software engineering class who are collaboratively developing a water droplet game as part of a learning project.

Reading Suggestions:

To make the most of this SRS document, readers are suggested to focus on the following sections:

1. Project Overview **(Introduction):** Provides a high-level summary of the game project, including its scope and objectives.
2. Functional Requirements **(Section 2)**: Details the specific functionalities and behaviors expected from the game.
3. Design Specifications **(Section 3)**: Provides insights into the internal structure and organization of the game, including diagrams and architectural considerations.
4. Testing Guidelines **(Section 4):** Outlines the testing methodologies and procedures to ensure

the game's quality and stability.
5. Appendices: Additional resources and references may be provided in the appendices for further clarification or context. Be sure to consult these supplementary materials for any additional information relevant to the project.

1.4 Product Scope

The development of a water droplet catching game within the Qt gaming framework, along with a user login authentication system. The game will challenge players to control a bucket to collect falling water droplets while avoiding missed droplets. The game will feature multiple difficulty levels, a scoring system, and sound effects.

Sprint 1: User Authentication System

Sprint 2: Game Development (Water Droplet Catching Game)

The user login authentication system will enable players to sign up, sign in, or play as a guest. During the sign-up process, users will be required to provide personal information such as their first name, last name, date of birth, gender (optional), profile picture, username, and password. Upon successful authentication, users will have access to their profiles and the game(s) at the last reached level. They will also be able to view their performance and history scores in every game.

The gameplay will revolve around the following key elements:

1. *Bucket Control: Players will be able to move a bucket left and right to catch falling water droplets.*
2. *Scoring System: Points will be awarded for each successfully caught droplet, with a target score required to win the game.*
3. *Win and Lose Conditions: Players will win the game upon reaching the target score and lose if they miss a certain number of droplets.*
4. *Difficulty Levels: The game will offer multiple difficulty levels, affecting factors such as droplet speed and game mechanics.*

This SRS document primarily focuses on defining the functional requirements of the water droplet catching game and the user login authentication system. It outlines the features, behaviors, and interactions expected from both components, providing a comprehensive guide for their development and implementation.ad

1.5 References

This SRS document for the water droplet game project does not refer to any external documents or web addresses. All specifications and requirements are contained within this document itself.

2. Overall Description

2.1 Product Perspective

The water droplet game project is a self-contained product developed as part of a software engineering class. It is not a follow-on member of a product family or a replacement for existing systems. The game is designed to provide an engaging and educational experience for students while demonstrating key concepts in software development, including user interface design, game mechanics, and system architecture.

The game operates as a standalone application within the Qt gaming framework. It does not interface with other systems or components.

Water Droplet Game

- ***User Interface****: This component includes different screens such as menu screens and the main game screen. The game screen contains interactive elements such as the bucket and falling water droplets.*
- ***Game Logic:*** *This component handles the core game mechanics, including collision detection between the bucket and water droplets, as well as the scoring mechanism.*
- ***User Authentication:*** *If applicable, this component manages user sign-up and sign-in processes to access game features and save progress.*

2.2 Product Functions

The water droplet game must perform the following major functions or allow users to perform the following actions:

- Control the movement of the bucket to catch falling water droplets.
- Detects collisions between the bucket and water droplets.
- Calculate and update the player's score based on the number of collected droplets.
- Display the player's current score and other relevant game information.
- Implement win and lose conditions based on the player's performance.
- Provide multiple difficulty levels for players to choose from.
- Play sound effects to indicate when a droplet is collected or missed.
- Allow users to sign up for an account, sign in, or play as a guest (if user authentication is implemented).

These functions represent the core features and interactions of the water droplet game. Detailed implementation specifications for each function will be provided in Section 3 of the SRS.

2.3 User Classes and Characteristics

The most important user class for this product is the Players category, as they are the primary users and consumers of the game experience. Other more typical user classes do not apply here since it is a school project and our goal satisfaction and engagement with the product.

2.4 Operating Environment

The game is designed to run on standard desktop and laptop computers. It should be

compatible with common hardware configurations found in typical consumer devices.

2.5 Design and Implementation Constraints

User Interface Design: The game's user interface must be intuitive and accessible to users of all skill levels. Developers must adhere to established design conventions and best practices to ensure a seamless user experience across different platforms and devices.

Programming Language and Framework: The game is developed using the Qt framework and programmed primarily in C++. Developers must have proficiency in these technologies to effectively implement game features and functionalities.

2.6 User Documentation

The water droplet game will include minimal user documentation aimed at providing basic instructions for gameplay. The documentation will consist of concise information on how to play the game, including control instructions and game objectives. Here's an example of the type of information that will be included:

Game Controls:

- To control the player holding the bucket, users should use the left and right arrow keys on the keyboard.
- Pressing the left arrow key moves the bucket to the left, while pressing the right arrow key moves the bucket to the right.

Game Objective:

- The objective of the game is to catch as many falling water droplets as possible using the bucket.
- Each caught droplet earns the player points, while missing a droplet deducts points.
- The game continues until the player reaches a certain score target or misses a specific number of droplets, depending on the selected game mode.

The user documentation will be concise and straightforward, providing users with the necessary information to start playing the game without the need for extensive tutorials or manuals.

2.7 Assumptions and Dependencies

Assumptions for the water droplet game project include the stability of the development environment, compatibility with common desktop platforms, and stability of third-party libraries. Dependencies encompass reliance on the Qt framework, graphics libraries for rendering, input devices for controls, sound libraries if implemented, and specific deployment platforms for release. Any changes to these assumptions or dependencies could impact the project's timeline and success.

3. External Interface Requirements

3.1 User Interfaces

The software components requiring a user interface include:

- Login-in/Sign-up Screen
- Main Game Screen
- User Profile Screen (includes game performance metrics)

3.2 Hardware Interfaces

The water droplet game primarily interfaces with standard desktop hardware components such as keyboard input. The nature of data and control interactions involves capturing user input events to control the movement of the bucket object (and user) within the game scene.

3.3 Software Interfaces

The water droplet game may interface with various software components to support its functionality. These interfaces include:

1. Graphics Libraries: The game utilizes graphics libraries such as Qt for rendering graphical elements and managing the user interface.
2. Input Handling: Interaction with user input devices, such as keyboards, is facilitated through software interfaces provided by the operating system.
3. Timer Mechanisms: Timer mechanisms provided by the Qt framework or other software libraries are used to implement game logic, including the movement of water droplets and timing of events.
4. Resource Management: Integration with the Qt resource system or similar mechanisms allows the game to efficiently manage and access external resources such as images and sound files.

These software interfaces enable the water droplet game to interact with external components and systems, ensuring smooth operation and a seamless user experience.

3.4 Communications Interfaces

The water droplet game does not require external communication interfaces as it operates as a standalone desktop application. Therefore, there are no specific communication protocols or interfaces involved beyond the standard interactions between the software and the user's input devices. The game relies on user input from devices such as keyboards and mice for interaction, with no need for communication with external systems or networks.

4. System Features

4.1 Bucket Movement

4.1.1 Description and Priority
Allow users to control the horizontal movement of a bucket object using keyboard input. Priority: High.

4.1.2 Stimulus/Response Sequences
- User presses the left arrow key.
- Bucket object moves left on the screen.
- User presses the right arrow key.
- Bucket object moves right on the screen.

4.1.3 Functional Requirements
- REQ-1: The system must detect keyboard input from the user.
- REQ-2: When the left arrow key is pressed, the bucket object must move left on the screen.
- REQ-3: When the right arrow key is pressed, the bucket object must move right on the screen.
- REQ-4: The bucket object's movement speed and responsiveness should be appropriate for smooth gameplay.
- REQ-5: The bucket object should not move beyond the boundaries of the game scene.

4.2 Water Droplet Generation

4.2.1 Description and Priority
Automatically generate water droplets falling from the top of the screen at regular intervals. Priority: High.

4.2.2 Stimulus/Response Sequences
- System initiates droplet generation at predefined intervals.
- Water droplet objects appear at random positions at the top of the screen.
- Droplets move downward in a straight line at a constant speed.

4.2.3 Functional Requirements
- REQ-1: The system must initiate droplet generation at regular intervals.
- REQ-2: Water droplets must appear at random positions at the top of the screen.
- REQ-3: Droplets must move downward in a straight line at a constant speed.
- REQ-4: Droplets must be generated in a continuous manner throughout the gameplay.
- REQ-5: The frequency and speed of droplet generation should be adjustable for gameplay balancing.

4.3 Collision Detection

4.3.1 Description and Priority
Detects collisions between the bucket object and falling water droplets. Priority: High.

4.3.2 Stimulus/Response Sequences
- Bucket object collides with a falling water droplet.

- System registers the collision event.
- Player's score is updated accordingly.

4.3.3 Functional Requirements
- REQ-1: The system must detect collisions between the bucket object and water droplets.
- REQ-2: Upon collision, the system must update the player's score based on predefined rules.
- REQ-3: Collisions must be accurately detected to ensure fair gameplay.

4.4 Scoring System

4.4.1 Description and Priority
Track and display the player's score based on successful water droplet catches. Priority: Medium.

4.4.2 Stimulus/Response Sequences
- Player catches a water droplet with the bucket object.
- System updates the player's score.
- Score is displayed on the game interface.

4.4.3 Functional Requirements
- REQ-1: The system must track the number of water droplets caught by the player.
- REQ-2: Each successfully caught droplet must increment the player's score by a predetermined value.
- REQ-3: The current score must be displayed prominently on the game interface for player feedback.

4.5 Game Over Condition

4.5.1 Description and Priority
Define conditions for triggering a game over state. Priority: Medium.

4.5.2 Stimulus/Response Sequences
- Player misses a certain number of water droplets.
- Player fails to achieve a target score within a specified time limit.

4.5.3 Functional Requirements
- REQ-1: The system must track the number of missed water droplets by the player.
- REQ-2: The game must end if the player misses a predefined number of droplets.
- REQ-3: The system must check if the player achieves a target score within a specified time limit.
- REQ-4: If the target score is not reached within the time limit, the game must end and display the final score.

4.1 User Authentication

4.1.1 Description and Priority

Allow users to sign up, sign in, or play as a guest. Priority: High.

4.1.2 Stimulus/Response Sequences

- User selects the sign-up option and fills in their personal details.
- System validates the entered information and creates a new user account.
- User selects the sign-in option and enters their username and password.
- System verifies the credentials and grants access to the user's account.
- User chooses to play as a guest, bypassing the authentication process.

4.1.3 Functional Requirements

- REQ-1: The system must provide a sign-up form to collect user information including first name, last name, date of birth, gender (optional), profile picture, username, and password.
- REQ-2: Usernames must be unique identifiers for each account.
- REQ-3: Passwords must consist of at least 8 characters and contain at least one number, upper and lower case letters.
- REQ-4: Upon successful sign-up, the system must create a new user account and store the provided information securely.
- REQ-5: The system must allow users to sign in using their username and password.
- REQ-6: Upon successful sign-in, the system must display the user's name, profile picture, and the current date.
- REQ-7: If the current date matches the user's date of birth, the system should send a birthday greeting.
- REQ-8: Users should have the option to play as a guest without creating an account.
- REQ-9: The system must provide functionality to view the history of scores from previously played games and compare them to the global best score.

4.2 Gameplay Mechanics

4.2.1 Description and Priority

Define the core gameplay mechanics including catching water droplets, scoring points, and determining game outcomes. Priority: High.

4.2.2 Stimulus/Response Sequences

- Water droplets fall from the top of the screen at varying speeds.
- Player moves the bucket left or right to catch falling droplets using keyboard input.
- When a droplet is caught, the player's score increases, and the droplet disappears.
- If a droplet is missed, the player loses a life.
- The game ends if the player misses a certain number of droplets or reaches the target score.
- Game speed increases gradually based on the number of droplets caught.

4.2.3 Functional Requirements

- REQ-1: Water droplets must fall from the top of the screen at varying speeds, increasing as the game progresses.
- REQ-2: The player must move the bucket left or right using keyboard input to catch

falling droplets.
- REQ-3: Each caught droplet increases the player's score by 5 points.
- REQ-4: The player loses a life if 5 droplets are missed.
- REQ-5: The game ends if the player misses the target score of 150 points or loses all lives.
- REQ-6: Game speed must increase gradually based on the number of droplets caught, doubling after every 5 droplets caught, up to a maximum speed of 16x.
- REQ-7: Sound effects must play when a droplet is caught and when a droplet is missed.
- REQ-8: The game should offer multiple difficulty levels (Easy, Medium, Hard) that adjust the speed of falling droplets.

5. Other Nonfunctional Requirements

5.1 Performance Requirements

The system should respond to user inputs promptly, with minimal lag or delay, to ensure smooth gameplay. Water droplets should fall at a gradually increasing speed, as per the game mechanics described in Section 4.

5.2 Safety Requirements

Since this is a software application, there are no specific safety requirements associated with it.

5.3 Security Requirements

- User authentication: The system should securely store user passwords and ensure that only authenticated users can access their accounts.
- Data protection: User data, including personal information and game scores, should be encrypted to prevent unauthorized access.
- Secure communication: Any communication between the client-side application and the server should be encrypted to protect sensitive information.

5.4 Software Quality Attributes

- Reliability: The system should be reliable and stable, with minimal crashes or unexpected behavior during gameplay.
- Usability: The user interface should be intuitive and easy to navigate, allowing players of all skill levels to enjoy the game.
- Maintainability: The codebase should be well-organized and documented to facilitate future updates and enhancements.
- Performance: The system should be able to handle a reasonable number of concurrent users without experiencing significant slowdowns or performance issues.
- Portability: The game should be compatible with different operating systems and screen resolutions to reach a wide audience of players.

5.5 Business Rules

As this project is primarily a learning exercise and not intended for commercial deployment, traditional business rules such as revenue generation, advertising policies, and user acquisition strategies are not applicable. However, ethical considerations regarding content appropriateness and user data privacy still apply.

6. Other Requirements

Database Requirements:

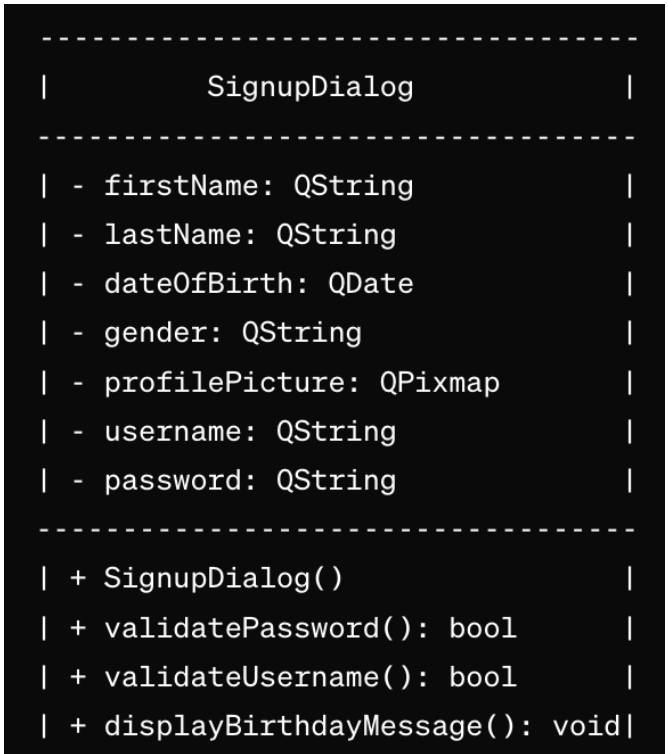
- The game will require a database to store user profiles, including information such as first name, last name, date of birth, gender, profile picture, username, password, and game scores.
- Additionally, the database will store game history data, including scores achieved by users in each game session.
- The database should support efficient data retrieval and management to provide a seamless user experience.

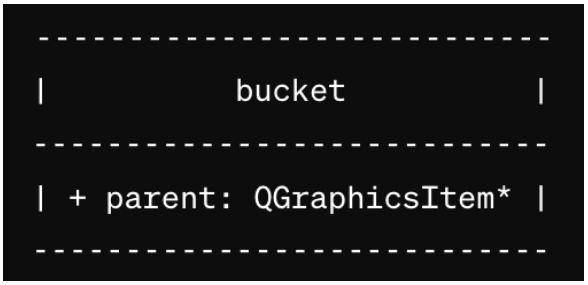
Appendix A: Glossary

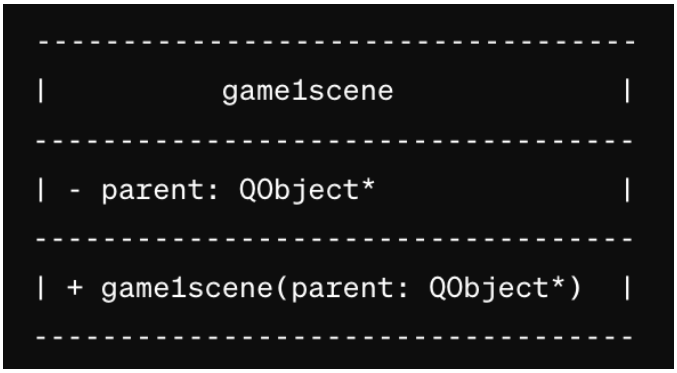
- *TBD*

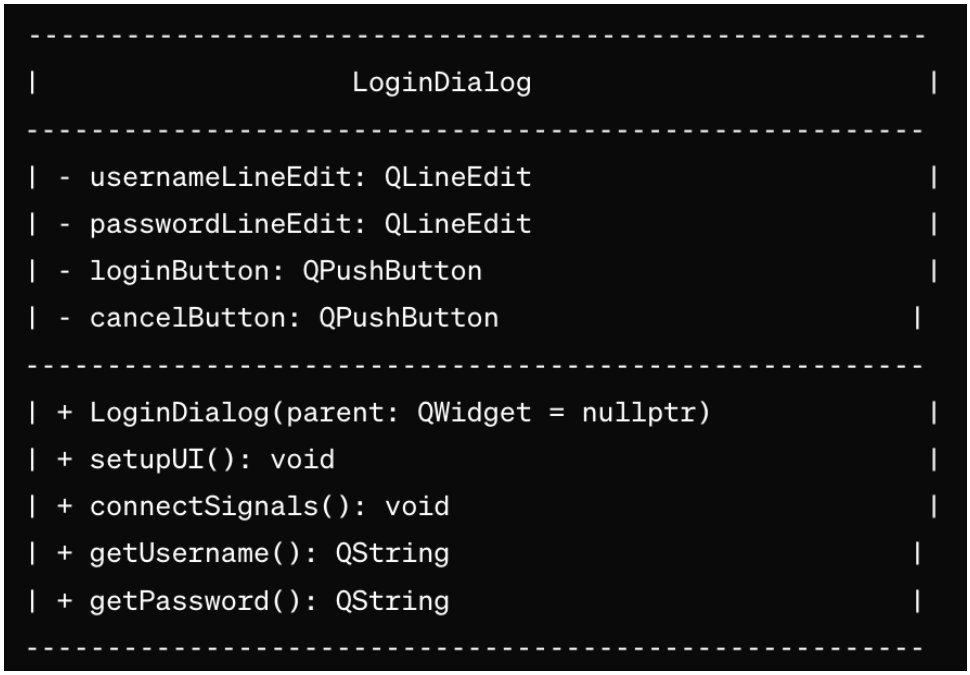
Appendix B: Analysis Models

Below are the UML Diagrams for the various components of this project

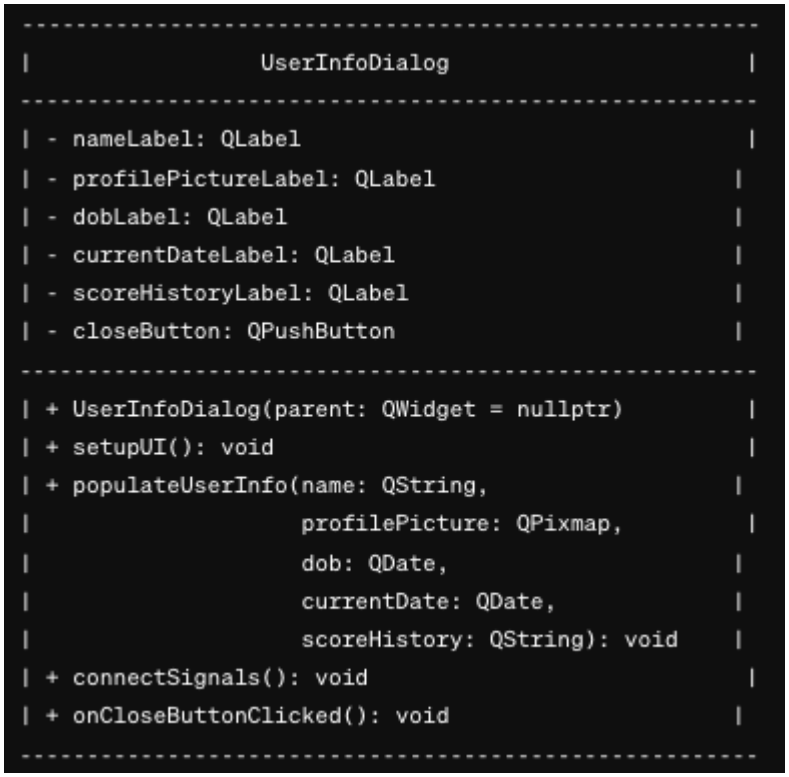







LoginDialog
- usernameLineEdit: QLineEdit
- passwordLineEdit: QLineEdit
- loginButton: QPushButton
- cancelButton: QPushButton
+ LoginDialog(parent: QWidget = nullptr)
+ setupUI(): void
+ connectSignals(): void
+ getUsername(): QString
+ getPassword(): QString


User
- firstName: QString
- lastName: QString
- dateOfBirth: QDate
- gender: QString
- username: QString
- password: QString
- profilePicture: QPixmap
+ setFirstName(firstName: QString): void
+ getFirstName(): QString
+ setLastName(lastName: QString): void
+ getLastName(): QString
+ setDateOfBirth(dateOfBirth: QDate): void
+ getDateOfBirth(): QDate
+ setGender(gender: QString): void
+ getGender(): QString
+ setUsername(username: QString): void
+ getUsername(): QString
+ setPassword(password: QString): void
+ getPassword(): QString
+ setProfilePicture(profilePicture: QPixmap): void
+ getProfilePicture(): QPixmap



Appendix C: To Be Determined List

1. *Glossary terms for Appendix A.*
2. *Database schema details for user profiles and game history.*