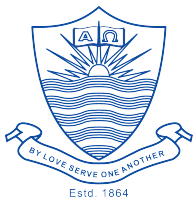

FORMAN CHRISTIAN COLLEGE
(A CHARTERED UNIVERSITY)

Software Requirement Specification

Department of Computer Science

Project Title

version: 1.4

Date: 15/03/2024

Faculty advisors:

Primary:	*Dr. Mubashar*
Secondary:	*Fakhir Shaheen*

Project group members:

Roll no	***Name***
2012114	*Ali Zaman*
2112190	*Gul Khan*
2255487	*Timothius Gill*

Contents

1 Revision History

1.1 Version 1.0

- Initial release of the software.

1.2 Version 1.1

- Bug fixes and performance improvements.

1.3 Version 1.2

- Added new feature XYZ.
- Improved user interface.

2 Introduction and Background

In this section, we provide an overview of the project and its background. We start by introducing the problem statement and the motivation behind the project. Next, we discuss the objectives and scope of the project, highlighting the key deliverables and target audience.

Furthermore, we present a brief background on the relevant technologies and concepts that form the foundation of our project. This includes a discussion on the existing literature, related work, and any previous research or implementations in the field.

By the end of this section, readers will have a clear understanding of the project's purpose, its significance, and the context in which it operates.

3 Game Overview

The game is set in a post-apocalyptic world where players must navigate through a desolate landscape filled with dangerous creatures and treacherous obstacles. The objective of the game is to survive and find a way to restore civilization.

Players start with limited resources and must scavenge for supplies, weapons, and shelter. They can form alliances with other survivors or choose to go solo. The game features a dynamic weather system and day-night cycle, adding an extra layer of challenge and realism.

As players progress, they will encounter various quests and missions that will test their skills and decision-making abilities. They can choose to follow the main storyline or explore the open world at their own pace.

The game offers a wide range of customization options, allowing players to personalize their characters and equipment. They can upgrade their weapons, craft new items, and unlock special abilities as they level up.

With stunning graphics and immersive gameplay, this game promises to deliver an unforgettable gaming experience. Are you ready to embark on this epic adventure and shape the fate of the post-apocalyptic world?

4 Gameplay and Mechanics

The gameplay of our game revolves around a unique set of mechanics that provide an immersive and engaging experience for the players. In this section, we will discuss the key gameplay elements and mechanics that make our game stand out.

4.1 Objective

The main objective of the game is to complete various challenging levels by overcoming obstacles, solving puzzles, and defeating enemies. Players will need to use their skills, strategy, and quick thinking to progress through the game.

4.2 Character Abilities

Our game features a diverse range of characters, each with their own unique abilities and skills. Players can choose their preferred character and utilize their abilities to overcome different challenges in the game. Whether it's a powerful attack, a special ability to manipulate the environment, or a skill to heal and support teammates, each character brings a distinct playstyle to the game.

4.3 Level Design

The levels in our game are meticulously designed to provide a balanced and enjoyable experience for the players. Each level presents a new set of challenges, puzzles, and enemies that require players to think creatively and adapt their strategies. The level design incorporates both linear and non-linear paths, offering multiple ways to approach and complete objectives.

4.4 Progression System

To keep players engaged and motivated, our game features a progression system that rewards them for their achievements and progress. As players complete levels, they earn experience points, unlock new abilities, and discover hidden treasures. This progression system adds depth and replayability to the game, encouraging players to explore and master different gameplay mechanics.

4.5 Optional Figures

The figures above showcase some examples of the gameplay and mechanics in our game. Figure 1 demonstrates a challenging level with various obstacles and enemies. Figure 2 showcases different characters with their unique abilities. Finally, Figure 3 illustrates the rewards players can earn through the progression system.

5 Functional Requirements

5.1 User Authentication

The system shall provide user authentication functionality to ensure that only authorized users can access the system. This functionality should include the following requirements:

Figure 1: Example of a challenging level

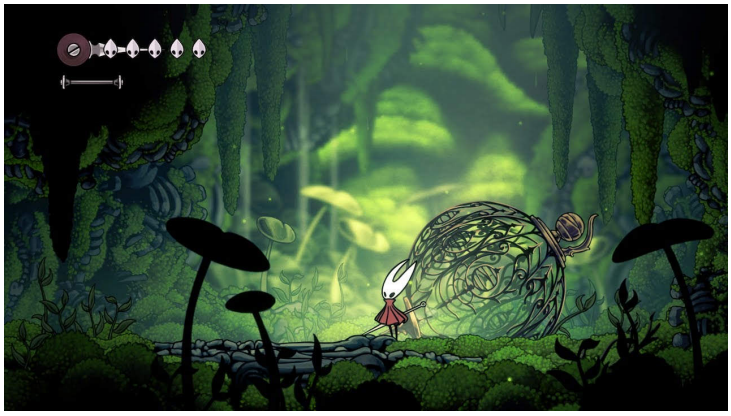

Figure 2: Different characters with unique abilities

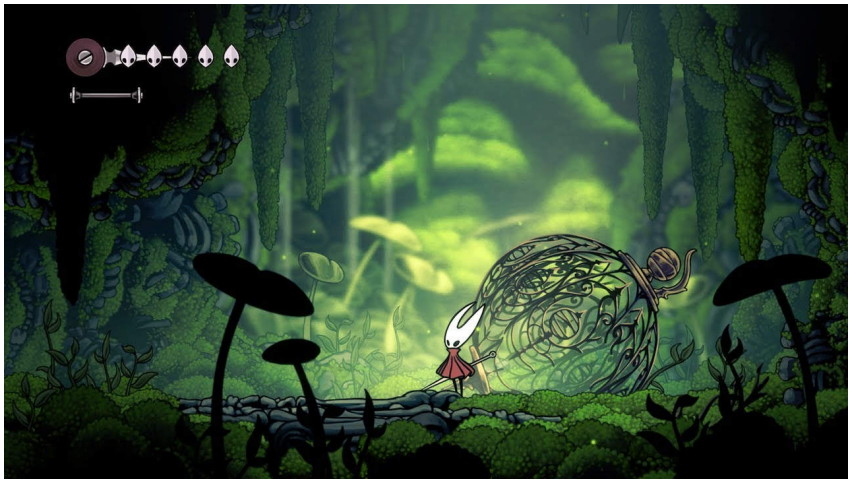

Figure 3: Progression system rewards

- The system shall allow users to create an account by providing a unique username and password.
- The system shall store user credentials securely to protect against unauthorized access.
- The system shall provide a login page where users can enter their credentials to access the system.
- The system shall enforce password complexity rules, such as minimum length and the use of alphanumeric characters.
- The system shall support password recovery options, such as email verification or security questions.

5.2 Data Management

The system shall provide functionality for managing data. This functionality should include the following requirements:

- The system shall allow users to create, read, update, and delete data records.
- The system shall enforce data validation rules to ensure the integrity and consistency of the data.
- The system shall provide search and filtering capabilities to allow users to find specific data records.
- The system shall support data import and export functionality to facilitate data exchange with other systems.

5.3 Reporting

The system shall provide reporting functionality to allow users to generate and view reports based on the stored data. This functionality should include the following requirements:

- The system shall provide predefined report templates for common use cases.
- The system shall allow users to customize report templates or create new ones.
- The system shall support exporting reports in various formats, such as PDF or Excel.
- The system shall provide options for scheduling and automating report generation.

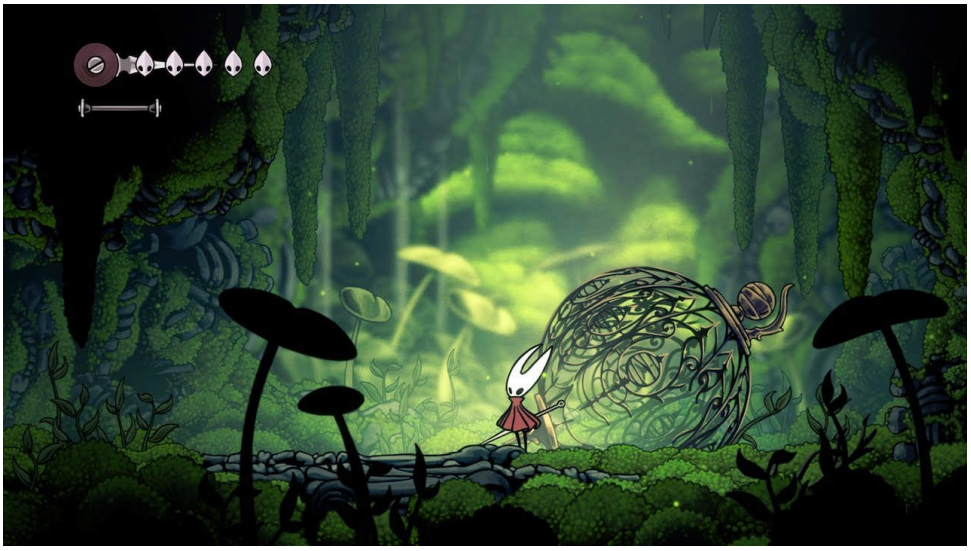

Figure 4: Example figure 1

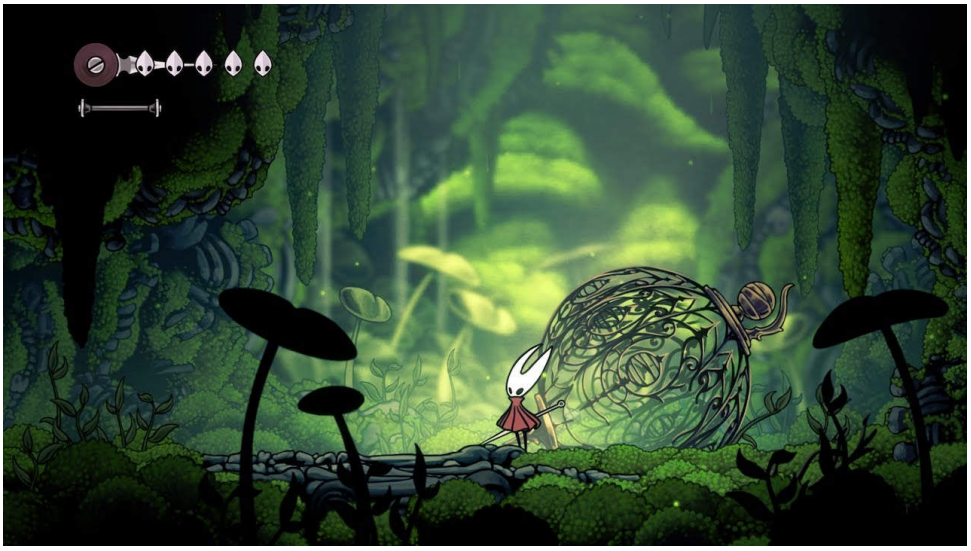

Figure 5: Example figure 2

6 Nonfunctional Requirements

6.1 Performance

The system should be able to handle a large number of concurrent users without significant degradation in response time. To achieve this, the backend server should be optimized for efficient data processing and the database should be properly indexed for quick retrieval of information.

6.2 Reliability

The system should have a high level of reliability to ensure that it is available and functional at all times. This can be achieved by implementing fault-tolerant mechanisms such as redundant servers, backup systems, and automated error detection and recovery.

6.3 Security

The system should have robust security measures in place to protect sensitive user data and prevent unauthorized access. This includes implementing secure authentication and authorization mechanisms, encrypting data in transit and at rest, and regularly updating and patching any security vulnerabilities.

6.4 Scalability

The system should be designed to handle increasing amounts of data and user traffic without sacrificing performance. This can be achieved by using scalable infrastructure components such as cloud services, load balancing, and horizontal scaling techniques.

6.5 Usability

The system should be intuitive and easy to use, even for users with limited technical knowledge. This can be achieved by following established user interface design principles, conducting user testing and feedback sessions, and providing clear and concise documentation.

6.6 Maintainability

The system should be designed in a modular and extensible manner to facilitate future enhancements and maintenance. This includes writing clean and well-documented code, using version control systems, and adhering to coding best practices and standards.

6.7 Compatibility

The system should be compatible with a wide range of devices, operating systems, and web browsers. This can be achieved by following web standards, using responsive design techniques, and conducting compatibility testing on different platforms.

6.8 Accessibility

The system should be accessible to users with disabilities, conforming to accessibility guidelines such as WCAG 2.0. This includes providing alternative text for images, using proper heading structures, and ensuring keyboard navigation support.

6.9 Legal and Regulatory Compliance

The system should comply with relevant laws, regulations, and industry standards. This includes data protection and privacy laws, intellectual property rights, and any specific regulations applicable to the domain of the system.

6.10 Performance

The system should be able to handle a large number of concurrent users without significant degradation in response time. To achieve this, the backend server should be

optimized for efficient data processing and the database should be properly indexed for quick retrieval of information.

7 Other Requirements

7.1 User Interface Mockups

Here are some user interface mockups for the system.

Figure 6: Mockup 1: Login Screen

Figure 7: Mockup 2: Dashboard

7.2 Performance Requirements

The system should be able to handle a minimum of 1000 concurrent users without any significant degradation in performance. Response time for common operations should be less than 1 second.

7.3 Security Requirements

The system should implement secure authentication and authorization mechanisms to protect user data. Passwords should be stored securely using industry-standard encryption algorithms.

7.4 Availability Requirements

The system should have a minimum uptime of 99.9

7.5 Documentation Requirements

The system should have comprehensive documentation that includes installation instructions, user guides, and API documentation.

7.6 Legal Requirements

The system should comply with all applicable laws and regulations, including data protection and privacy laws.

7.7 Training Requirements

The system should be intuitive and easy to use, requiring minimal training for end users. However, a training program should be provided to administrators and support staff to ensure they can effectively manage and troubleshoot the system.

7.8 Localization Requirements

The system should support multiple languages and provide localization options for date formats, currency symbols, and other region-specific settings.

7.9 Usability Requirements

The system should have a clean and intuitive user interface, with consistent navigation and clear error messages. It should be accessible to users with disabilities, following WCAG 2.0 guidelines.

7.10 Compatibility Requirements

The system should be compatible with modern web browsers (Chrome, Firefox, Safari, Edge) and mobile devices (iOS, Android).

7.11 Scalability Requirements

The system should be designed to scale horizontally, allowing for easy addition of new servers to handle increased load. It should also support vertical scaling by utilizing resources efficiently.

7.12 Maintainability Requirements

The system should be modular and well-documented, allowing for easy maintenance and future enhancements. Code should follow best practices and be thoroughly tested.

7.13 Support Requirements

The system should have a dedicated support team to address user inquiries and provide timely assistance. Support should be available during business hours and include a ticketing system for issue tracking.

7.14 Backup and Recovery Requirements

The system should have regular backups of data to prevent data loss in case of hardware failure or other disasters. A recovery plan should be in place to restore the system to a functional state in the event of a failure.

7.15 Performance Monitoring Requirements

The system should have performance monitoring tools in place to track system metrics, identify bottlenecks, and optimize performance as needed.

7.16 Third-Party Integrations

The system should be able to integrate with external systems, such as payment gateways, email services, and CRM systems, as required by the business.

7.17 Testing Requirements

The system should undergo thorough testing, including unit tests, integration tests, and user acceptance testing, to ensure its functionality and reliability.

7.18 Deployment Requirements

The system should have a well-defined deployment process, including version control, continuous integration, and automated deployment pipelines.

7.19 Change Management Requirements

The system should have a change management process in place to handle updates, bug fixes, and new feature releases. Changes should be properly documented and tested before being deployed to production.

8 Revised Project Plan

9 References

- John Doe, "A Comprehensive Guide to LaTeX", Publisher, 2021.
- Jane Smith, "Mastering LaTeX: Tips and Tricks", Another Publisher, 2022.
- David Johnson, "LaTeX for Beginners", Yet Another Publisher, 2023.

10 Appendix A: Glossary

10.1 Term 1

Definition of term 1 goes here.

10.2 Term 2

Definition of term 2 goes here.

10.3 Term 3

Definition of term 3 goes here.

11 Appendix B: IV & V Report