

Aim: Preparation of Software Requirements Specifications (SRS).

Results: SRS Document in given format

Certainly! I'll fill in the provided template based on the information about QuirkIQ as an e-learning platform.

1. INTRODUCTION**1.1 Product Overview**

QuirkIQ is an e-learning platform designed to transform the science education experience for 11th and 12th-grade students. The software aims to provide an intuitive and user-friendly environment, fostering a practical understanding of science concepts. The target audience includes high school science students seeking an engaging and comprehensive learning experience.

2. SPECIFIC REQUIREMENTS**2.1 External Interface Requirements****2.1.1 User Interfaces**

a) Characteristics:

- Intuitive and user-friendly design.
- Required screen formats, page layouts, and content for reports and menus.
- Optimized interface for seamless user engagement.

b) Optimization:

- List of do's and don'ts ensuring optimal user interaction.
- Verifiable requirements for the system's appearance.

2.1.2 Hardware Interfaces

- No specific hardware interfaces are required for QuirkIQ.

2.1.3 Software Interfaces

- Database System:
 - Name and Version: QuirkDB v1.0
 - Purpose: Manage user profiles, course content, and assessments.
 - Interface Definition: Reference to the document defining the interface.
- Operating System:
 - Name and Version: QuirkOS v2.5
 - Purpose: Support the execution of QuirkIQ.
 - Interface Definition: Reference to the document defining the interface.

2.1.4 Communications Protocols

- Specify various interfaces to communications, e.g., local network protocols.
- Reference well-defined protocols and parts/options the software needs to support.

2.2 Software Product Features

1. Essential Feature - User Registration and Profiles:

- Input: User registration details.
- Output: Unique user profiles tracking progress and achievements.
- State Change: Creation of user accounts and profiles.
- Functions: Secure registration, profile management, and personalized learning experiences.

2. Important Feature - Course Content Delivery:

- Input: User selection of courses.
- Output: Structured modules covering science curriculum.
- State Change: Progress tracking and completion of course modules.

- Functions: Organize and present content through multimedia, text, and simulations.

3. Desirable Feature - Virtual Internship Platform:

- Input: Student engagement in simulated scenarios.
- Output: Practical application of theoretical knowledge.
- State Change: Progress in virtual internships.
- Functions: Develop real-world skills through challenges, experiments, and projects.

2.3 Software System Attributes

2.3.1 Reliability

- Specify the required reliability of the final software system.
- Use MTTF (Mean Time To Failure) measurements for quantification.

2.3.2 Availability

- Specify required availability, including check pointing, recovery, and restart requirements.

2.3.3 Security

- Utilize cryptographic techniques.
- Maintain log or history data sets.
- Implement module-based functions for restricted communications.

2.3.4 Maintainability

- Specify attributes related to the ease of software maintenance.
- Include requirements for modularity, interfaces, and complexity.

2.3.5 Portability

- Define attributes related to the ease of porting the software to other systems.
- Include host dependency, portable language use, and operating system considerations.

2.3.6 Performance

- Specify static and dynamic numerical requirements for the software.
- Define measurable terms for minimum data storage, transactions per second, etc.

2.4 Database Requirements

- Specify logical requirements for information in the database.
 - Include types of information, accessing capabilities, data entities, and integrity constraints.
-

Questions:

1. Explain various steps involved in Requirement Engineering.

Requirement Engineering (RE) is a crucial phase in software development that focuses on understanding and defining the needs and constraints of the stakeholders. The goal is to establish a clear and complete set of requirements that serve as a foundation for designing and building the software system. The following are the various steps involved in Requirement Engineering:

- Feasibility Study:
 - Identify and evaluate the feasibility of the proposed system.
 - Consider technical, operational, and economic aspects to determine if the project is viable.
- Requirement Elicitation:
 - Interact with stakeholders to collect requirements.
 - Use various techniques such as interviews, surveys, workshops, and observations to gather information.
- Requirement Analysis:
 - Analyse and refine the collected requirements.
 - Identify inconsistencies, conflicts, and ambiguities.
 - Prioritize requirements based on importance and dependencies.
- Requirement Specification:
 - Document the requirements in a clear and unambiguous manner.
 - Use natural language, diagrams, and other appropriate notations to represent requirements.
 - Create different types of documents like a requirements document, use cases, and user stories.
- Requirement Validation:
 - Review and validate the requirements with stakeholders.
 - Ensure that the documented requirements accurately represent the stakeholders' needs and expectations.

- Address any concerns or issues raised during validation.
- Requirement Management:
 - Establish a process to manage changes to requirements.
 - Track and control changes to ensure that they are properly evaluated, approved, and implemented.
 - Maintain a traceability matrix to link requirements to design, implementation, and testing.
- Prototyping:
 - Create prototypes to provide stakeholders with a tangible representation of the system.
 - Gather feedback on the prototype to refine and finalize requirements.
- Documentation:
 - Maintain comprehensive documentation throughout the requirement engineering process.
 - Keep records of decisions, changes, and rationale to aid in future development and maintenance.
- Communication:
 - Ensure effective communication between all stakeholders.
 - Keep stakeholders informed about the progress of requirement engineering activities.
- Verification and Validation:
 - Verify that the specified requirements are complete, consistent, and meet quality standards.
 - Validate that the implemented system satisfies the specified requirements.
- Requirements Traceability:
 - Establish and maintain traceability links between requirements and other artifacts.
 - Traceability helps in understanding the impact of changes and ensures that all requirements are addressed.

Outcomes:

CO3 : Demonstrate requirements, modelling and design of a system

Conclusion: (Conclusion to be based on the outcomes achieved)

Prepared a Software Requirements Specifications (SRS) for my own project successfully.

Grade: AA / AB / BB / BC / CC / CD /DD

Signature of faculty in-charge with date

References:

Books:

1. Roger S. Pressman, Software Engineering: A practitioners Approach, 7th Edition, McGraw Hill, 2010.
2. Technical report on Guidelines for Documents Produced by Student Projects In Software Engineering based on IEEE standards
3. <https://www.sharelatex.com/>