Software Requirements Specification

for

Placement Preparation Module Focusing on DSA

Version 1.0 approved

Prepared by <author>

<organization>

<date created>

Table of Contents

Contents

Revision History

Name	Date	Reason For Changes	Version

1. Introduction

1.1 Purpose

The placement preparation DSA module is a dynamic website for students to build a strong foundation in data structures and algorithms, providing them with structured learning resources, visualized learning materials, practice quizzes and challenging programming questions.
The placement preparation DSA progress tracking website can provide a comprehensive and supportive learning environment to the students to prepare better for the placements, and ultimately helping students master the essentials DSA concepts.

1.2 Intended Audience and Reading Suggestions

The intended audience of the Placement Preparation Module are students which give the quizzes and practice coding through coding questions and learn through content provided. And the admin manages content and add content of the website.

1.3 Product Scope

- The placement preparation DSA module provides the topic wise and level wise practice quizzes and challenging programming questions.
- It gives user a detailed performance and activity report in graphical format.
- Users gets content and short study material for whole DSA syllabus.
- User can ask questions and it can be answered by any other users so that user's doubt can be cleared.
- There is Cheatsheet provided for summarized revision of the syllabus.
- DSA learning Roadmap is provided to guide users in best possible way.
- User can view content with or without login. Only giving quizzes, solving coding questions, participating in discussion and viewing dashboard requires login,.
- Admin will have access to control and manage content, quizzes and coding questions.
- Code editor will be provided wit multi language support.
- User will get notifications for any updates and activities and also reminders

1.4 References

[1] M. Mcquaigue, D. Burlinson, K. Subramanian, E. Saule, and J. Payton, "Visualization, assessment and analytics in data structures learning modules," Proceedings of the 49th ACM Technical Symposium on Computer Science Education, 2018. doi:10.1145/3159450.3159460
[2] E. B. Costa, A. M. Toda, M. A. Mesquita, F. T. Matsunaga, and J. D. Brancher, "Interactive Data Structure Learning Platform," Computational Science and Its Applications – ICCSA 2014, pp. 186–196, June. 2014. doi:10.1007/978-3-319-09153-2_14

[3] D. Dicheva and A. Hodge, "Active learning through game play in a Data Structures course," Proceedings of the 49th ACM Technical Symposium on Computer Science Education, 2018. doi:10.1145/3159450.3159605
[4] E. Fouh et al., "Design and architecture of an interactive etextbook – the opendsa system," Science of Computer Programming, vol. 88, pp. 22–40, 2014. doi:10.1016/j.scico.2013.11.040

2. Overall Description

2.1 Product Perspective

Student: The student will be able to give the test on all the topics of DSA and solve programming questions related to the topic. They need to login for give test and solving programming questions and other than that all contents of the website are visible to all users with or without login. With login user can see their performance metrics.

Admin: The admin will manage and add new content of the website. Admin will also add or edit or delete quiz and programming questions.

2.2 Product Functions

Main Functionalities of this Placement Preparation Module are:

- *Providing a platform where users can learn DSA concepts, give quizzes, practice through coding questions.*
- *The platform includes the platform includes level wise and topic wise questions for better performance of students.*
- *It will also contain a question-and-answer feature so that the user gets their doubt clear.*
- *Providing Roadmap for users to see their journey of learning DSA.*
- *User can view contents of website with or without login, only giving quizzes, submitting coding questions, participating in discussion and viewing user dashboard will require login.*
- *Cheatsheet is provide for summarized content of time and space complexities of various data structure and algorithms.*
- *Allowing users to create and manage their profiles, including their personal information, interests, and academic history.*
- *The platform will show users their performance in a visual format to give user a better understanding of their progress.*
- *Coding problems can be solved through the code editor provided which supports various languages.*
- *User will get notifications for any updates and activities and also reminders to maintain consistency in learning and practice*

2.3 User Classes and Characteristics

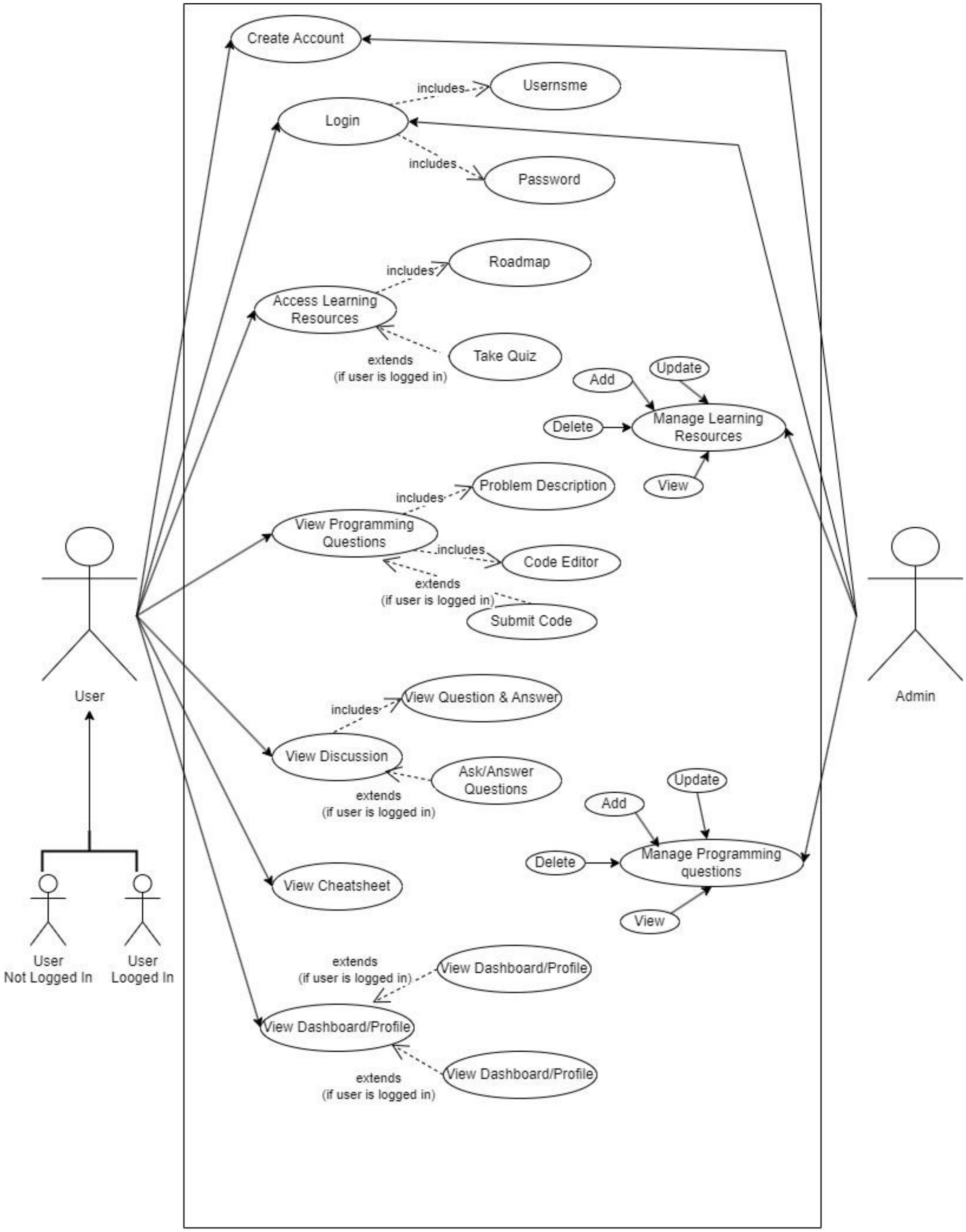


Fig 2.3 Use Case Diagram

2.4 Operating Environment

Our website can smoothly work on any latest version of browsers in both windows and mac-os operating systems.

2.5 Design and Implementation Constraints

Since the entire website is in only one language i.e English it might be difficult for other people speaking different languages.

2.6 Assumptions and Dependencies

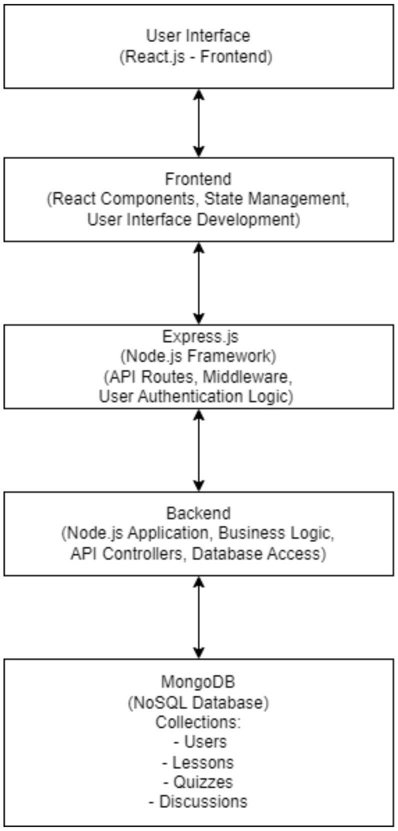
Assumptions:

- The users have access to a reliable internet connection to use the platform.
- The users will be able to receive timely support and assistance from the platform administrators in case of any issues or queries.

Dependencies:

- The platform will require regular updates and maintenance to keep up with changing technology and user needs.
- The platform's success will depend on the level of user adoption and engagement.

2.7 System Architecture



3. External Interface Requirements

3.1 User Interfaces

- Login UI
  - The login interface enables you to integrate user login with the content of our website. The system offers protection by storing passwords in encrypted form.
- RegistrationUI
  - The registration interface enables a new user to register to the system. Image based authentication is provided while registration
- HomePage UI
  - The homepage interface enables user to navigate through the concept and purpose of learning DSA and learn about features of our website. There is also contact us sections with option of newsletters.
- DSA Resourses/Content UI
  - there is a side navigation bar which contains topics of DSA and the page contains roadmap which displays users learning path. It contains topic wise short contents with short and precise information.
- Cheatsheet UI
  - The interface contains the summary of important DSA contents such as time and space complexity of various data structures and algorithms.
- Programming/coding questions UI
  - this interface contains all the coding questions will be displayed d based on topic wise and difficulty level wise.
- User Dashboard UI
  - The user dashboard displays the user information and their performance metrics in graphical format.
- Admin Edit/Change Content UI
  - In this interface Admin will add new content or edit the previous content. The old content will be displayed on left side and on right side new content will be fetched from the internet. the new content should be reviewed and then added to the website.
- Discussion UI
  - This interface has question and answer section and user will add question through filling a form with question and submit it and also user can add answers to the questions.
- Quiz UI
  - The exam interface enables the students to take exams. This interface will have timer to monitor the timing. There will be auto grading system.
- Coding environment and code editor UI
  - The coding interface will have Problem Description on left half side and code editor will be on right side.

3.2 Hardware Interfaces

Any Computer with

- Processor: Minimum 1 GHz; Recommended 2GHz or more

- Ethernet connection (LAN) OR a wireless adapter (Wi-Fi)
- Hard Drive: Minimum 32 GB; Recommended 64 GB or more
- Memory (RAM): Minimum 1 GB; Recommended 4 GB or above

3.3 Software Interfaces

- Windows: 7 or newer
- MAC: OS X v10.7 or higher
- The user's browser should be HTML5 compatible for a satisfactory user experience. such as Firefox, Chrome, Internet Explorer, Safari.

3.4 Communications Interfaces

The platform will use HTTP/HTTPS protocol over the internet.

4. System Features

The section illustrates the functional requirements for the product and the major services provided by the product.

4.1 Functional Requirement

REQ-1: User Authentication:
- Users should be able to create accounts and log in to access personalized features.

REQ-2: Learning Resources:
- Provide lessons on Data Structures and Algorithms (DSA) concepts.
- Categorize questions based on difficulty levels and topics.

REQ-3: Quizzes and Practice:
- Allow users to take quizzes to assess their understanding.
- Offer a coding practice section with a variety of problems.

REQ-4: Discussion Forum:
- Implement a Q&A feature for users to ask and answer questions.
- Enable users to participate in discussions related to DSA.

REQ-5: Roadmap:
- Design a roadmap for users to track their progress in learning DSA.

REQ-6: Content Access:
- Enable users to view website content without login.
- Require login for activities like giving quizzes, submitting coding questions, and participating in discussions.

REQ-7: Cheatsheet:
- Include cheatsheets summarizing time and space complexities.

REQ-8: Profile Management:

- Allow users to create, edit, and manage their profiles.
- Include personal information, interests, and academic history.

REQ-9: Visual Progress Tracking:

- Display visual representations of users' performance and progress.

REQ-10: Code Editor:

- Provide a code editor supporting multiple programming languages.

REQ-12: Accessibility:

- Ensure the platform is user-friendly for seamless navigation.

REQ-13: Feedback Mechanism:

- Implement a feedback system for users to provide suggestions or report issues.

REQ-14: Responsive Design

- Ensure the platform is accessible on various devices and screen sizes.

REQ-15: Security Measures:

- Implement secure protocols to protect user data and privacy.

REQ-16: Notifications:

- Incorporate a notification system to keep users informed about updates and activities.

REQ-18: Search Functionality:

- Include a search feature for users to quickly find specific content or questions.

REQ-20: Documentation:

- Provide clear documentation for users on how to use different features.

4.2 System Feature 2 (and so on)

5. Other Nonfunctional Requirements

5.1 Performance Requirements:

- Response Time: The platform should respond to user actions within 2 seconds for optimal user experience.
- Throughput: Support a minimum of 1000 concurrent users without significant performance degradation.
- Load Time: Web pages should load within 3 seconds on standard internet connections.

5.2 Safety Requirements:

- User Data Protection: Implement measures to prevent unauthorized access to user data.
- Data Integrity: Ensure that user data remains intact and unaltered during storage and processing.
- Error Handling: Provide clear error messages and gracefully handle unexpected situations to prevent data corruption.

5.3 Security Requirements:

- SSL Encryption: Use SSL/TLS encryption to secure data transmission between users and the server.
- Authentication: Employ secure authentication mechanisms, such as multi-factor authentication, to validate user identity.
- Authorization: Implement role-based access control to restrict users' access based on their roles.
- Data Encryption: Encrypt sensitive data, both at rest and in transit, to protect against data breaches.

5.4 Software Quality Attributes:

- Reliability: Ensure high system availability with a maximum of 99.9% uptime.
- Usability: Maintain a user-friendly interface with clear navigation and intuitive design.
- Scalability: Design the system to scale horizontally to handle a growing user base.
- Maintainability: Develop clean and modular code with comprehensive documentation to facilitate future maintenance.
- Compatibility: Ensure cross-browser compatibility and support for major operating systems.

5.5 Maintenance:

- Documentation: Provide extensive documentation covering system architecture, codebase, and APIs to aid future maintenance.
- Version Control: Use version control systems (e.g., Git) to track changes and manage codebase versions.
- Bug Tracking: Implement a bug tracking system to identify, prioritize, and resolve issues efficiently.
- Automated Testing: Maintain a suite of automated tests to ensure new updates do not introduce regressions.
- Regular Updates: Plan for regular software updates to incorporate new features, improvements, and security patches.

6. Other Requirements

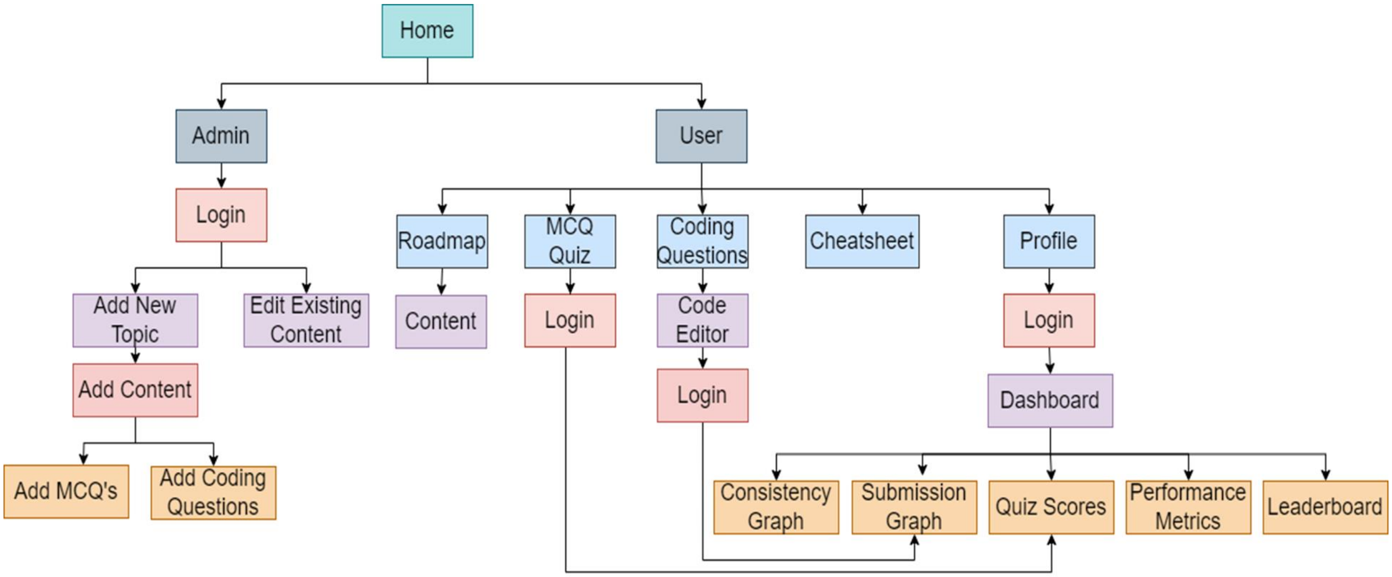
<Define any other requirements not covered elsewhere in the SRS. This might include database requirements, internationalization requirements, legal requirements, reuse objectives for the project, and so on. Add any new sections that are pertinent to the project.>

Appendix A: Glossary

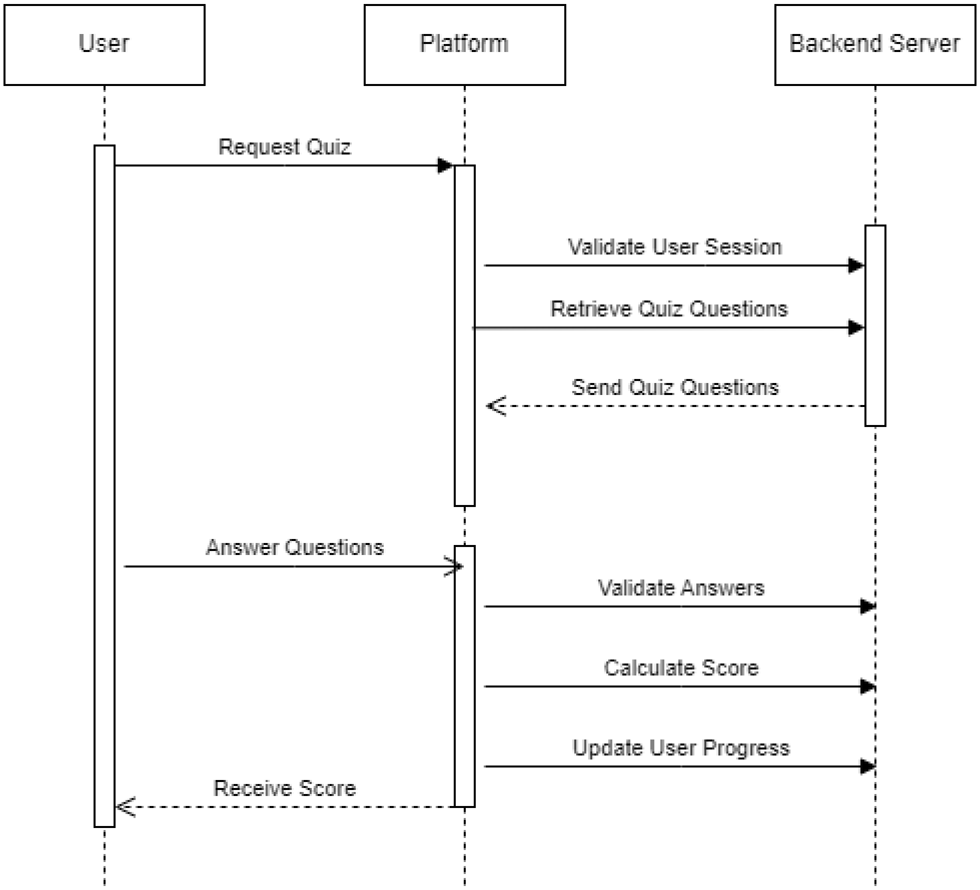
<Define all the terms necessary to properly interpret the SRS, including acronyms and abbreviations. You may wish to build a separate glossary that spans multiple projects or the entire organization, and just include terms specific to a single project in each SRS.>

Appendix B: Analysis Models

6.1 System Flow Diagram



6.2 Sequence Diagram


User
Platform
Backend Server
Request Quiz
Validate User Session
Retrieve Quiz Questions
Send Quiz Questions
Answer Questions
Validate Answers
Calculate Score
Update User Progress
Receive Score