TSS	Software Requirement Specification

Version ID Date: 10/07/2023
Document ID: SWD/ 01
Version ID: 1.0

Revision History

Doc. Ver.	Date	Author	Reviewer	Description of Revision
1.0	7/07/23	MD. Alamin Hossain	Sarwar Miral	
2.0	27/07/23	MD. Alamin Hossain	Sarwar Miral	

Table of Contents

Revision History **2**
Table of Contents **3**
1. Introduction **5**
1.1 Purpose 5
1.2 Scope 5
1.3 Intended Stakeholder 5
1.4 Overall Description 5
1.5 Overview 5
2. Technical platform **6**
2.1 Operating System 6
2.2 Development Environment 6
2.2.1 Backend 6
2.2.2 API Generation 6
2.2.3 Frontend 6
2.2.4 Database 7
2.2.5 Security 7
2.2.6 Mobile Device Compatibility 7
2.3 Overview 8
3. Functional Requirements **8**
3.1 User Registration 8
3.1.1 Requirements 8
3.2 Login 8
3.2.1 Admin Login: 8
3.2.2 Applicant Login: 8
3.2.3 Evaluator Login: 9
3.2.4 Requirements 9
3.3 Job Circular Application 9
3.3.1 Create Circular 9
3.3.2 Apply to Circular 10
3.3.3 Requirements 10
3.4 Applicant Approval 10
3.4.1 Requirements 10
3.5 Admit Card Generation 11
Requirements 11

3.6 Participant Tracking (Hidden Code on Copies)....11
Requirements....11
3.7 Marks Upload....12
Requirements....12
3.8 Internal Mailing System....13
Requirements....13
3.9 Applicant Dashboard and Notice Board....13
Requirements....13
3.10 Result Preparation....14
Requirements....14
3.11 Final Candidate Selection....14
Requirements....14
3.12 Add Evaluator And Assign Candidate....15
Requirements....15
3. 13 Overview....15
4. Non Functional Requirements....16
4.1 Performance....16
4.2 Security....17
4.3 Usability and User Experience....17
4.4 Reliability....17
4.5 Maintainability....18
4.6 Performance Requirements....18
4.7 Compatibility....18
4.8 Design Constraints....18
5. Software Quality Attributes....19
6. System Constraints....20
6.1 Technology and Platform Constraints....20
6.2 Timeframe and Deadlines:....20
6.3 Budget Limitations....20
7. System Design....21
7.1 Database Schema Design....21
7.2 User Interface Design....21
7.3 Integration with Mailing Service....21
7.4 Admit Card Generation Process....21
7.5 Evaluation and Marking Process....21
References....22
Definitions, Acronyms, and Abbreviations....22

1. Introduction

The Trainee Selection System(TSS) is a module/ Service that facilitates the selection process of BJIT academy trainees in different programs for the organization. It provides a platform for the applicants to register, apply for the desired job circular, undergo the approval process for a specific circular, generate admit card, trace the participants for the selection process, uploading the marks or every steps with the comments, communicate through internal mailing system/ notice board in different steps and selecting the final candidate for the training.

1.1 Purpose

The purpose of the TSS module is to optimize and simplify the trainee selection process within the BJIT. It aims to replace the manual and time-consuming tasks with an efficient and automated system by providing a user-friendly interface and range of functionalities. The TSS aims to enhance the overall selection process, improve communication, ensure fair and transparent selection procedure. It will serve as a central hub for managing the applicant's huge data, facilitating decision-making, and maintaining the records throughout the trainee selection process.

1.2 Scope

The scope of the TSS module/service includes the development of a platform to automate and streamline the trainee selection process. It covers the functionalities such as applicant registration, creating the job circular, application through the portal, applicant approval, admit card generation, participation confirmation, marks upload, internal mailing system, notice board system, result preparation and final candidate selection. Also .The Software Requirement Specification (SRS) document, which specifies the project's expectations and needs, will establish the module/service exact scope.

1.3 Intended Stakeholder

The BJIT Academy is the main Stakeholder of the project.

1.4 Overall Description

The TSS is a module/service designed to streamline the selection process for the BJIT academy trainees in various programs. It provides a platform for the applicant to register, apply for job circulars, undergo the approval process, generate admit cards, upload marks, communicate through internal mailing system/notice board. The TSS aims to optimize the selection process by replacing the manual tasks with an efficient and automated system.

1.5 Overview

The TSS module/service aims to enhance the trainee selection process within BJIT by providing a user-friendly interface and a range of functionalities. It serves as a central hub for managing applicant data, facilitating

decision-making, and maintaining records throughout the selection process. The TSS covers the entire process, from applicant registration to the final candidate selection.

Key functionalities of the TSS include applicant registration, job circular creation, application submission through the portal, applicant approval, admit card generation, participation confirmation, marks upload, internal mailing system, notice board system, result preparation, and final candidate selection. The module/service aims to improve communication, ensure fairness and transparency, and enhance the overall efficiency of the selection process.

The TSS module/service operates within the scope defined by the Software Requirement Specification (SRS) document, which outlines the specific expectations and needs of the project. It is intended for the exclusive use of BJIT Academy as the main stakeholder of the project.

The TSS module/service aims to optimize and simplify the trainee selection process within BJIT Academy by providing a user-friendly and efficient platform. It streamlines various activities involved in the selection process, from applicant registration to final candidate selection. The module/service ensures transparency, fairness, and effective communication throughout the process, ultimately facilitating the identification and selection of qualified trainee candidates.

2. Technical platform

The Trainee Selection System(TSS module utilizes a specific technical platform to ensure efficient functionality and seamless integration between the backend and frontend components.The technical platform for the TSS module may include the following components:

2.1 Operating System

The TSS module is designed to be platform-independent, allowing it to run on various operating systems. It is compatible with popular operating systems such as Windows, macOS and Linux. The system is developed using Java Spring Boot for backend and React for the frontend, both of which are compatible with multiple operating systems.

2.2 Development Environment

2.2.1 Backend

Java Spring Boot: The backend of the TSS module is developed using the Java Spring Boot framework. Spring Boot provides a robust and scalable environment for building enterprise-level application and API generation , making it suitable for developing the backend logic and handling data operation.

2.2.2 API Generation

Java Spring Boot: the Java Spring Boot framework is used to generate APIs for seamless communication between the backend and frontend components of the TSS module. It simplifies the process of designing and exposing the RESTful APIs, enabling efficient data transfer and interaction.

2.2.3 Frontend

React: the frontend off TSS module is developed using the React library. React is a popular JAVAScript library for building user interfaces, providing a component-based approach and efficient rendering. It allows the creation of interactive and responsive user interfaces, ensuring a smooth user-experience.

2.2.4 Database

The TSS module utilizes a SQL-based system to store and manage data. SQL is a standard language for relational databases, providing efficient data manipulation, retrieval and management capabilities. It allows creation and maintenance of a structured database schema to store applicant information, job circular data, marks, comments and other relevant data.

2.2.5 Security

The TSS module prioritizes the security of data and ensures the protection of sensitive information throughout the selection process. The following security features will be implemented:

1. **JWT Token-Based Authentication and Authorization:** The TSS module utilizes JSON Web Tokens(JWT) for authentication. When a user logs in, a JWT token will be created and sent to the client. This token will contain encrypted user information and is used for subsequent requests to authenticate and authorize the user. JWT tokens are secure, tamper-proof and can be validated without the need for server-side session storage which enhance the scalability and security.
2. **Spring Security:** The TSS leverages Spring Security which is a powerful framework for implementing authentication and authorization mechanisms. It provides a set of features such as role-based access control, request filtering and secure session management to ensure a secure seamless interaction within the system.
3. **Role-Based Access Control(RBAC):** Spring Security facilitates the RBAC, allowing administrators to define different roles and permissions for users. Each user is assigned to specific roles that determine specific endpoint, functionality and resources within the TSS. RBAC ensures this functionality that only the authorized individual can perform specific actions and access information, enhancing the overall access security.
4. **Secure Transmission:** The TSS module utilizes secure protocols such as HTTP to encrypt data transmitted between the client and server. This will ensure the security of sensitive information including users credentials, JWT token and protects from unauthorized interception or tempering.
5. **Logging and Auditing:** The TSS will maintain detailed logs of user activities, including login attempts, API requests.
6. **Input Validation and Sanitization:** The TSS will have strict input validation and sanitization techniques to mitigate some common security vulnerabilities, such as SQL injection, cross-site scripting(XSS) attacks.

2.2.6 Mobile Device Compatibility

The TSS module/service is designed to be responsive and compatible with mobile devices. The React frontend ensures that the user interface adapts to different screen sizes and resolutions, providing a consistent experience across devices, including smartphones and tablets. Applicants and stakeholders can access the system and perform necessary tasks on the go, enhancing flexibility and convenience.

To further enhance mobile device compatibility, the TSS module/service can be developed as a mobile application using frameworks such as React Native or as a progressive web application (PWA). This allows users to access the system through dedicated mobile apps or via web browsers on their mobile devices, providing a native-like experience and offline capabilities if needed.

2.3 Overview

In summary, the TSS module is designed to be compatible with various operating systems, flexible development which will enhance the accessibility for users on Windows, macOS and Linux platforms, and some basic and advanced security facilities.

3. Functional Requirements

3.1 User Registration

- Applicants should be able to register by providing their first name, last name, gender, date of birth, valid G-mail account, contact number, degree name, educational institute, CGPA, passing year, and present address.
- The registration process should allow applicants to upload their photo and CV/resume.

3.1.1 Requirements

REQUIREMENT ID	Requirement Description	Acceptability/ Completion Criteria	Limitations/ Constraints	Test case Identifier
TSS_001	Applicants can create a new account by providing basic information such as their required information and upload their CV.	Essential	Server might not be available	

3.2 Login

3.2.1 Admin Login:

- The Admin accesses the login page of the TSS.
- The system presents a login form requesting the Admin's username and password.
- The Admin enters their credentials.
- The system verifies the entered credentials against the stored admin user data.
- If the credentials are valid, the system grants the Admin access to the administrative dashboard.
- If the credentials are invalid, the system displays an error message, indicating incorrect login information.

3.2.2 Applicant Login:

- The Applicant navigates to the TSS login page.
- The system provides a login form for the Applicant to enter their username and password.
- The Applicant inputs their credentials.

- The system validates the entered credentials against the stored applicant user data.
- Upon successful validation, the system grants the Applicant access to their applicant dashboard or profile page.
- In case of invalid credentials, the system displays an error message, notifying the Applicant of the incorrect login details.

3.2.3 Evaluator Login:

- The Evaluator visits the TSS login page.
- The system presents a login form where the Evaluator can input their username and password.
- The Evaluator provides their login credentials.
- The system verifies the provided credentials against the stored evaluator user data.
- If the credentials are valid, the system allows the Evaluator to access their evaluation dashboard or relevant pages.
- If the credentials are invalid, the system displays an error message, indicating incorrect login information.

The login process serves as a means for the Admin, Applicant, and Evaluator to authenticate themselves and gain access to the respective areas of the TSS system. It helps ensure that only authorized individuals can perform their designated roles and access the associated functionalities.

3.2.4 Requirements

REQUIREMENT ID	Requirement Description	Acceptability/ Completion Criteria	Limitations/ Constraints	Test case Identifier
TSS_002	Users can log in using their email address and password.	Essential	Server might not be available	
TSS_003	If applicants forget their password then they should be able to recover the password	Essential	Applicants may not be registered.	
TSS_004	Users can log out of their account at any time by clicking the logout button in the app's menu.	Essential	Users may not be logged in.	

3.3 Job Circular Application

3.3.1 Create Circular

- Admin Can create a new job Circular with the needed information.
- Admin can also view the available circulars

3.3.2 Apply to Circular

- Registered applicants should be able to browse and apply for desired job circulars.
- The system should display available job posts and their details, such as job title, requirements, and application deadline.
- Applicants should be able to submit their application for specific job circulars.

3.3.3 Requirements

REQUIREMENT ID	Requirement Description	Acceptability/ Completion Criteria	Limitations/ Constraints	Test case Identifier
TSS_005	Admin can view the current job circular	Essential	Server down No circular Data available	
TSS_006	Create new Circular	Essential	Server down Circular exists	
TSS_007	Edit Circular information	Optional	Server down	
TSS_008	Applicants can view current circular in their dashboard	Essential	Server down	
TSS_009	Applicant can view the details of the circular and if suitable can apply	Essential	Server down	

3.4 Applicant Approval

- Admin panel members should have access to view applicant information.
- Admins should be able to sort and filter applicant data based on job post, gender, degree name, educational institute, CGPA, and passing year.
- Admins should be able to mark applicants as "APPROVED FOR INTERVIEW" for specific job circulars.

3.4.1 Requirements

REQUIREMENT ID	Requirement Description	Acceptability/ Completion Criteria	Limitations/ Constraints	Test case Identifier
TSS_010	Admin can see the applicants list based on the course and can filter the candidate in different categories.	Essential	Server might not be available No applicant available	
TSS_011	Admin can select or approve an applicant for the interview. That applicant will be selected only for that particular circular interview.	Essential	None	

3.5 Admit Card Generation

- The system should automatically generate admit cards for selected applicants.
- Admit cards should contain unique serial numbers, barcodes, or QR codes for personal identification during the selection process.

Requirements

REQUIREMENT ID	Requirement Description	Acceptability/ Completion Criteria	Limitations/ Constraints	Test case Identifier
TSS_012	Admit card will be generated for the particular candidate who will be selected for the written exam or interview	Essential	data not available	

3.6 Participant Tracking (Hidden Code on Copies)

- The system should generate a unique code for each participant's answer sheet during the written exam.
- The generated codes should be stored for future reference and identification.

Requirements

REQUIREMENT ID	Requirement Description	Acceptability/ Completion Criteria	Limitations/ Constraints	Test case Identifier
TSS_013	The system will show the selected candidate for the written exam	Essential	No data available	
TSS_014	On the candidate the system will generate a unique code for the written exam paper.	Essential	None	

3.7 Marks Upload

- Admins should be able to assign evaluators for marking.
- Evaluators should have the option to upload marks for each candidate, categorized based on predefined criteria.
- The system should validate and store the uploaded marks accurately.

Requirements

REQUIREMENT ID	Requirement Description	Acceptability/ Completion Criteria	Limitations/ Constraints	Test case Identifier
TSS_015	Admin can assign some candidate to a evaluator	Essential	Server might not be available no evaluator available	
TSS_016	Evaluator can see the candidates unique code list which he have been assigned	Essential	No candidate assigned	
TSS_017	Can upload marks based on the question numbers of that candidate	Essential	None	

3.8 Internal Mailing System

- The system should integrate a mailing service to send emails to applicants.
- Mails should be sent to inform applicants about their application status, such as selection for an interview, passing the written exam, or passing the technical viva.

Requirements

REQUIREMENT ID	Requirement Description	Acceptability/ Completion Criteria	Limitations/ Constraints	Test case Identifier
TSS_018	Admin can send mail to the selected candidates such as written, techenical_interview, aptitude test and HR interview.	Essential	Data not available	
TSS_019	Can notify to the notice board about the rest publication or course related update	Essential	System down	

3.9 Applicant Dashboard and Notice Board

- The system should provide an applicant dashboard or notice board section.
- Applicants should be able to view notices and notifications regarding their application status, such as interview selection, written exam results, and technical viva results.

Requirements

REQUIREMENT ID	Requirement Description	Acceptability/ Completion Criteria	Limitations/ Constraints	Test case Identifier
TSS_020	There will be an admin dashboard where he can see various informations about the courses and selection process or current status of a particular candidate	Essential	Server might not be available	

TSS_021	Can send notice through the dashboard to some participants	Optional	None	
TSS_022	Can publish results	Essential	Server now available	

3.10 Result Preparation

- Admins should be able to upload marks for the technical viva and HR viva rounds.
- The system should calculate and prepare the final results based on the uploaded marks.

Requirements

REQUIREMENT ID	Requirement Description	Acceptability/ Completion Criteria	Limitations/ Constraints	Test case Identifier
TSS_023	Admin can see the current status of a certain course	Essential	Server might not be available	
TSS_024	Admin can update or change the marks if necessary	Optional	None	
TSS_025	Admin can upload the marks of the interviews with comments and remarks	Essential	Server now available Data not available	

3.11 Final Candidate Selection

- Admins should have access to a dashboard or page displaying the final selected candidates for each job circular.
- The page should present a rank list of candidates sorted according to their scores, facilitating the selection of the final trainees.

Requirements

REQUIREMENT ID	Requirement Description	Acceptability/ Completion Criteria	Limitations/ Constraints	Test case Identifier
TSS_026	Admin can select the final candidate through system	Essential	Server might not be available data not available	
TSS_027	Can filter by remarks or comments	Optional	None	
TSS_028	Send the final selection candidate information	Optional	Data not available	

3.12 Add Evaluator And Assign Candidate

- Admin can addEvaluators for the system.
- Admil will add some candidates' unique code to the evaluator so that evaluator can not know the candidate informaions.

Requirements

REQUIREMENT ID	Requirement Description	Acceptability/ Completion Criteria	Limitations/ Constraints	Test case Identifier
TSS_028	Add some evaluator	Essential	Server might not be available	
TSS_029	Send login credential information to the evaluator.	Essential	None	

3. 13 Overview

This section sums up in the below table the main functionalities or services provided by the sub-system, which will be detailed in the following subsections. A use case diagram could be also used to list the main functionalities.

Serial No	Main Features	Description
1.	Registration	Applicants can register through the website by providing some basic user details.
2.	Login	Different user can login to the system by giving their information
3.	Job Circular Application	Admin can Create new job circular. Applicants can apply their desired circular through dashboard.
4.	Application Approval	Admin will approve some applicants by some filtering then they will be notified.
5.	Admit card generation	The system will auto generate the admit cards of the candidate who accepts the invitation for the written exam.
6.	Participant Tracking(Code)	System will generate unique code for the written exam paper which will be used to identify the participants mark by evaluator
7.	Marks Upload	The assigned evaluator can upload the marks of the written exam by the participant tracking code.
8.	Internal Mailing System	Admin can send mail to the participants.
9.	Applicant Dashboard and NoticeBoard	Applicants have a notice board and they can view the notices of the organization.
10.	Result Preparation	Admin will have the functionality to upload the marks of technical interview, aptitude test and HR interview marks and prepare the results of the candidate.
11.	Final Candidate Selection	Admin can select the final candidate for every particular job circular.
12.	Add Evaluator and Assign candidate	Admin can add evaluators for the organization. Admin can assign some candidates unique code to the evaluator so that he can upload the marks of the candidate without knowing the candidate information.

4. Non Functional Requirements

4.1 Performance

- Response Time:
    - The system shall provide fast response times to ensure a seamless user experience.

    - User interactions, such as registration, login, and application submission, should be processed promptly.
- Scalability
    - The system should be scalable to accommodate a growing number of users and job circulars.
    - It should be able to handle increased traffic and load without significant performance degradation.

4.2 Security

- User Authentication and Authorization
    - The system shall utilize secure authentication mechanisms, such as JWT (JSON Web Tokens), to authenticate and authorize users.
    - Only authenticated users with the appropriate permissions shall be able to access and perform specific actions within the system.
- Data Protection
    - User data, including personal information and uploaded documents, shall be stored securely and protected from unauthorized access or tampering.
    - The system shall employ encryption and secure data transmission protocols (HTTPS) to ensure data privacy.
- Access Control:
    - Different user roles (admin, applicant) shall have distinct levels of access and permissions within the system.
    - Admin users shall have privileged access to manage job circulars, applicant approvals, and result preparation.
    - Applicants shall have access to view and manage their own profile, application status, and notifications.

4.3 Usability and User Experience

- User-Friendly Interface:
    - The system shall provide an intuitive and user-friendly interface for easy navigation and interaction.
    - Clear and concise instructions and labels should be provided to guide users through the various processes and functionalities.
- Mobile Device Compatibility:
    - The system should be responsive and compatible with different mobile devices, ensuring a consistent user experience across various screen sizes and resolutions.
    - The user interface elements should adapt and adjust dynamically to provide optimal usability on mobile devices.

4.4 Reliability

- Availability:
    - The system shall be available and accessible to users without frequent interruptions or downtime.
    - Adequate measures, such as server redundancy and backup systems, shall be implemented to ensure high availability.
- Error Handling:

- The system should handle errors gracefully and provide informative error messages to users when unexpected situations occur.
- Proper error logging and monitoring mechanisms should be in place to facilitate issue resolution and system improvement.

4.5 Maintainability

- Code Modularity and Documentation:
  - The system's codebase shall be structured in a modular manner, promoting code reusability and maintainability.
  - Clear and comprehensive documentation should be provided, including code comments, API documentation, and system architecture overview, to aid in future maintenance and enhancements.

4.6 Performance Requirements

- The system should be able to handle a significant number of concurrent users, ensuring smooth performance even during peak usage periods.
- The system should be optimized to minimize resource consumption, such as memory and CPU usage, to support efficient and scalable operations.

4.7 Compatibility

- The system should be compatible with modern web browsers, ensuring proper functionality and consistent user experience across different browser versions.

4.8 Design Constraints

Scalability: The design should be scalable to accommodate potential growth in the number of applicants, job circulars, and evaluators. It should handle increased database size and user load without significant degradation in performance.

Security: The design should incorporate robust security measures to protect applicant data, evaluation results, and system resources. It should ensure secure user authentication, data encryption, and protection against common security vulnerabilities, such as SQL injection and cross-site scripting.

Usability: The user interface design should prioritize usability and provide a user-friendly experience for all user roles (admin, evaluator, applicant). It should be intuitive, responsive, and accessible, allowing users to easily navigate through the system and perform their tasks efficiently.

Performance: The system should be designed to achieve optimal performance. It should be able to handle concurrent user requests, execute database queries efficiently, and minimize response times. The design should consider techniques like caching, query optimization, and load balancing to ensure optimal performance.

Reliability: The system should be reliable and available for use. It should minimize the occurrence of system failures, errors, and data corruption. The design should consider fault tolerance, backup mechanisms, and error handling strategies to ensure system reliability.

Integration: The design should allow seamless integration with external systems, such as the mailing service for sending notifications and other external APIs or services that may be required for specific functionalities. It should define the integration points, data formats, and communication protocols needed for successful integration.

Maintainability: The design should facilitate ease of maintenance and future enhancements. It should adhere to coding best practices, use modular and reusable components, and follow industry-standard design patterns. The documentation should be comprehensive and up-to-date to assist with system maintenance and future development.

Compliance: The design should comply with relevant regulations and standards, such as data protection laws, privacy regulations, and industry-specific requirements. It should incorporate necessary safeguards and controls to ensure compliance with applicable regulations.

These design constraints provide guidance and considerations for the development team to ensure that the TSS system is designed in a way that meets performance, security, usability, and maintainability requirements while adhering to applicable constraints and standards.

5. Software Quality Attributes

REQUIREMENT ID	Requirement Description	Acceptability/ Completion Criteria	Limitations/ Constraints	Test case Identifier
TSS_029	The app should be easy to use and intuitive for users, with a clear and straightforward interface.	Essential	May require regular user testing and feedback to refine usability	
TSS_030	The app should have a fast and responsive interface, with minimal lag or delay in interactions.	Essential	May require regular performance testing and optimization to ensure high performance	
TSS_031	The app should be able to handle an increasing number of users and transactions without degradation in performance.	Essential	May require regular testing and scalability improvements to ensure scalability	

TSS_032	The app should have high availability and minimal downtime, with the ability to recover from failures and errors.	Essential	May require regular testing and reliability improvements to ensure reliability	

6. System Constraints

This section identifies the constraints and limitations that need to be considered during the development and implementation of the TSS system. The constraints can be categorized into technology and platform constraints, timeframe and deadlines, and budget limitations.

6.1 Technology and Platform Constraints

- The TSS system should be developed using specific technologies, frameworks, and programming languages as determined by the project requirements or organization's technology stack.
- The system should be compatible with the targeted platform(s) (e.g., web-based, mobile) and support the required operating systems, browsers, and devices.
- Any third-party software or libraries used should be compatible and properly integrated with the chosen technology stack.
- Security and privacy regulations or compliance requirements must be adhered to when implementing the system.

6.2 Timeframe and Deadlines:

- The project should adhere to specific timelines and deadlines for each phase, such as requirements gathering, design, development, testing, and deployment.
- The project schedule should consider any milestones, dependencies, or constraints specified by stakeholders.
- Adequate time should be allocated for testing, bug fixing, and user acceptance testing to ensure a high-quality and reliable system.

6.3 Budget Limitations

- The development and implementation of the TSS system should fit within the allocated budget, taking into account resources, licenses, hardware, software, and any other expenses.
- The project should consider cost-effective solutions and prioritize features and functionalities based on budget constraints.

It is essential to identify and address these constraints early in the project to ensure that the system is designed and implemented effectively within the available resources, time, and budget. By considering these constraints, the project team can make informed decisions and plan accordingly to meet the project objectives.

7. System Design

This section focuses on the design aspects of the TSS system, including the database schema, user interface, integration with a mailing service, admit card generation process, and evaluation and marking process.

7.1 Database Schema Design

The database schema design involves defining the structure and relationships of the database tables to efficiently store and manage the data. It includes the creation of tables, specification of primary and foreign keys, and establishing relationships between entities. The proposed database schema for the TSS system should be implemented based on the requirements outlined in the previous discussions.

7.2 User Interface Design

The user interface design aims to create an intuitive and user-friendly interface for the TSS system. It involves designing the layout, navigation, and interaction elements of the system to ensure a seamless user experience. The design should consider the needs of different user roles (admin, evaluator, applicant) and provide appropriate functionalities and views for each role.

7.3 Integration with Mailing Service

To enable communication with applicants, the TSS system needs to integrate with a mailing service. This integration allows sending necessary emails to applicants, such as notifications about application status, interview schedules, exam results, etc. The design should define the integration points, email templates, and mechanisms to trigger and track email communications.

7.4 Admit Card Generation Process

The admit card generation process involves automatically generating system-generated admit cards for selected applicants. Each admit card should have a unique serial number, barcode, and/or QR code for personal identification. The design should outline the steps, algorithms, and data sources required to generate the admit cards. It should also specify the format and layout of the admit cards, including the applicant's details and identification codes.

7.5 Evaluation and Marking Process

The evaluation and marking process involves assessing applicants' performance in different categories and recording their marks. The design should define the categories for evaluation (e.g., technical knowledge, communication skills) and provide a mechanism for evaluators to input and upload the marks for each applicant. The design should also consider the integration of marking categories with the database schema and provide appropriate interfaces for evaluators to access and update the marks.

The system design phase is critical to ensure a well-structured and efficient implementation of the TSS system. It establishes the foundation for the subsequent development and testing phases by defining the structure, interfaces, and processes that drive the system's functionalities.

References

Reference	Location
Requirement Specification	

Definitions, Acronyms, and Abbreviations

Term/Acronym	Definition
APP	Abbreviation of Application
API	Application Programming Interface
SRS	Software Requirement Specification
TSS	Trainee Selection System