

Hamdard University
Department of Computing
Final Year Project



Automated Platform For Undergraduate Admissions
(FYP-014/FL24)

Software Requirements Specifications

Submitted by
Waleed Ahmed (1734-2021)
Dua Rahim (1597-2021)
Zoya Sayeed (1696-2021)

Supervisor
Sir Iqbal Uddin Khan

Fall 2021

Automated Platform for Undergraduate Admissions	Version: 1.0
Software Requirements Specifications	Date: 12/01/2025
<document identifier>	

Document Sign off Sheet

Document Information

Project Title	Automated Platform for Undergraduate Admissions
Project Code	FYP-014/FL24
Document Name	Software Requirements Specifications
Document Version	1.0
Document Identifier	FYP-014-SRS
Document Status	Final draft
Author(s)	Waleed Ahmed, Dua Rahim, Zoya Sayeed
Approver(s)	Sir Iqbal Uddin Khan
Issue Date	12/01/2025

Name	Role	Signature	Date
Waleed Ahmed	Team Lead		
Dua Rahim	Team Member 2		
Zoya Sayeed	Team Member 3		
Sir Iqbal Uddin Khan	Supervisor		
Sir Suleman Naz	Project Coordinator		

Automated Platform for Undergraduate Admissions	Version: 1.0
Software Requirements Specifications	Date: 12/01/2025
<document identifier>	

Revision History

Date	Version	Description	Author
09-01-2025	0.1	First draft	Zoya, Dua
12-01-2025	0.2	Revised version	Sir Iqbal Uddin
15-01-2025	1.0	Final report	Zoya, Dua

Automated Platform for Undergraduate Admissions	Version: 1.0
Software Requirements Specifications	Date: 12/01/2025
<document identifier>	

Definition of Terms, Acronyms, and Abbreviations

Term	Description
Scrum	An agile project management methodology emphasizing iterative development and continuous feedback.
UI/UX	User Interface/User Experience, design principles focused on ease of use and accessibility.
MongoDB	A NoSQL database system used for scalable and flexible data storage.
Firebase	A platform for building web and mobile applications with backend support and hosting.
Frontend	The part of the application the user interacts with directly (HTML, CSS, JavaScript, React/Next.js).
Backend	The server side of the application responsible for logic and database operations (Node.js/Express.js).

Automated Platform for Undergraduate Admissions	Version: 1.0
Software Requirements Specifications	Date: 12/01/2025
<document identifier>	

Table of Contents

1.	Introduction	6
1.1	Purpose of Document	6
1.2	Intended Audience	
1.3	Abbreviations	
2.	Overall System Description	8
2.1	Project Background	
2.2	Problem Statement	8
2.3	Project Scope	
2.4	Not In Scope	9
2.5	Project Objectives	9
2.6	Stakeholders & Affected Groups	10
2.7	Operating Environment	10
2.8	System Constraints	
2.9	Assumptions & Dependencies	11
3.	External Interface Requirements	12
3.1	Hardware Interfaces	12
3.2	Software Interfaces	12
3.3	Communications Interfaces	13
4.	System Functions / Functional Requirements	14
4.1	System Functions	14
4.2	Use Cases	19
4.2.1	List of Actors	19
4.2.2	List of Use Case	
4.2.3	Use Case Diagram	20
4.2.4	Description of Use Cases	
5.	Non - Functional Requirements	
5.1	Performance Requirements	24
5.2	Safety Requirements	24
5.3	Security Requirements	
5.4	Reliability Requirements	
5.6	Supportability Requirements	
5.7	User Documentation	
6.	References	26

Automated Platform for Undergraduate Admissions	Version: 1.0
Software Requirements Specifications	Date: 12/01/2025
<document identifier>	

1. Introduction

The objective of this project is to offer a practical way to improve the admissions flow at FEST (Faculty of Engineering, Science, and Technology). The suggested system addresses the difficulties of manual operations by enhancing the admissions process's security, flexibility, reliability, and integrity. Data mishandling, errors, and delays are common outcomes of manual processing. By automating the admissions workflow, the suggested approach increases accuracy, decreases administrative burden, and improves efficiency. Through encryption and access control, it guarantees the safe handling of sensitive data and provides real-time updates and improved tracking. With the goal of making the process easier for both employees and candidates, the system is made to be scalable and flexible enough to accommodate future modifications. This solution guarantees a more dependable, safe, and efficient method for FEST by revolutionizing the admissions process.

1.1 Purpose of Document

The purpose of this Software Requirements Specification (SRS) document is to provide an overview of the design, architecture, and functionality of the Vulnerability Assessment and Penetration Testing (VAPT) application, developed using Scrum methodology. It serves as a blueprint for the development team, outlining the technical requirements and specifications necessary for successful implementation. This document ensures alignment with user needs, providing clear guidelines for the application's development, while setting milestones and tracking progress throughout the project.

Automated Platform for Undergraduate Admissions	Version: 1.0
Software Requirements Specifications	Date: 12/01/2025
<document identifier>	

1.2 Intended Audience

Development Team: Developers, architects, and engineers responsible for the design, implementation, and testing of the VAPT application.

Security Team: Security analysts, VAPT specialists, and information security officers focused on assessing and ensuring the application's security.

Project Managers: Individuals overseeing the entire development lifecycle, including planning, execution, and delivery.

Stakeholders: Individuals or groups with an interest in the functionality, security, and performance of the application, including clients, investors, and regulatory bodies.

Automated Platform for Undergraduate Admissions	Version: 1.0
Software Requirements Specifications	Date: 12/01/2025
<document identifier>	

2. Overall System Description

2.1 Project Background:

The admissions process at FEST (Faculty of Engineering, Science, and Technology) has been handled manually, leading to inefficiencies, errors, and delays. With the increasing number of applicants each year, the limitations of the manual system have become more evident, including difficulties in tracking data, verifying payments, and scheduling tests. The lack of automation has resulted in duplication of applications, communication delays, and a poor applicant experience. To address these issues, there is a clear need for an automated admissions system that streamlines data entry, payment verification, test scheduling, and communication. The proposed solution will improve accuracy, reduce administrative burdens, and enhance efficiency, providing a better experience for both staff and applicants, while supporting the growing volume of applications.

2.2 Problem Statement

The current manual processes at FEST for handling admissions present several challenges. Manual data entry is a key issue, as admission staff manually collect and input student data into Excel sheets, leading to potential errors and data duplication. Additionally, there is no automated system in place to check for duplicate applications, resulting in multiple entries for the same student. Payment verification is another area of concern, as the process is manual, time-consuming, and prone to errors. Delays in verifying admission fee payments can lead to postponements in scheduling admission tests. Similarly, test scheduling is inefficient when done manually, often causing scheduling conflicts and miscommunication with students. Furthermore, communication delays arise from the manual notification of students about test dates, eligibility, and admission offers. The lack of an automated communication system not only results in delays but also leads to a poor experience for prospective students, as they are left uncertain about the status of their applications.

Automated Platform for Undergraduate Admissions	Version: 1.0
Software Requirements Specifications	Date: 12/01/2025
<document identifier>	

2.3 Project Scope

The main objectives of this project are:

- 1-To automate the admissions process at FEST, reducing human error and delays.
- 2-To improve the security of sensitive applicant data through encryption and access control.
- 3-To enhance the overall user experience for both applicants and administrative staff.
- 4-To increase efficiency by providing real-time updates and better tracking of applications.
- 5-To ensure the system is scalable and flexible to accommodate future changes in admission requirements.

2.4 Not In Scope

The scope of this project does not include manual processes beyond the initial migration of data into the system. Additionally, the system will not integrate with university-wide platforms or portals other than the admissions platform itself. Post-admission management, such as course registration and tracking academic progress, is also outside the project scope. Furthermore, advanced features such as AI-based predictive analytics or machine learning for forecasting admissions trends are excluded from the current development.

2.5 Project Objectives

Enhance Accuracy: Ensure correct data handling, reducing human errors in the admission process.

Consistency: Apply admission criteria uniformly across all applicants for fairness.

Efficiency: Reduce time and effort required to process applications, payments, and notifications.

Automated Platform for Undergraduate Admissions	Version: 1.0
Software Requirements Specifications	Date: 12/01/2025
<document identifier>	

User-Friendly Interface: Provide an intuitive platform for applicants to submit and track applications.

Data Management & Scalability: Enable efficient storage and retrieval of data to handle large application volumes.

Transparency: Provide real-time updates and notifications about application status.

Compliance: Ensure adherence to institutional policies and regulations.

Improved Experience: Enhance overall satisfaction for applicants and administrative staff.

2.6 Stakeholders & Affected Groups

Applicants: The primary users of the admissions system who will benefit from a faster, more secure application process.

Administrative Staff: Employees who manage the admission process and will experience reduced manual workload and improved system efficiency.

Project Managers: Individuals overseeing the project's execution and ensuring its alignment with timelines and objectives.

FEST Faculty and Department Heads: Groups that require accurate, timely data on admissions to support resource planning.

Security Analysts: Personnel responsible for ensuring the safe handling and storage of sensitive data throughout the admissions process.

2.7 Operating Environment

The platform will operate as a secure web-based system accessible via both desktop and mobile devices. The environment is designed to ensure high availability, as it must remain functional during peak admissions periods. Robust security measures will be implemented to safeguard sensitive student data. The backend will leverage MongoDB or Firebase for efficient data storage, while the frontend will use React/Next.js with

Automated Platform for Undergraduate Admissions	Version: 1.0
Software Requirements Specifications	Date: 12/01/2025
<document identifier>	

Tailwind CSS for an intuitive and responsive user interface. The system will be hosted on cloud infrastructure to ensure scalability and reliability, accommodating large user volumes and evolving institutional needs.

2.8 System_Constraints

The project faces constraints including a limited timeframe for development, as it must be completed within the academic deadlines set for final year projects. Budget limitations restrict the purchase of additional hardware or premium software services. Institutional policies and regulatory standards may impose boundaries on system features and functionality. Resource limitations, such as a small team size and specific skillsets, also influence the project's scope and progress.

2.9 Assumptions & Dependencies

The success of the project assumes reliable internet connectivity for both applicants and university staff to access the system. Adequate hardware resources, such as development servers and devices for testing, are assumed to be available throughout the project lifecycle. The project's success also depends on active cooperation from Hamdard University for requirements gathering, system testing, and deployment. The system relies on third-party tools and frameworks, including Visual Studio Code, MongoDB, and Firebase, for development and hosting. Furthermore, regular engagement and constructive feedback from stakeholders are critical for ensuring the project aligns with their needs and expectations.

Automated Platform for Undergraduate Admissions	Version: 1.0
Software Requirements Specifications	Date: 12/01/2025
<document identifier>	

3. External Interface Requirements

3.1 Hardware Interfaces

The system requires specific hardware interfaces to support its operation effectively. On the user side, it requires devices capable of running modern web browsers, including desktops, laptops, tablets, and smartphones. For development and hosting, the system utilizes development servers equipped with adequate processing power and memory, such as machines with a minimum configuration of Intel Core i5 processors, 8GB RAM, and 500GB SSDs. Backup storage, such as external hard drives with at least 250GB capacity, will be used to ensure data redundancy and recovery. The physical addresses and logical structure for server hosting will align with cloud-based solutions for scalability and high availability. The expected behavior includes seamless data transfer, efficient processing of application requests, and reliable access across multiple hardware configurations.

3.2 Software Interfaces

The system interfaces with several applications to ensure seamless functionality. These include:

1. Database:
 - Name: MongoDB / Firebase
 - Owner: Project Team
 - Interface Details: The system interacts with MongoDB or Firebase for real-time data storage and retrieval, ensuring scalability and efficient data management.
2. Frontend Framework:
 - Name: React.js / Next.js
 - Owner: Project Team
 - Interface Details: The frontend application communicates with the backend through REST APIs to provide an interactive and responsive user experience.

Automated Platform for Undergraduate Admissions	Version: 1.0
Software Requirements Specifications	Date: 12/01/2025
<document identifier>	

3. Backend Framework:

- Name: Node.js / Express.js
- Owner: Project Team
- Interface Details: The backend processes application logic, communicates with the database, and serves data to the frontend using secure API endpoints.

4. UI Design Tool:

- Name: Figma
- Owner: Third-party Provider
- Interface Details: Used during the design phase for creating wireframes and mockups, ensuring consistency in UI/UX design.

3.3 Communications Interfaces

The system relies on several communication interfaces to facilitate connectivity and interaction with other systems and devices. A primary communication interface is the local area network (LAN) or internet connection, which ensures the system's web-based nature is accessible across devices and locations. Secure HTTP (HTTPS) protocols will be used for data transmission, ensuring encryption and secure interactions between clients and servers. Additionally, the system may interface with email servers for sending notifications and updates to applicants and staff. APIs will enable integration with payment gateways for secure fee processing and with potential university-wide systems for data exchange. These interfaces ensure the system remains reliable, secure, and capable of supporting real-time communication needs.

Automated Platform for Undergraduate Admissions	Version: 1.0
Software Requirements Specifications	Date: 12/01/2025
<document identifier>	

4. System Functions / Functional Requirements

4.1 System Functions

The system is designed to automate the undergraduate admissions process, ensuring accuracy, efficiency, and user satisfaction.

1. *End-User Functions*

- Applicants can create accounts, submit applications, make payments, and track their application status.
- Admission staff can review applications, schedule tests, and manage communications with applicants.

2. *Operator Functions*

- Manage user roles and access levels, ensuring secure role-based functionality for applicants, admins, and support staff.
- Oversee and modify application workflows, including deadlines, criteria, and admission stages. Generate summary reports on applications, payments, and test results for decision-making.
- Monitor system logs and user activity for auditing and compliance purposes.

3. *Support Functions*

- Assist applicants in account creation, application submission, and payment-related queries.
- Troubleshoot and resolve technical issues related to login, form submission, and notifications.
- Provide real-time support during admission test scheduling and result announcements.
- Facilitate communication between applicants and the admissions team for personalized assistance.

4. *Integration Functions*

- Connect securely with payment gateways for real-time fee verification and processing.

Automated Platform for Undergraduate Admissions	Version: 1.0
Software Requirements Specifications	Date: 12/01/2025
<document identifier>	

- Integrate with email and SMS services for automated notifications to applicants. Synchronize with third-party tools (e.g., MongoDB/Firebase) for storage and retrieval of application data.
- Enable API-based communication with university-wide systems for seamless data exchange.

4.2 Performance Requirements

- The system should handle up to 1,000 concurrent users without noticeable performance degradation.
- All operations, including form submission and status updates, should complete within 5 seconds

4.3 Design Constraints

- The system must adhere to web accessibility standards to ensure usability for all applicants.
- Secure encryption must be implemented for data transmission.

4.4 Programming Language

- Frontend: HTML, CSS, JavaScript, React.js, Next.js
- Backend: Node.js, Express.js
- Database: MongoDB or Firebase

4.5 Interface Requirements

- A user-friendly web interface for applicants and administrative staff.
- REST APIs for communication between the frontend and backend components.
- Secure integration with payment gateways for fee verification.

Automated Platform for Undergraduate Admissions	Version: 1.0
Software Requirements Specifications	Date: 12/01/2025
<document identifier>	

System Function Table

Ref#	Functions	Category	Attribute	Details & Boundary Constraints
R1.1	Display admission portal for Fall 2024	Evident	Interface availability	Ensure the admission portal is accessible.
R1.2	Register new applicants	Evident	Input validation	Validate required fields (e.g., name, email, NIC).
R1.3	Allow program and campus selection	Evident	Dynamic dropdown	Options should be updated dynamically.
R1.4	Save applicant data to the database	Hidden	Data persistence	Ensure all submitted data is stored reliably.
R1.5	Display applicant information for editing	Evident	Editable fields	Allow users to update their info (e.g., address, phone).
R1.6	Manage concurrent user submissions	Hidden	Concurrent load handling	Support multiple users without delay.
R1.7	Provide password creation and confirmation	Evident	Password policy	Enforce strong password rules.

Automated Platform for Undergraduate Admissions	Version: 1.0
Software Requirements Specifications	Date: 12/01/2025
<document identifier>	

R1.8	Generate applicant ID	Hidden	Unique ID generation	Ensure each applicant has a unique identifier.
R1.9	Applicant General Info Page	Evident	Form Fields	Includes "Student Name," "Father Name," and other personal details.
R1.10	Program Preferences Selection Page	Evident	Program Options	Allows selection of campus, faculty, and preferences for programs.
R1.11	Qualification Details Page	Evident	Academic Records	Displays qualification details such as matric/intermediate marks.
R1.12	Voucher Details Page	Evident	Payment Info	Displays voucher details like account title, amount, and status.
R1.13	Summary Page for Applicant	Evident	Review & Confirm	Provides a summary for final confirmation of application.
R1.14	Online Admission Portal Interface	Evident	Admin Options	Displays functions like "Print Offer Letter" and "Reset Password."

Automated Platform for Undergraduate Admissions	Version: 1.0
Software Requirements Specifications	Date: 12/01/2025
<document identifier>	

R1.15	Provisional Admission Offer Letter	Evident	Confirmation Letter	Shows details like name, father name, application ID, and semester fee.
-------	------------------------------------	---------	---------------------	---

System Attributes/ Nonfunctional Requirements

Attribute	Details and Boundary Constraints	Category
Response Time	The system must provide feedback like form submissions, and status updates within 5 seconds.	Mandatory
Concurrent User Load	The system should support at least 1,000 users connected simultaneously.	Mandatory
Interface Design	The interface must be browser-based, graphical, and intuitive for users.	Mandatory
Accessibility	The system must comply with web accessibility standards.	Mandatory
Scalability	The system must handle increased workloads without significant performance drops.	Optional

Automated Platform for Undergraduate Admissions	Version: 1.0
Software Requirements Specifications	Date: 12/01/2025
<document identifier>	

4.6 Use Cases

4.6.1 List of Actors

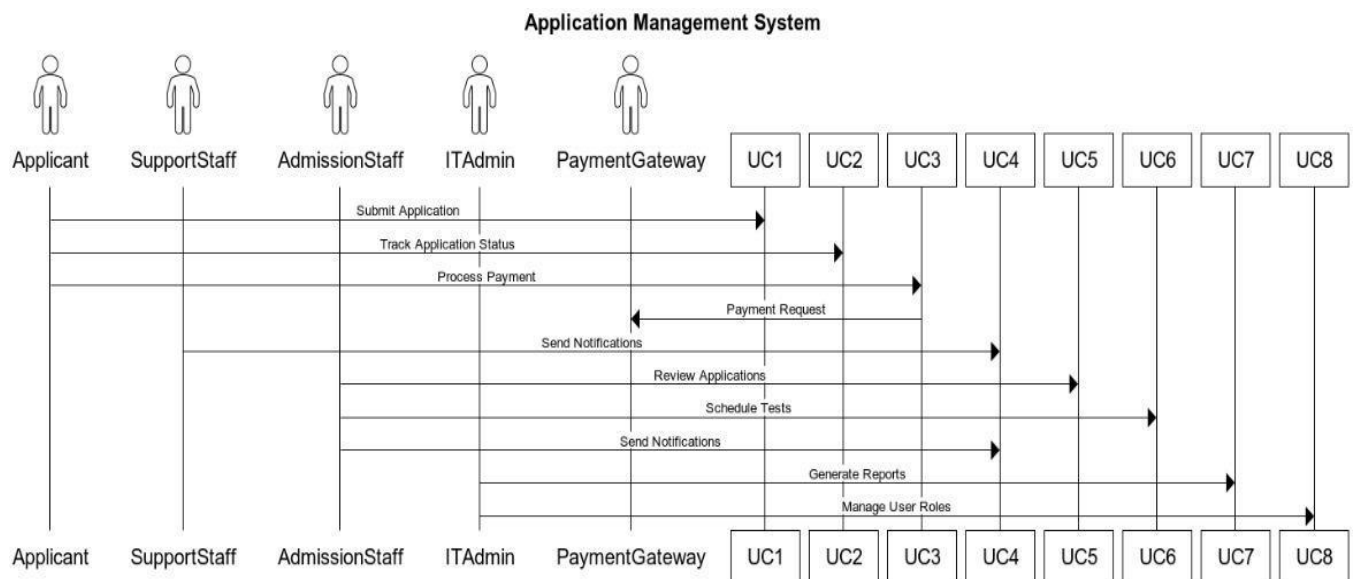
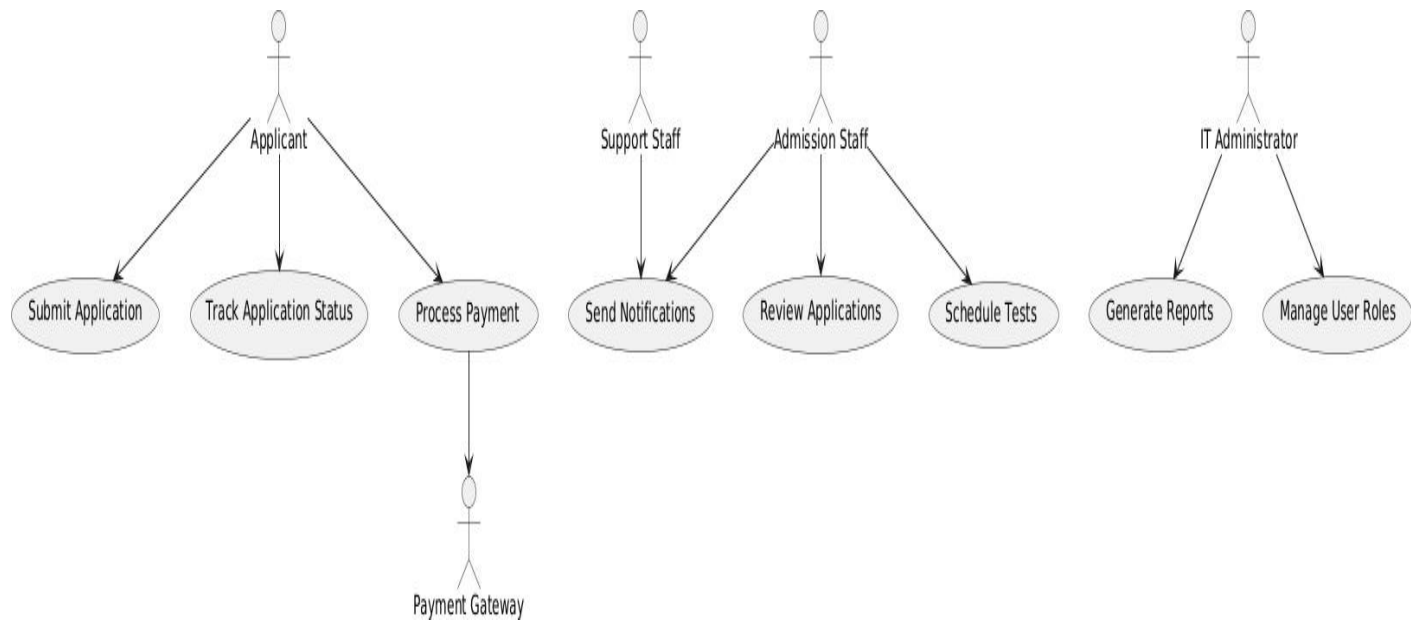
1. Applicant: Submits applications, tracks their status, and makes payments.
2. Admission Staff: Reviews applications, schedules tests, and communicates with applicants.
3. IT Administrator: Manages user roles, monitors system health, and performs system maintenance.
4. Support Staff: Assists users with technical or process-related issues.
5. Payment Gateway: Processes and verifies application fee transactions securely.

4.6.2 List Of Use Case

- 1- Submit Application: Applicants can submit their application forms online.
- 2- Track Application Status: Applicants can view the progress and status of their applications
- 3-. Process Payments: The system verifies and processes application fee payments.
- 4-. Review Applications: Admission staff reviews and validates submitted applications.
- 5- Schedule Tests: Admission staff schedules admission tests and communicates dates to applicants.
- 6- Generate Reports: Administrators can create reports summarizing admission statistics.
- 7- User Role Management: IT administrators assign and manage user roles for secure system access.

Automated Platform for Undergraduate Admissions	Version: 1.0
Software Requirements Specifications	Date: 12/01/2025
<document identifier>	

4.6.3 Use Case Diagram



Automated Platform for Undergraduate Admissions	Version: 1.0
Software Requirements Specifications	Date: 12/01/2025
<document identifier>	

4.6.4 Description of Use Cases

Section: Main			
Name:		Submit Application	
Actors:		Applicant	
Purpose:		Allow an applicant to complete and submit their application online.	
Description:		An applicant creates an account, fills out the necessary forms, uploads documents, and submits the application. The system validates the information and provides a confirmation upon successful submission.	
Cross References:		Functions: R1.1, R1.2 Use Cases: The applicant must have completed the "Create Account" use case.	
Pre-Conditions		The system is online and accessible. The applicant has created an account and logged in.	
Successful Post-Conditions		The application data is saved in the system. The applicant receives a confirmation email or notification.	
Failure Post-Conditions		The application is not saved. The system displays validation errors for missing or incorrect fields.	
Typical Course of Events			
Actor Action		System Response	
1	The applicant logs into their account.	2	The system verifies credentials and grants access.
3	The applicant fills out the application form.	4	The system validates the entered data in real time.
5	The applicant uploads required documents.	6	The system verifies the documents for format and size compliance.

Automated Platform for Undergraduate Admissions	Version: 1.0
Software Requirements Specifications	Date: 12/01/2025
<document identifier>	

7	The applicant submits the application.	8	The system saves the data and sends a confirmation notification.
Alternative Course			
Step 3:		The applicant provides incomplete or invalid data. The system highlights the errors and prompts corrections.	
Step 5:		The uploaded document format or size is invalid. The system rejects the file and requests a valid upload.	
Step 7:		The system encounters a technical error during submission. The applicant receives an error message and is asked to retry.	
Section: Track Application Status			
Typical Course of Events			
Actor Action		System Response	
1	Applicant logs into the system.	2	System verifies credentials and grants access.
3	Applicant navigates to the status page.	4	System retrieves and displays the status.
Alternative Courses			
Step 3		The system denies access and prompts the applicant to reset their password.	
Step 4		The system displays a "temporarily unavailable" message and logs the error for resolution.	
Section: Process Payment			
Typical Course of Events			
Actor Action		System Response	
1	The applicant selects the "Pay Now" option.	2	The system redirects to the payment gateway.
3	The applicant enters payment details and confirms.	4	The gateway processes the payment and provides a success response.
5	The system updates the application status to fee paid.	6	A receipt is generated and sent to the applicant.
Alternative Courses			
Step 3		The payment gateway rejects the transaction and prompts for correction.	
Step 4		The system notifies the applicant and allows retrying the transaction.	
Step 5		A support ticket is generated for manual intervention to resolve the discrepancy.	

Automated Platform for Undergraduate Admissions	Version: 1.0
Software Requirements Specifications	Date: 12/01/2025
<document identifier>	

Section: Review Applications			
Typical Course of Events			
Actor Action		System Response	
1	Staff logs into the admin portal.	2	System verifies credentials.
3	Staff views the list of applications.	4	System retrieves and displays the data.
5	Staff reviews and updates application status.	6	System saves the changes.
Alternative Courses			
Step 3		The system highlights the issue and flags the application for further review.	
Step 5		The system logs the error and the staff is promoted to retry after a refresh	
Section: Schedule Tests			
Typical Course of Events			
Actor Action		System Response	
1	Staff selects applicants for scheduling.	2	System retrieves eligible applicants.
3	Staff assigns test dates and times.	4	System updates the test schedule.
5	Notifications are sent to applicants.	6	Applicants receive test details.
Alternative Courses			
Step 3		The system suggests alternative dates or times based on availability.	
Step 5		The system logs the error and notifies staff for manual correction.	
Section: Send Notifications			
Typical Course of Events			
Actor Action		System Response	
1	A triggering event occurs	2	System generates a notification message.
3	System sends the notification.	4	Confirmation is logged for delivery status.
Alternative Courses			
Step 3		The system skips the notification and alerts staff for updates.	
Step 4		The system retries delivery or schedules it for later.	

Automated Platform for Undergraduate Admissions	Version: 1.0
Software Requirements Specifications	Date: 12/01/2025
<document identifier>	

5. Non - Functional Requirements

5.1 Performance Requirements

- The system must handle up to 1,000 concurrent users without noticeable performance degradation.
- All operations, such as form submissions, payment processing, and status updates, should complete within 5 seconds.
- The database should efficiently handle large volumes of applications without impacting query times.
- A comprehensive user manual must be provided for applicants, detailing steps for account creation, application submission, and payment processing.
- Administrative documentation should include guides for managing user roles, reviewing applications, and troubleshooting common issues.
- FAQs and tutorial videos must be made available on the platform to assist users in resolving queries independently.

5.2 Safety Requirements

- The system must prevent data loss during critical operations, such as application submission or payment processing, using robust fail-safe mechanisms
- Regular backups of application data should be performed to mitigate the risk of loss due to hardware or software failure and the system must log errors for immediate resolution and notify administrators of system-level failures.

5.3 Security Requirements

- All data transmissions between users and the system must be encrypted using SSL/TLS protocols to prevent unauthorized access.
- User authentication must include secure login mechanisms, such as hashed passwords and session management.
- Role-based access control (RBAC) must ensure that users can only access features and data relevant to their roles.

Automated Platform for Undergraduate Admissions	Version: 1.0
Software Requirements Specifications	Date: 12/01/2025
<document identifier>	

- The system must detect and prevent unauthorized access attempts, including brute-force login attempts.

5.4 Reliability Requirements

- The system must maintain 99.9% uptime during the admissions cycle to support applicants and staff.
- Critical functions, such as payment processing and application submission, must operate without disruption under peak load conditions.

5.5 Usability Requirements

- The user interface must be intuitive and accessible, allowing applicants to navigate easily without prior training.
- The system must adhere to Web Content Accessibility Guidelines (WCAG) to ensure accessibility for users with disabilities.
- Real-time error messages and form validation must guide users in completing tasks without confusion.

5.6 Supportability Requirements

- system should allow for easy updates and patches without disrupting active user sessions.
- The modular architecture should enable future enhancements, such as integration with additional third-party tools.
- Detailed system logs should facilitate debugging and maintenance by the IT team.

5.7 User Documentation

- A comprehensive user manual must be provided for applicants, detailing steps for account creation, application submission, and payment processing.
- Administrative documentation should include guides for managing user roles, reviewing applications, and troubleshooting common issues.
- FAQs and tutorial videos must be made available on the platform to assist users in resolving queries independently.

Automated Platform for Undergraduate Admissions	Version: 1.0
Software Requirements Specifications	Date: 12/01/2025
<document identifier>	

6. References

1. **Creatrix Campus.** *Student Admission Management System.* Available at <https://www.creatrixcampus.com/student-admission-management-system>. This resource explains features and benefits of automated admission management systems.
2. **HEIApply.** *Admissions Software for Higher Education.* Available at <https://heiapply.com/education-admissions/admissions-software-higher-education>. A comprehensive guide to features, functionalities, and use cases of automated platforms for admissions.
3. **Capterra.** *Best Admissions Software Tools & Reviews.* Available at <https://www.capterra.com/admissions-software/>. A detailed review and comparison of various tools available for automating admission processes.
4. <https://element451.com/blog/how-to-automate-student-admissions>