

Hamdard University
Department of Computing
Final Year Project



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

Software Design 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 Design Specifications	Date:13-01-2025
<document identifier>	

Document Sign off Sheet

1.1.1 Document Information

Project Title	Automated Platform for Undergraduate Admissions
Project Code	FYP-014/FL24
Document Name	Software Design Specifications
Document Version	1.0
Document Identifier	FYP-014-SDS
Document Status	Final draft
Author(s)	Waleed Ahmed, Dua Rahim, Zoya Sayeed
Approver(s)	Sir Iqbal Uddin Khan
Issue Date	13/01/2025

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 Design Specifications	Date:13-01-2025
<document identifier>	

Revision History

Date	Version	Description	Author
13-01-2025	0.1	First draft	Zoya,Dua
15-01-2025	0.2	Revised version	Sir Iqbal Uddin
16-01-2025	1.0	Final report	Zoya,Dua

Automated Platform for Undergraduate Admissions	Version:1.0
Software Design Specifications	Date:13-01-2025
<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 Design Specifications	Date:13-01-2025
<document identifier>	

2 Table of Contents

1.1.1	Document Information	2
2	Table of Contents	5
3	Introduction	6
3.1	<i>Purpose of Document</i>	6
3.2	<i>Intended Audience</i>	6
3.3	<i>Document Convention</i>	6
3.4	<i>Project Overview</i>	6
3.5	<i>Scope</i>	7
4	Design Considerations	8
4.1	<i>Assumptions and Dependencies</i>	8
4.2	<i>Risks and Volatile Areas</i>	8
5	System Architecture	10
5.1	<i>System Level Architecture</i>	10
5.2	<i>Software Architecture</i>	12
	Layers	12
	Interactions Between Layers	12
6	Design Strategy	14
	Future System Extension or Enhancement	14
6.1	<i>Database Design</i>	16
6.1.1	ER Diagram	16
6.1.2	Data Dictionary	17
6.1.2.1	Data 1	17
6.1.2.2	Data 2	17
6.1.2.3	Data 3	18
6.1.2.4	Data 4	19
6.1.2.5	Data 5	19
6.2	<i>GUI Design</i>	20
6.2.1		20
		20
7	References	24
8	Appendices	25

Automated Platform for Undergraduate Admissions	Version:1.0
Software Design Specifications	Date:13-01-2025
<document identifier>	

3 Introduction

This project "**Automated Platform for Undergraduate Admissions**" aims to streamline the admission process, replacing manual operations with an automated, secure, and efficient system. The system's primary focus is reliability, flexibility, integrity, and security, ensuring a seamless experience for both applicants and administrative staff. Scrum methodology is used, allowing iterative development and continuous stakeholder feedback.

3.1 Purpose of Document

The purpose of this document is to provide a detailed overview of the project, including its objectives, methodology, scope, and intended outcomes. It is designed for stakeholders, including developers, project supervisors, and end-users such as admissions teams. The document employs object-oriented design methodology to ensure scalability and modularity in system development.

3.2 Intended Audience

This document is intended for:

- 1- Project Supervisors – To oversee and guide project progress.
- 2- Development Team – For implementing and adhering to the design and technical requirements.
- 3- Admissions Staff – As end-users who will operate the automated system.
- 4- Applicants – Indirect stakeholders benefiting from improved processes.

3.3 Document Convention

Font Style: Arial

Font Size: 12pt for body text; headings may use larger sizes for emphasis.

Automated Platform for Undergraduate Admissions	Version:1.0
Software Design Specifications	Date:13-01-2025
<document identifier>	

3.4 Project Overview

The proposed software system automates undergraduate admissions at Hamdard University. Its functionalities include:

Automated Platform for Undergraduate Admissions	Version:1.0
Software Design Specifications	Date:13-01-2025
<document identifier>	

Application Submission and Tracking: Applicants can apply online and track their progress.

Automated Payment Verification: Ensures timely and error-free fee verification.

Test Scheduling and Results: Manages scheduling for admission tests and sends automated notifications.

Transparency: Provides real-time updates to applicants regarding their status.

The system leverages modern technologies like ReactJS, NodeJS, and MongoDB for development and follows Scrum methodology to ensure adaptability and timely delivery.

3.5 Scope

The system will:

1. Automate Data Entry: Eliminate duplicate entries with unique ID verification.
2. Payment Processing: Streamline admission fee verification securely.
3. Test Assignment: Generate credentials for online tests post-payment verification.
4. Admission Confirmation: Automate test result evaluation and offer letters.

Out of Scope:

- Management of programs beyond undergraduate admissions.
- Integration with non-academic modules like financial aid.

Automated Platform for Undergraduate Admissions	Version:1.0
Software Design Specifications	Date:13-01-2025
<document identifier>	

4 Design Considerations

This section outlines the foundational elements necessary to create a comprehensive and efficient design for the "Automated Platform for Undergraduate Admissions." It addresses the assumptions, dependencies, and potential risks, ensuring the design remains adaptable to future changes.

4.1 Assumptions and Dependencies

The system design relies on the following assumptions and dependencies, which influence its architecture and functionality:

1. **Stable Internet Access:** Reliable internet connectivity is assumed for both applicants and the administrative staff to access the online platform.
2. **Software Environment:** The platform will be hosted on a scalable cloud environment (e.g., Firebase or MongoDB) to manage user load and ensure high availability.
3. **Technology Stack Compatibility:** Tools like ReactJS for frontend, NodeJS for backend, and MongoDB for database are assumed to integrate seamlessly without major technical issues.
4. **User Training:** Administrative staff are assumed to have basic technical skills or will undergo training to use the system effectively.
5. **Compliance Standards:** The system will adhere to relevant regulatory and institutional policies to ensure data security and integrity.

4.2 Risks and Volatile Areas

Several factors pose risks to the system design, and addressing these areas is crucial for a robust and flexible architecture:

-New Requirements: Changes in university admission policies or additional features requested by stakeholders can impact the design.

Mitigation: Incorporate a modular design approach with loosely coupled components, allowing easy updates or integration of new features.

-Technology Evolution: The rapid pace of technological change may render some selected tools or frameworks obsolete.

Mitigation: Choose widely supported, open-source technologies with active communities to ensure long-term viability. Also, maintain backward compatibility during upgrades.

Automated Platform for Undergraduate Admissions	Version:1.0
Software Design Specifications	Date:13-01-2025
<document identifier>	

-Scalability Challenges: As the number of applicants increases, the system may face performance bottlenecks.

Mitigation: Use cloud-based solutions with dynamic scaling capabilities and optimize database queries to handle high user loads efficiently.

-Data Security and Privacy: Handling sensitive applicant data introduces risks of breaches or compliance violations.

Mitigation: Implement robust encryption for data at rest and in transit, along with secure authentication mechanisms like multi-factor authentication.

-Dependency on External Systems: Integration with payment gateways or external test scheduling platforms could face interruptions.

Mitigation: Design the system with fallback mechanisms, such as manual verification workflows, to handle external system failures.

-Team Collaboration and Resource Management: Miscommunication or delays in development phases could lead to inefficiencies.

Mitigation: Utilize the Scrum methodology with daily stand-ups, sprint reviews, and retrospectives to ensure continuous progress and issue resolution.

-System Adoption: Users may resist switching from manual processes to an automated system.

Mitigation: Conduct user training sessions and provide a comprehensive user manual to ease the transition

Automated Platform for Undergraduate Admissions	Version:1.0
Software Design Specifications	Date:13-01-2025
<document identifier>	

5 System Architecture

5.1 System Level Architecture

The system is decomposed into several subsystems to ensure modularity, scalability, and maintainability. These subsystems and their roles are described below:

System Decomposition

- **User Management Subsystem:** Handles user authentication, registration, and role-based access control.
- **Application Management Subsystem:** Facilitates application submission, tracking, and status updates.
- **Payment Processing Subsystem:** Manages fee verification and integrates with external payment gateways.
- **Test Management Subsystem:** Automates scheduling, communication, and result management for admission tests.
- **Administrative Subsystem:** Provides tools for the admissions team to monitor and manage processes.

Relationships Between Elements

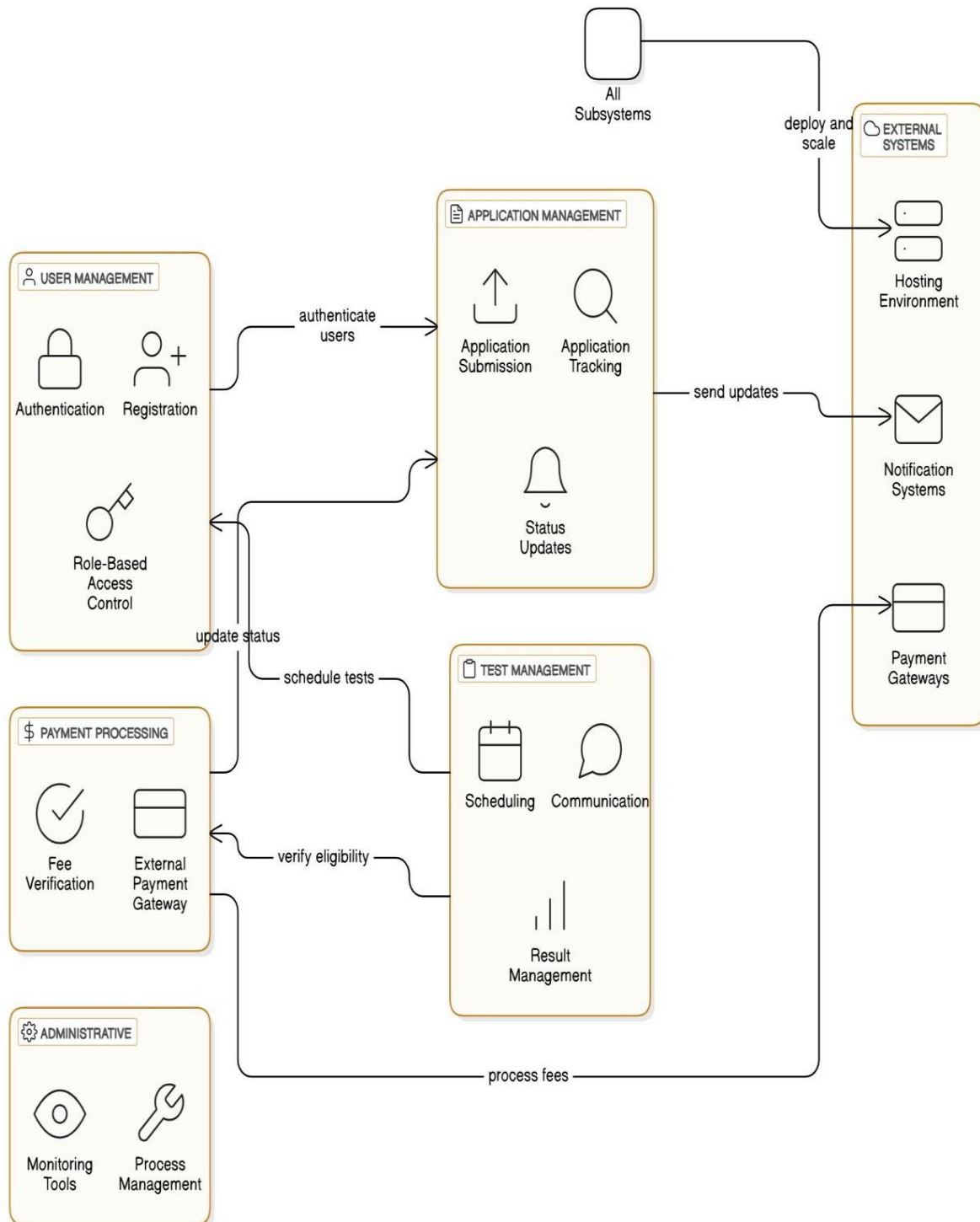
- **User Management interacts with Application Management** to ensure only authenticated users can submit or modify applications.
- **Payment Processing communicates with Application Management** to update the application status upon fee verification.
- **Test Management integrates with User Management and Payment Processing** to schedule tests only for eligible applicants.

Interfaces to External Systems

- **Payment Gateways:** Secure API integration for processing application fees.
- **Notification Systems:** Integration with email and SMS services for sending updates to applicants.
- **Hosting Environment:** Cloud-based infrastructure (e.g., Firebase or AWS) to deploy the application and manage scalability.

Global Design Strategies

- **Error Handling:** Centralized logging and error-handling mechanisms to track and resolve issues across subsystems.
- **Security:** Role-based access control, data encryption, and compliance with data protection regulations.



Automated Platform for Undergraduate Admissions	Version:1.0
Software Design Specifications	Date:13-01-2025
<document identifier>	

5.2 Software Architecture

The software architecture follows a layered architecture to ensure separation of concerns and scalability. It includes the following layers:

Layers

User Interface Layer:

- Provides the front-end experience for users (applicants and administrators).
- Developed using modern frameworks like ReactJS or NextJS.
- Handles user input, application status viewing, and notifications.

Middle Tier:

- Manages business logic and acts as a bridge between the user interface and the database.
- Built using NodeJS/ExpressJS for server-side processing.
- Implements APIs for user authentication, data submission, and updates.

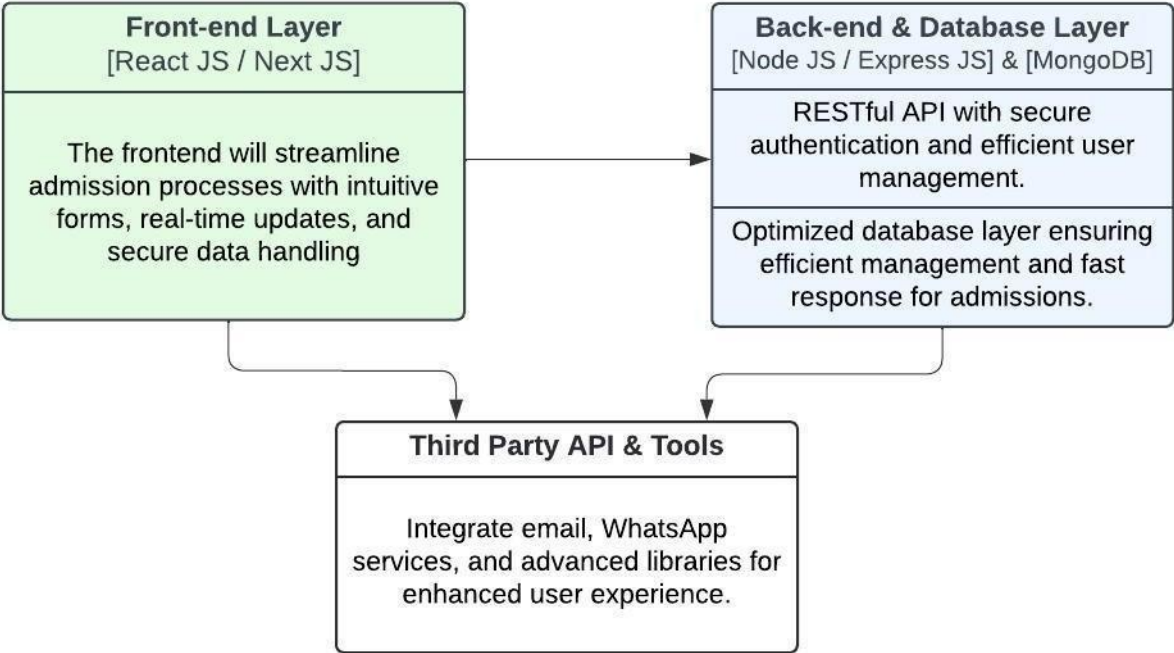
Data Access Layer:

- Handles interactions with the database.
- Ensures secure and optimized queries for operations like application submission, fee verification, and test scheduling.
- Uses MongoDB for scalability and flexibility in handling structured and unstructured data.

Interactions Between Layers

- User Interface Layer interacts with Middle Tier via REST APIs to submit data and retrieve status updates.
- Middle Tier communicates with the Data Access Layer for CRUD operations on the database.
- Middle Tier integrates with external systems (e.g., payment gateways, notification services) through APIs.

Automated Platform for Undergraduate Admissions	Version:1.0
Software Design Specifications	Date:13-01-2025
<document identifier>	



Automated Platform for Undergraduate Admissions	Version:1.0
Software Design Specifications	Date:13-01-2025
<document identifier>	

6 Design Strategy

The design strategy for the "Automated Platform for Undergraduate Admissions" focuses on achieving scalability, modularity, security, and user-centric functionality. The strategy emphasizes creating a robust system architecture that supports current needs while allowing for future enhancements.

Future System Extension or Enhancement

- **Modular Design:** The system is decomposed into independent modules, such as User Management, Application Management, and Payment Processing, enabling easy addition or modification of features.
- **Scalability:** Cloud-based technologies (e.g., Firebase, MongoDB) ensure the system can handle increased workloads as the number of users and applications grows.
- **Integration Capability:** The architecture supports seamless integration with external systems, such as new payment gateways or notification services.

System Reuse

- **Reusable Components:** Core functionalities like user authentication, data validation, and API integrations are developed as reusable components, minimizing development effort for similar projects.
- **Shared Libraries:** The use of widely adopted frameworks (ReactJS, NodeJS) promotes reusability across different teams and applications.
- **Database Models:** Standardized database schemas ensure compatibility for future projects requiring similar data handling.

User Interface Paradigms

- **Responsive Design:** The system employs a responsive web design approach, ensuring compatibility across devices (desktops, tablets, and smartphones).
- **User-Centric Interface:** Built with modern frameworks like ReactJS, the interface provides an intuitive, user-friendly experience for both applicants and administrators.
- **Role-Based Views:** Different user roles (e.g., applicant, administrator) are assigned customized dashboards, displaying only relevant information and functionalities.

Automated Platform for Undergraduate Admissions	Version:1.0
Software Design Specifications	Date:13-01-2025
<document identifier>	

Data Management

- **Centralized Database:** MongoDB serves as the primary database for storing application data, ensuring consistency and easy management.
- **Data Security:** Sensitive data is encrypted both at rest and in transit, adhering to compliance standards for data protection.
- **Scalable Storage:** The use of a NoSQL database (MongoDB) allows for dynamic scaling as the volume of data increases, supporting efficient query performance.
- **Backup and Recovery:** Regular data backups and recovery protocols are implemented to prevent data loss and ensure business continuity.

Concurrency and Synchronization

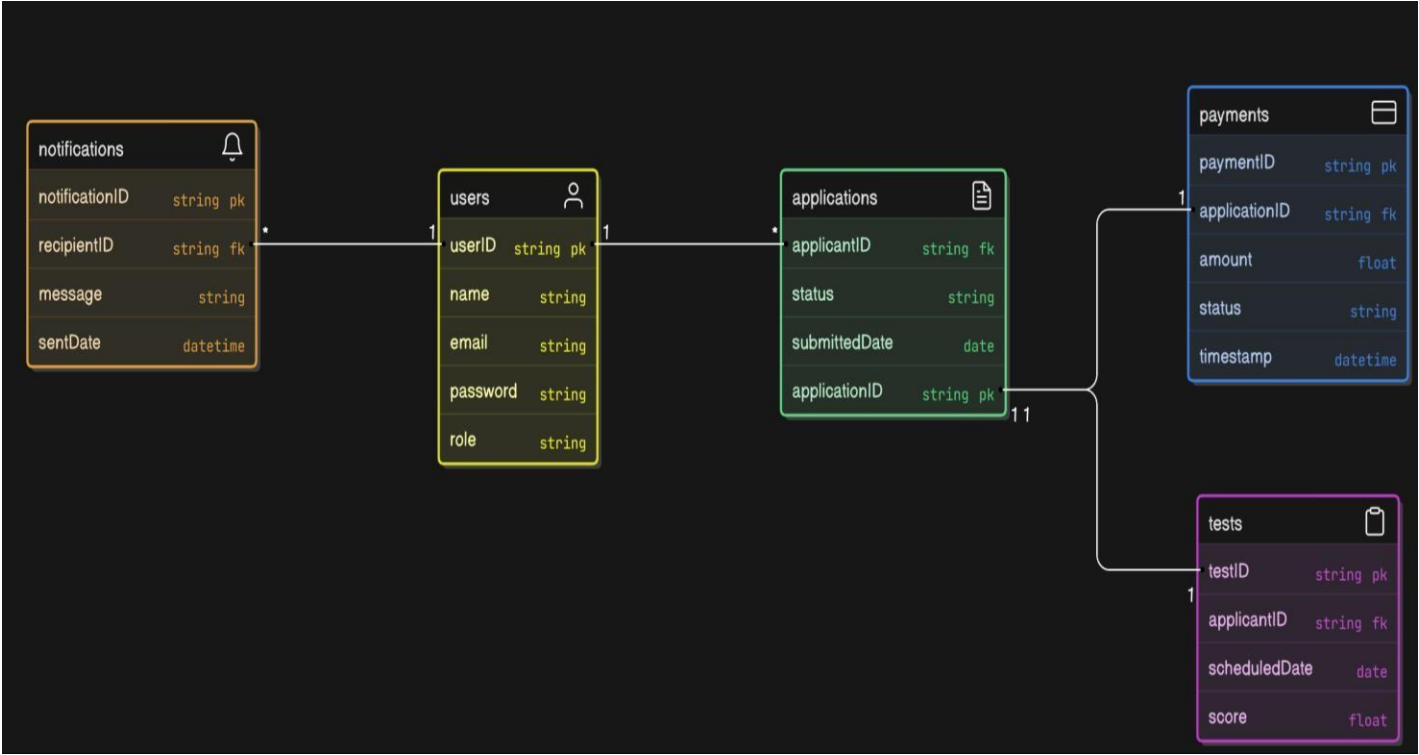
- **Asynchronous Operations:** APIs and database queries are designed to handle asynchronous operations, reducing system bottlenecks and improving performance.
- **Queue Management:** Task queues (e.g., for payment verification or test scheduling) are implemented to handle high-traffic scenarios efficiently.
- **Concurrency Control:** Optimistic concurrency control mechanisms are used to prevent data conflicts when multiple users interact with the system simultaneously.
- **Scalable Resources:** Load balancing is employed to manage concurrent user requests, ensuring consistent performance during peak usage.

Reasoning and Trade-Offs

- **Reasoning:** The system design reflects the project goals of accuracy, scalability, and user satisfaction. A modular architecture allows the system to evolve over time, while a layered approach ensures separation of concerns.
- **Trade-Offs:**
 - Using modern technologies like ReactJS and MongoDB prioritizes scalability and development speed but may require additional training for new developers.
 - Cloud hosting increases flexibility and performance but involves recurring costs compared to on-premises solutions.

6.1 Database Design

6.1.1 ER Diagram



Automated Platform for Undergraduate Admissions	Version:1.0
Software Design Specifications	Date:13-01-2025
<document identifier>	

6.1.2 Data Dictionary

6.1.2.1 Data 1

USER						
Name	User					
Alias	-					
Where-used/how-used	Used by Login, Registration, Application Submission, Notification, Admin functions. Role determines access rights.					
Content description	Represents a person in the system who can either be an applicant or an admin. Stores personal information and credentials.					
Column Name	Description	Type	Length	Null able	Default Value	Key Type
userID	Unique identifier for the user	Integer	10	No	None	PK
name	Full name of the user	Varchar	255	No	None	
email	User's email address for notifications.	Varchar	255	No	None	
password	Encrypted password for the user.	Varchar	255	No	None	
role	Role of the user (e.g., Applicant, Admin).	Varchar	50	No	None	

6.1.2.2 Data 2

Application						
Name	Application					
Alias	App					
Where-used/how-used	Used by User for submission, Admin for status updates. Stored in the system after submission.					
Content description	Represents a user's application for a program, containing status, submission date, and applicant information.					
Column Name	Description	Type	Length	Null able	Default Value	Key Type
applicationID	Unique identifier for	Integer	10	No	None	PK

Automated Platform for Undergraduate Admissions	Version:1.0
Software Design Specifications	Date:13-01-2025
<document identifier>	

	the application.					
applicantID	Reference to the user applying.	Integer	10	No	None	PK
status	Current status of the application (e.g., Pending).	Varchar	50	No	None	
submittedDate	Date of the application submission.	Date	-	No	None	

6.1.2.3 Data 3

Payment						
Name	Payment					
Alias	-					
Where-used/how-used	Used by Application Process to verify payments and handle refunds.					
Content description	Represents the payment made for an application, including the amount, payment status, and timestamp of the transaction.					
Column Name	Description	Type	Length	Null able	Default Value	Key Type
paymentID	Unique identifier for the payment.	Integer	10	No	None	PK
applicationID	Reference to the linked application for payment.	Integer	10	No	None	PK
amount	Payment amount.	Decimal	10,2	No	None	
status	Payment status (e.g., Paid, Pending).	Varchar	50	No	None	

Automated Platform for Undergraduate Admissions	Version:1.0
Software Design Specifications	Date:13-01-2025
<document identifier>	

timestamp	Timestamp of the payment transaction	Timestamp	-	No	None	
-----------	--------------------------------------	-----------	---	----	------	--

6.1.2.4 Data 4

Test						
Name	Test					
Alias	-					
Where-used/how-used	Used to schedule and record results for applicants. Linked with applications.					
Content description	Represents a test scheduled for an applicant, including date, score, and test results.					
Column Name	Description	Type	Length	Null able	Default Value	Key Type
testID	Unique identifier for the test.	Integer	10	No	None	PK
applicantID	Reference to the user taking the test.	Integer	10	No	None	FK
scheduledDate	Date of the test.	Date	-	No	None	
score	Test score for the applicant.	Decimal	5,2	Yes	NULL	

6.1.2.5 Data 5


Notification						
Name	Notification					
Alias	-					
Where-used/how-used	Used to send updates to users about application status, tests, payments, etc.					
Content description	Represents a notification sent to a user, either by email or SMS, containing important updates.					
Column Name	Description	Type	Length	Null able	Default Value	Key Type
notificationID	Unique identifier for the notification.	Integer	10	No	None	PK

Automated Platform for Undergraduate Admissions	Version:1.0
Software Design Specifications	Date:13-01-2025
<document identifier>	

recipientID	Reference to the user receiving the notification.	Integer	10	No	None	FK
message	Content of the notification.	Text	-	No	None	
sentDate	Date and time the notification was sent.	Timestamp	-	No	None	

6.2 GUI Design

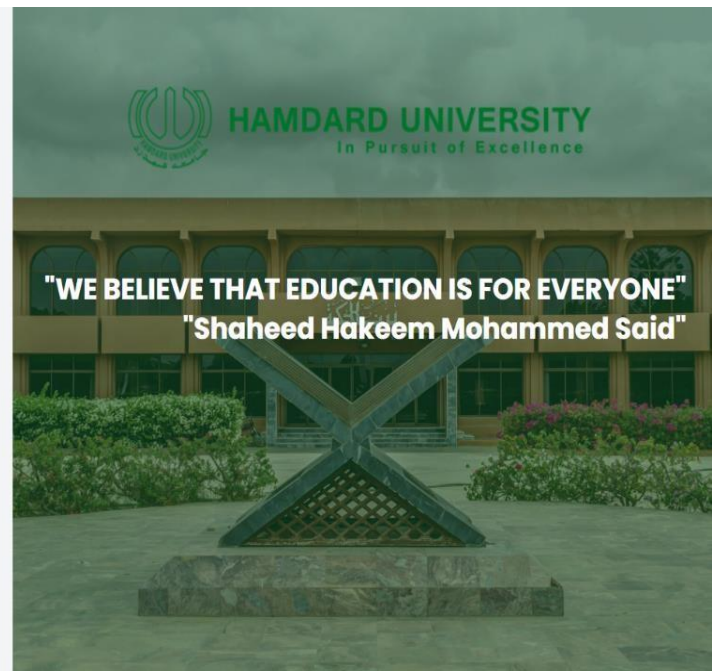
6.2.1


Welcome to Hamdard University Admissions for
Fall-2024

Admission Portal

☐ Karachi ☐ Islamabad

Register



Automated Platform for Undergraduate Admissions	Version:1.0
Software Design Specifications	Date:13-01-2025
<document identifier>	

6.2.2

General Info Program Preferences Qualification Info Voucher Detail Change Password

General Info

Student Name

Father Name

Student NIC

Student Date of Birth

dd----yyyy

Student Cell No

Student Email

Applicant ID

Test Center

Karachi

Gender

Female

How did you hear about us

Friend

Student Province

Sindh

Address

Upload Picture

Choose File

No file chosen

Submit

6.2.3

General Info Program Preferences Qualification Info Voucher Detail Change Password

Program Preferences

Campus

North Nazimabad KDA Campus

Faculty

Faculty of Engineering

1st Preference (Value Required)

BS (RIS)

2nd Preference

BS (DFCS)

3rd Preference

BS (RIS)

4th Preference

MSCS (SE)

5th Preference

MSCS

Submit

Automated Platform for Undergraduate Admissions	Version:1.0
Software Design Specifications	Date:13-01-2025
<document identifier>	

6.2.4

ces
Qualification Info
Voucher Detail
Change Password

ail

Subj

Scier

Scier

nt Awaited

Add Qualification

Qualification

Subject

Obtained Marks

Total Marks

CGPA

Board / University

Passing Year

Status

Cancel

Save

6.2.5

General Info
Program Preferences
Qualification Info
Voucher Detail
Change Password

Qualification Detail

Add Qualification

Qualification	Subject	Obtained Marks	Total Marks	CGPA	Board / University	Passing Year	Status	Actions
Intermediate/Equivalent	Science	489	550	-	Hyderabad board	2024	Result/Document Awaited	<div>Edit</div> <div>Delete</div>
MATRIC	Science	493	550	-	Hyderabad board	2021	Passed	<div>Edit</div> <div>Delete</div>

6.2.6

General Info
Program Preferences
Qualification Info
Voucher Detail
Change Password

Voucher Detail

Voucher No	Account Title	Account	Amount	Status	Issue Date	Due Date	Edit	Print
12345	John Doe	Savings	\$500	Pending	2024-01-01	2024-02-01	<div>Edit</div>	<div>Print</div>
67890	Jane Smith	Current	\$300	Paid	2024-01-15	2024-02-15	<div>Edit</div>	<div>Print</div>

Automated Platform for Undergraduate Admissions	Version:1.0
Software Design Specifications	Date:13-01-2025
<document identifier>	

6.2.7

General Info Program Preferences Qualification Info Voucher Detail Change Password

Change Password

Old Password

New Password

Confirm Password

Change Password

6.2.8

App ID	Inquiry Date	Student Name	Status	Project	Faculty	Program	Offer Type	Cell No	Edit	Offer Letter Amount
129175	09-Dec-2024	Safia Ahmed	No Voucher	NNKDA	FEST	BS (RIS)	Spring-2025	03312321838	Edit	52,500
129162	08-Dec-2024	Rehman	Unpaid Voucher	NNKDA	FEST	BS (DFCS)	Spring-2025	03150212162	Edit	52,500
129162	08-Dec-2024	Abdul	Unpaid Voucher	NNKDA	FEST	BS (DFCS)	Spring-2025	03150212162	Edit	52,500
129162	08-Dec-2024	Zoya Saeed	Unpaid Voucher	NNKDA	FEST	BS (DFCS)	Spring-2025	03150212162	Edit	52,500
129162	08-Dec-2024	Ahmed	Unpaid Voucher	NNKDA	FEST	BS (DFCS)	Spring-2025	03150212162	Edit	52,500
129162	08-Dec-2024	Dua Rahim	Unpaid Voucher	NNKDA	FEST	BS (DFCS)	Spring-2025	03150212162	Edit	52,500

6.2.9

Offer Letter	Father Name	Student Email	Verification Code	Last Updated	Student Province	Print	Reset Password
First Enter Offer Letter Amount then Print Offer Letter	Nazim Ahmed	safia.a2005@gmail.com	9HRC	2024-12-09 (00:00)	6	Print	Reset Password
First Enter Offer Letter Amount then Print Offer Letter	Muhammad Saleem	firefoxbpo@gmail.com	MEU7	2024-12-08 (00:00)	6	Print	Reset Password
First Enter Offer Letter Amount then Print Offer Letter	Syed Shahid Hussain	ishfatabool2468@gmail.com	5KL4	2024-12-08 (00:00)	6	Print	Reset Password

Automated Platform for Undergraduate Admissions	Version:1.0
Software Design Specifications	Date:13-01-2025
<document identifier>	

7 References

1. <https://www.creatrixcampus.com/student-admission-management-system>
2. <https://heiaply.com/education-admissions/admissions-software-higher-education>
3. <https://scrumguides.org>
4. <https://lucidchart.com/blog/sequence-diagram-tutorial>
5. <https://lucidchart.com/blog/sequence-diagram-tutorial>

Automated Platform for Undergraduate Admissions	Version:1.0
Software Design Specifications	Date:13-01-2025
<document identifier>	

8 Appendices

Appendix A – User Roles and Permissions Matrix

Role	Permissions
Applicant	Register, Submit Application, Track Status, Make Payments, Receive Updates
Admin	Review Applications, Update Status, Schedule Tests, Generate Reports
Super admin	Full control over user management, system settings, and auditing functions

Appendix B – API Endpoint Overview

Endpoint	Method	Description
/api/register	POST	Register a new user
/api/login	POST	Authenticates user credentials
/api/applications	GET	Retrieves user’s application details
/api/payment/verify	POST	Verifies payment made via gateway
/api/tests/schedule	POST	Schedules test for eligible applicants

Appendix C – Technologies Used

Technology	Purpose
ReactJS	Frontend development
NodeJS+Express	Backend services and API creation
MongoDB	Database management
Firebase Hosting	Web hosting and deployment
GitHub	Version control and collaboration

Automated Platform for Undergraduate Admissions	Version:1.0
Software Design Specifications	Date:13-01-2025
<document identifier>	

Appendix D – Sample Test Notification

To: applicant123@gmail.com

Subject: Admission Test Schedule Notification

Message:

Dear Applicant,

Your admission test has been scheduled for **March 15, 2025** at **10:00 AM**.

Please log in to your portal to access further instructions.

Thank you.

Hamdard University Admissions Team