Software Requirements Specification (SRS)

For: College Admission Management Portal

Prepared by: Group03

Date: 22nd September, 2025

1. Introduction

1.1 Purpose

The purpose of this Software Requirements Specification (SRS) document is to define the functionalities, constraints, and design considerations for the **College Admission Management Portal (CAP)**.

The CAP is intended to simplify and digitise the **end-to-end college admission process**, from applicant registration and application submission to verification, merit list generation, and admission finalisation.

This SRS aims to:

- Provide a **clear definition** of system behaviour for developers.
- Establish a **common understanding** between stakeholders, including students, faculty, administrators, and technical teams.
- Serve as a **foundation for system design, development, and testing**.
- Act as a **contract** that ensures the final implementation aligns with the stated requirements.

By documenting both **functional requirements (FRs)** and **non-functional requirements (NFRs)**, as well as use cases and data flow diagrams (DFDs), this SRS ensures that CAP is developed to be **user-friendly, scalable, secure, and reliable**.

1.2 Scope

The **College Admission Management Portal (CAP)** is a **web-based application** that integrates all activities related to student admissions into a **single, centralized system**.

Key Features & Functionalities:

1. **Applicant Services:**
    - Online registration and login.
    - Filling and editing of admission forms.
    - Uploading required documents (mark sheets, ID proofs, certificates).
    - Selecting desired courses/programs.
    - Payment of admission and application fees via payment gateway.
    - Receiving real-time notifications and updates.
    - Downloading merit lists, admission results, and admission letters.
2. **Admission Officer Services:**
    - Verification of applicant details and documents.
    - Requesting additional or corrected documents.
    - Scheduling interviews or entrance exams.
    - Updating application status (accepted, rejected, under review).

3. **Faculty Reviewer Services:**
    - Reviewing academic profiles and documents.
    - Conducting interviews or reviewing entrance exam performance.
    - Providing feedback for inclusion in merit list generation.
4. **Administrator Services:**
    - Monitoring system operations.
    - Approving and publishing merit lists and final results.
    - Configuring admission policies, eligibility criteria, and quotas.
    - Generating analytical reports for management decisions.
5. **External Systems Integration:**
    - **Payment Gateway**: To handle fee transactions securely.
    - **Email/SMS Service**: To notify applicants regarding confirmation, schedule, and results.

Benefits of the System:

- **Efficiency:** Reduces manual work and eliminates paperwork.
- **Transparency:** Provides applicants real-time visibility into their application status.
- **Scalability:** Supports thousands of applicants simultaneously.
- **Accessibility:** Enables remote access through web browsers and mobile devices.
- **Accuracy:** Automates sorting, merit generation, and reduces errors.

1.3 Definitions, Acronyms, and Abbreviations

- **CAP**: College Admission Management Portal.
- **SRS**: Software Requirements Specification.
- **DFD**: Data Flow Diagram.
- **FR**: Functional Requirement.
- **NFR**: Non-Functional Requirement.
- **Use Case**: Interaction between an actor and the system to achieve a goal.
- **Applicant**: Student applying for admission.
- **Admission Officer**: College staff verifying and processing applications.
- **Faculty Reviewer**: Academic staff reviewing applicants’ academic and interview performance.
- **Administrator**: Senior authority managing system settings and approvals.
- **Payment Gateway**: External system handling monetary transactions.
- **Email/SMS Service**: External service provider for communication with applicants.

- **Merit List**: List of shortlisted applicants generated based on academic and institutional criteria.

1.4 References

1. IEEE Std 830-1998 – IEEE Recommended Practice for Software Requirements Specifications.
2. Unified Modeling Language (UML) Specification, Version 2.5.
3. Pressman, R. S. *Software Engineering: A Practitioner's Approach*.
4. Sommerville, I. *Software Engineering* (10th Edition).
5. UGC and AICTE guidelines for admission processes in India.
6. Case studies from existing portals like DU Admissions, JEE Main, and NEET Counseling systems.

1.5 Overview

This document is structured to provide both a **high-level overview** and **detailed technical specifications** of CAP.

- **Section 2: General Description** – Outlines the overall product perspective, major functions, user types, and constraints.
- **Section 3: Specific Requirements** – Presents detailed descriptions of functional and non-functional requirements.
- **Section 4: Use Cases** – Models user interactions with CAP, highlighting <<include>> and <<extend>> relationships.
- **Section 5: Appendices** – Contains supporting diagrams (DFD Level 0, 1, and 2), glossary, and references.

The purpose is to ensure the system is described in a manner that is **clear, consistent, and comprehensive**, so that **stakeholders and developers** can collaborate effectively and align their expectations with system capabilities.

2. General Description

2.1 Product Perspective

The **College Admission Management Portal (CAP)** is designed as a **web-based multi-user system** that integrates seamlessly with existing college IT infrastructure.

- **System Type:** Centralized admission management system.
- **Architecture:** Follows a **3-tier architecture**:
    1. **Presentation Layer (Front-End)** – Web interface for applicants, faculty, admission officers, and administrators.
    2. **Application Layer (Back-End Services)** – Handles admission workflows, verification, merit generation, notifications, and reporting.
    3. **Database Layer** – Stores applicant details, documents, merit lists, results, and system logs.
- **External Interfaces:**
    - **Payment Gateway Integration** for secure fee transactions.
    - **Email/SMS Gateway** for sending admission notifications.
    - **Identity Verification (optional future scope)** via Aadhaar/UID system.
- **Relationship with Other Systems:**
    - If the college already uses **ERP** for student management, CAP can integrate for smooth data transfer.
    - Can provide APIs to share admission data with **Government/University central systems**.

2.2 Product Functions

At a high level, CAP provides the following **functional modules**:

1. **Applicant Module**
   - Register and create login credentials.
   - Fill admission form and select desired course(s).
   - Upload necessary documents.
   - Pay admission/application fee.
   - Track status of application.
   - Download admit card, merit list, and admission letter.
2. **Admission Officer Module**
   - Verify applicant information and uploaded documents.
   - Request re-submission of documents (if invalid).
   - Schedule and manage interviews/entrance tests.
   - Update admission status.
3. **Faculty Reviewer Module**
   - Review academic background and scores.
   - Conduct online/offline interviews.
   - Provide feedback and grading for selection.
4. **Administrator Module**
   - Manage admission policies, eligibility criteria, and quotas.
   - Generate and publish merit lists.
   - Approve/reject applications flagged by officers.
   - Monitor system usage and performance.
   - Generate analytical and statistical reports.
5. **External Services**
   - Securely process payments.
   - Send updates and reminders via SMS/email.

2.3 User Characteristics

Different categories of users will interact with CAP:

1. **Applicants (Students):**
   - Age: 17–30 years (primarily school graduates and postgraduates).
   - Technical skills: Basic internet and smartphone/laptop usage.
   - Needs: Simple interface, step-by-step guidance, multi-language support.
2. **Admission Officers (Clerical Staff):**
   - College employees trained in student admissions.
   - Moderate technical knowledge.
   - Needs: Tools for verification, document review, and communication with applicants.
3. **Faculty Reviewers (Professors):**
   - Subject matter experts evaluating candidates.
   - Limited time availability, so need efficient dashboards.
   - Needs: Quick access to applicant profiles, marks, and interview tools.
4. **Administrators (Management Staff):**
   - Senior officials in charge of admissions.
   - Strong decision-making authority.
   - Needs: System-wide visibility, report generation, and configuration access.

2.4 Constraints

1. **Technical Constraints**
   - Web application must be **cross-platform** (desktop, tablet, mobile).
   - Database should support **high concurrency** (thousands of applicants).
   - Must comply with **data privacy regulations** (e.g., GDPR, Indian IT Act).
   - Limited internet bandwidth in rural areas → portal should be **lightweight** and optimized.
2. **Business Constraints**
   - Admission policies governed by **university/AICTE/UGC regulations**.
   - Fixed admission deadlines.
   - Limited IT staff for maintenance.
3. **Operational Constraints**
   - Payment gateway failures → must provide retry and offline options.
   - System must support **24/7 availability** during admission season.
   - Backups and disaster recovery must be in place.

2.5 Assumptions and Dependencies

- Applicants will have access to **internet-enabled devices**.
- Payment gateway services (e.g., Razorpay, Paytm, PayPal) are **available and reliable**.
- Email/SMS delivery services (e.g., Twilio, AWS SES) function without downtime.
- College has necessary **IT infrastructure** (server hosting, storage, network bandwidth).
- Admission rules and eligibility criteria are provided by the **college administration** in advance.
- External government systems (if integration required) expose APIs for data sharing.

2.6 Apportioning of Requirements

Some requirements may be postponed for **future releases**:

- Mobile app version (Android/iOS).
- AI-based document verification (OCR + fraud detection).
- Biometric/Aadhaar verification.
- Chatbot for applicant queries.
- Multilingual voice assistance.

3. Specific Requirements

3.1 Functional Requirements (FRs)

The **functional requirements** define the services and behaviors that the College Admission Portal (CAP) must provide.
We will group these by **major system modules**.

3.1.1 Applicant Module

- **FR1 – Applicant Registration and Login**
    - The system shall allow applicants to register using name, email, phone number, and password.
    - The system shall validate uniqueness of email/phone.
    - The system shall allow login using registered credentials.
    - The system shall provide password reset via OTP/email verification.
- **FR2 – Profile Management**
    - Applicants shall update personal information (address, DOB, nationality, category).
    - Applicants shall upload photo and signature.
    - The system shall allow applicants to edit information before final submission.
- **FR3 – Course Selection**
    - The system shall display available courses with eligibility criteria.
    - Applicants shall select one or more courses (as per rules).
    - System shall validate eligibility before submission.
- **FR4 – Document Upload**
    - The system shall allow upload of academic records (10th/12th certificates, ID proof, caste certificate, income certificate, etc.).
    - The system shall accept PDF/JPEG/PNG formats within size limits.
    - Applicants shall preview uploaded documents.
- **FR5 – Fee Payment**
    - The system shall allow applicants to pay application/admission fees via payment gateway.
    - The system shall generate a receipt upon successful payment.
    - The system shall allow retry if payment fails.
- **FR6 – Application Tracking**
    - Applicants shall check real-time status: *submitted, under review, verified, shortlisted, admitted, rejected*.
    - Applicants shall receive SMS/email updates at each stage.
- **FR7 – Download & Print Facilities**

  - Applicants shall download admit card for interviews/exams.
  - Applicants shall download merit list and admission letter.

3.1.2 Admission Officer Module

- **FR8 – Application Verification**
  - Officers shall view submitted applications.
  - Officers shall verify applicant documents.
  - Officers shall mark status as *verified, pending, or rejected*.
- **FR9 – Communication with Applicants**
  - Officers shall send re-upload requests for invalid documents.
  - Officers shall send interview/exam schedules.
- **FR10 – Interview/Exam Management**
  - Officers shall create interview/exam slots.
  - The system shall auto-assign applicants to available slots.
  - The system shall notify applicants of their schedule.

3.1.3 Faculty Reviewer Module

- **FR11 – Applicant Review**
  - Faculty reviewers shall view applicant academic background.
  - Faculty reviewers shall mark evaluation scores/grades.
  - Faculty reviewers shall provide interview feedback.

3.1.4 Administrator Module

- **FR12 – Admission Policy Management**
  - Administrators shall configure eligibility criteria.
  - Administrators shall set seat quotas (e.g., category-wise, course-wise).
- **FR13 – Merit List Generation**
  - The system shall generate provisional merit lists based on academic weightage, test scores, and interview performance.
  - Administrators shall approve or reject generated lists before publication.
- **FR14 – Admission Confirmation**
  - Administrators shall confirm final admission for shortlisted applicants.
  - The system shall auto-generate admission letters.
- **FR15 – Reports and Analytics**

    - Administrators shall generate reports on number of applicants, selected students, fees collected, category-wise admissions, etc.

3.1.5 External Service Integrations

- **FR16 – Payment Gateway Integration**
    - CAP shall integrate with third-party payment gateways.
    - System shall securely store payment transaction IDs.
- **FR17 – SMS/Email Notifications**
    - CAP shall integrate with SMS/email service providers.
    - System shall send notifications for registration, status updates, interview calls, merit list publication, and admission confirmation.

3.2 Non-Functional Requirements (NFRs)

Each **NFR** is linked to one or more functional requirements.

3.2.1 Performance

- NFR-1 *(FR1–FR7)*: The system shall support **10,000+ concurrent users** during peak admission season.
- NFR-2 *(FR3, FR4, FR5)* : The system shall process application form submissions within **3 seconds**.

3.2.2 Reliability

- NFR-3 *(All FRs)* : The system shall provide **99.5% uptime** during admission period.
- NFR-4 *(FR12–FR15)*: Data backups shall be taken **daily** and stored in secure servers.

3.2.3 Security

- NFR-5 *(FR1–FR17)* : All user data shall be **encrypted** at rest and in transit.
- NFR-6: The system shall comply with **Indian IT Act and GDPR** for data privacy.
- NFR-7 (Applicant, Officer, Faculty, Admin): Access control shall ensure role-based permissions .

3.2.4 Usability

- NFR-8 *(FR1–FR7)* : The system shall provide **multi-language support** (English, Hindi, regional languages).
- NFR-9 *(FR1–FR7)* : The user interface shall be **mobile-friendly**.
- NFR-10 *(FR6, FR9)* : The system shall include **helpdesk/chatbot** support for applicants.

3.2.5 Maintainability

- NFR-11:The system shall be designed using **modular architecture**.
- NFR-12 : Updates shall not affect running admissions.
- NFR-13 : Logs and error reports shall be automatically generated.

3.2.6 Scalability

- NFR-14 *(All FRs)* : The system shall allow scaling of servers to handle increased load.
- NFR-15 *(FR3–FR15)* : The database shall be optimized for millions of records.

3.2.7 Interoperability

- NFR-16 *(FR12–FR15)* : CAP shall integrate with existing **College ERP**.
- NFR-17 *(FR12–FR15)* : CAP shall provide **APIs for government education boards/universities**.

3.3 Interface Requirements

3.3.1 User Interface Requirements (UI)

The user interface (UI) defines how **applicants, admission officers, faculty reviewers, and administrators** interact with the system. CAP will have a **web-based responsive interface**.

3.3.1.1 General UI Guidelines

- **Consistency:** All pages shall follow a consistent design with the college logo, color scheme, and navigation.
- **Accessibility:** The portal shall support screen readers and comply with **WCAG 2.1 guidelines**.
- **Responsive Design:** Interface shall work on **desktop, tablet, and mobile browsers**.
- **Navigation:** Easy-to-understand menus with clear labels for **Applicant, Officer, Faculty, Admin**.
- **Feedback:** Forms and actions shall display success/error messages immediately.
- **Multilingual Support:** English, Hindi, and optionally regional language toggle.

3.3.1.2 Module-Specific UI Requirements

1. **Applicant Module**
    - **Registration/Login Page:** Form fields for email, phone, password; captcha for security.
    - **Dashboard:** Overview of application status, notifications, upcoming interview/exam.
    - **Application Form Page:** Stepwise wizard (personal info → academic info → document upload → course selection).
    - **Payment Page:** Integration with payment gateway; show payment amount, instructions, and receipt.
    - **Notification Panel:** Alerts for document verification, interview schedule, merit list, admission status.
    - **Download Section:** Merit list, admit card, admission letter in PDF format.
2. **Admission Officer Module**
    - **Officer Dashboard:** List of pending/verifying applications, search and filter options.

   - **Application Review Page:** View applicant info, uploaded documents, and mark verification status.
   - **Communication Page:** Option to send messages or document re-upload requests to applicants.
   - **Interview Scheduler:** Drag-and-drop calendar to schedule interviews/exams.
3. **Faculty Reviewer Module**
   - **Reviewer Dashboard:** List of assigned applicants with filter by course/program.
   - **Evaluation Page:** View academic records and documents; add interview feedback and grades.
4. **Administrator Module**
   - **Admin Dashboard:** System overview with applicant statistics, seat allotment, merit list status.
   - **Merit List Generation Page:** Upload weightage criteria, generate provisional lists, approve for publishing.
   - **Reports Page:** Export reports in CSV/PDF format for analytics.

3.3.2 Hardware Interface Requirements

- **Server:** CAP shall run on a **cloud server** or college-hosted server supporting **Linux/Windows** OS.
- **Database Server:** Must support **MySQL/PostgreSQL/Oracle**.
- **Client Devices:** Users may access via **PCs, laptops, tablets, or smartphones** with modern browsers.
- **Network:** 10 Mbps minimum for normal operations; scalable bandwidth for peak admission periods.

3.3.3 Software Interface Requirements

- **Operating System:**
   - Server: Linux (Ubuntu 20.04 or higher) or Windows Server 2019+.
   - Client: Any OS supporting modern web browsers (Windows, macOS, Android, iOS).
- **Web Server:** Apache/Nginx/IIS.
- **Database:** MySQL/PostgreSQL/Oracle.
- **Frameworks:** React.js/Angular/Vue.js for frontend, Node.js/Django/Java Spring Boot for backend.
- **Payment Gateway SDKs:** Razorpay/Paytm/Stripe.
- **Email/SMS APIs:** Twilio, SendGrid, or AWS SES.
- **Security:** SSL/TLS 1.3, OAuth2.0 for authentication, AES-256 for sensitive data encryption.

3.3.4 Communication Interface Requirements

- **Internet Access:** Required for applicants, officers, faculty, and admins to access portal.
- **Data Exchange:** JSON/REST APIs for internal modules and external systems (payment, email/SMS).

- **Notification Protocols:** SMTP for email, HTTP API for SMS.
- **Error Handling:** System shall provide clear error messages for failed communication (payment failure, email not sent, network downtime).
- **Session Management:** Secure session cookies and automatic logout after inactivity (15–30 minutes).

3.3.5 Accessibility & Compatibility

- Portal shall comply with **WCAG 2.1 standards** for visually impaired users.
- Portal shall support **latest versions of Chrome, Firefox, Edge, and Safari**.
- PDF documents generated (merit list, admission letter) must be **printable and readable offline**.

4. Appendices

4.1 Glossary

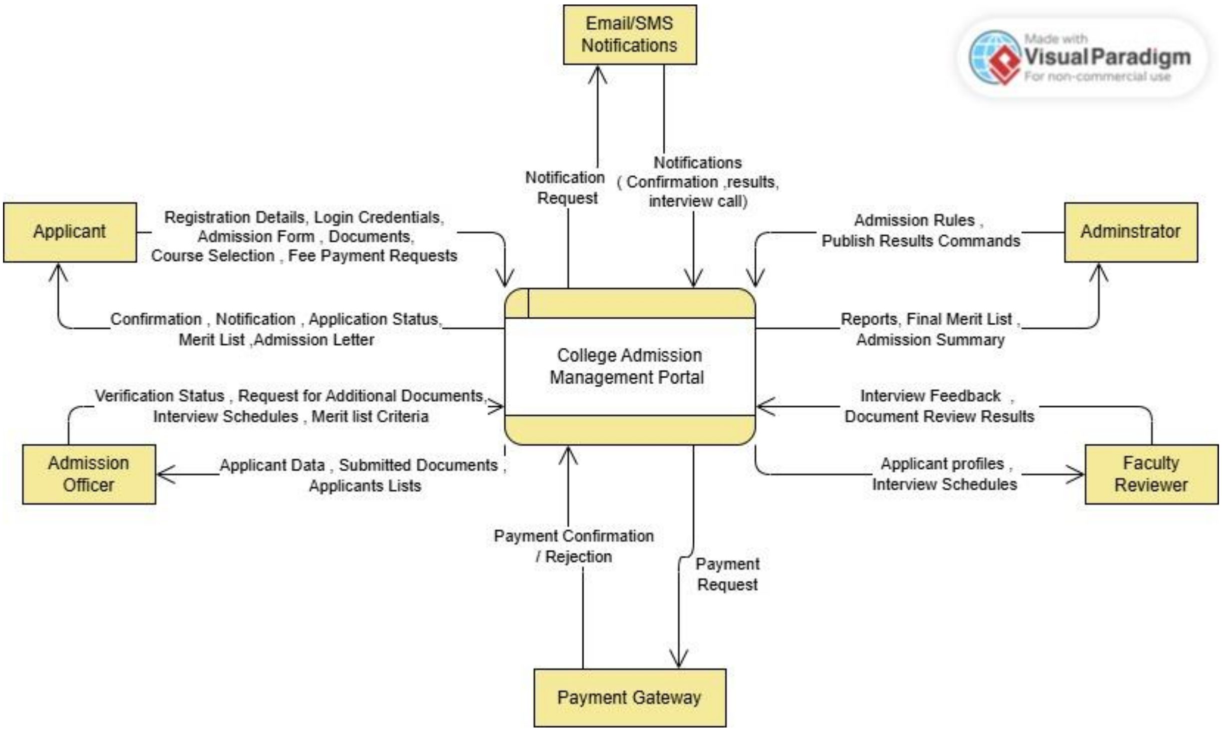
Term / Acronym	Definition / Description
CAP	College Admission Portal – the software system for managing end-to-end college admissions.
SRS	Software Requirements Specification – a document detailing system requirements.
DFD	Data Flow Diagram – graphical representation of data movement through the system.
FR	Functional Requirement – describes what the system must do (behaviors/services).
NFR	Non-Functional Requirement – describes system constraints (performance, security, usability).
Use Case	A specific interaction between an actor and the system to achieve a goal.
Actor	Entity (user or external system) interacting with the system.
Applicant	Student applying for admission through the CAP.
Admission Officer	College staff responsible for verifying applicant documents and status.
Faculty Reviewer	Academic staff responsible for reviewing applicant qualifications and interviews.
Administrator	Senior authority managing CAP settings, policies, and result publication.
Payment Gateway	External service handling fee payments securely (e.g., Razorpay, Paytm).
Email/SMS Service	External system for sending notifications to users.
Merit List	List of shortlisted applicants based on academic and institutional criteria.
Admission Letter	Official document confirming admission of a student.
Eligibility Criteria	Requirements applicants must meet to apply for a specific course/program.
Interview/Exam Schedule	Timetable assigned to applicants for interviews or entrance exams.
Document Verification	Process of checking submitted documents for authenticity and completeness.
Status Update	Notification to the applicant regarding the current stage of application (submitted, verified, shortlisted, rejected, admitted).
OTP (One-Time Password)	Temporary code sent to email/phone for authentication.
UI	User Interface – layout and design of the application front-end.

Term / Acronym	Definition / Description
UX	User Experience – the overall experience of the user interacting with CAP.
REST API	Application programming interface using HTTP for communication between modules/services.
JSON	JavaScript Object Notation – data format for API communication.
SSL/TLS	Secure Sockets Layer / Transport Layer Security – protocols for encrypting data in transit.
AES-256	Advanced Encryption Standard with 256-bit key – used for encrypting sensitive data.
ERP	Enterprise Resource Planning – existing college systems for student, finance, and HR management.
CRUD	Create, Read, Update, Delete – basic database operations.
WCAG 2.1	Web Content Accessibility Guidelines – standards for making web content accessible to people with disabilities.
Load/Concurrency	Number of users the system can handle simultaneously.
Responsive Design	Design approach to make the interface usable on multiple screen sizes.
Admin Dashboard	Portal interface for administrators to manage users, policies, and reports.
Retry Mechanism	Automatic procedure to repeat failed operations, e.g., payment retry.
Provisional Merit List	Temporary list of shortlisted applicants before final approval by admin.
Final Merit List	Approved list of applicants eligible for admission.
Quota Management	Allocation of seats according to categories (General, OBC, SC/ST, etc.).
Admission Confirmation	Final step where applicant confirms seat by paying admission fee and receiving the letter.

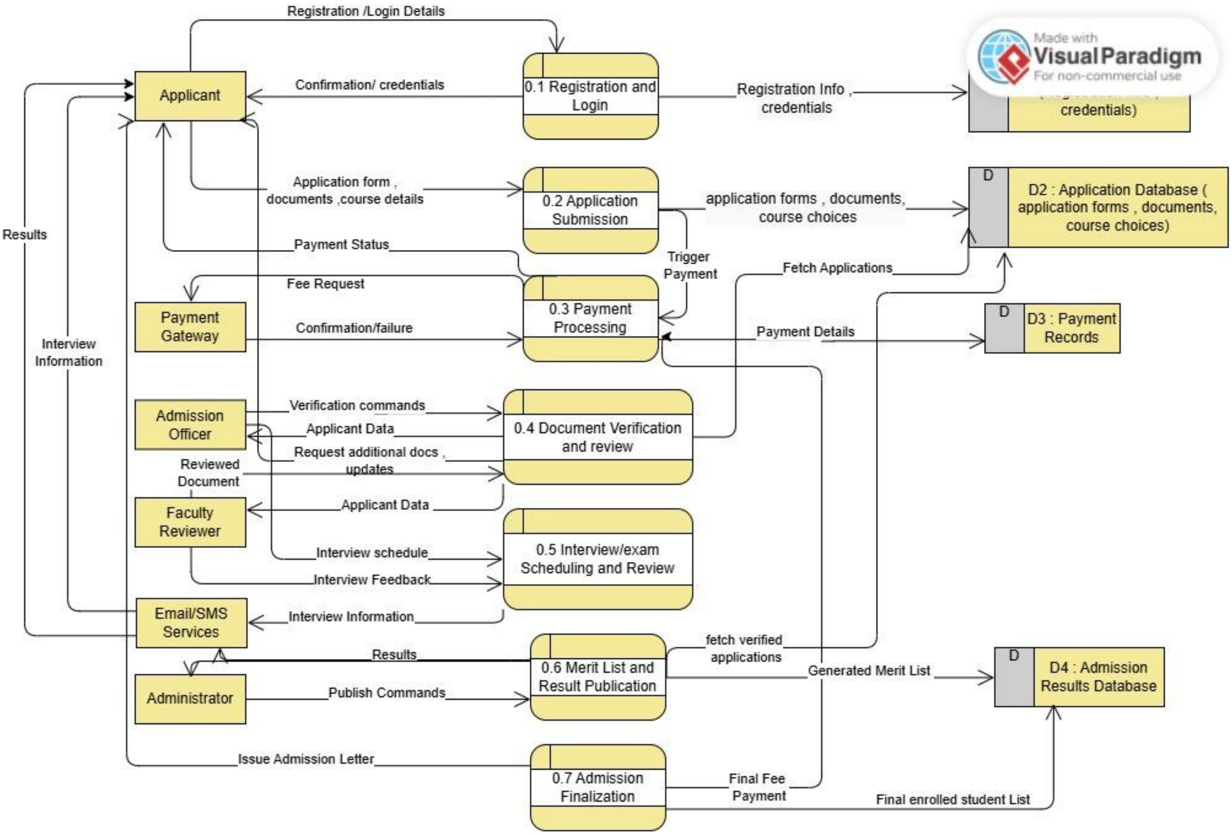
4.2 Supporting Diagrams

Below are the supporting diagrams of the College Admission Portal. These diagrams reinforce the functional and architectural understanding of the system.

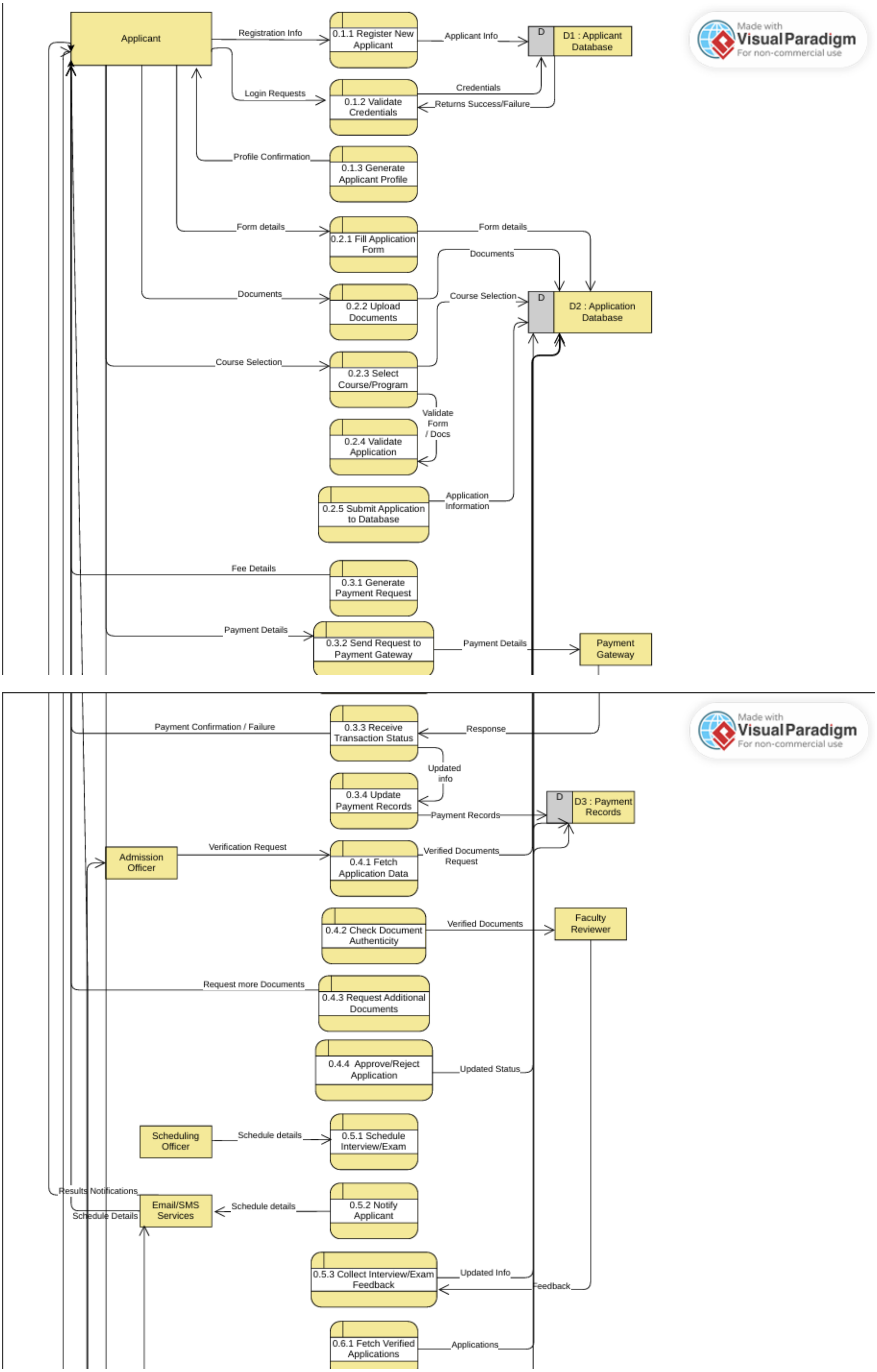
4.2.1 Context Level DFD (Level 0)

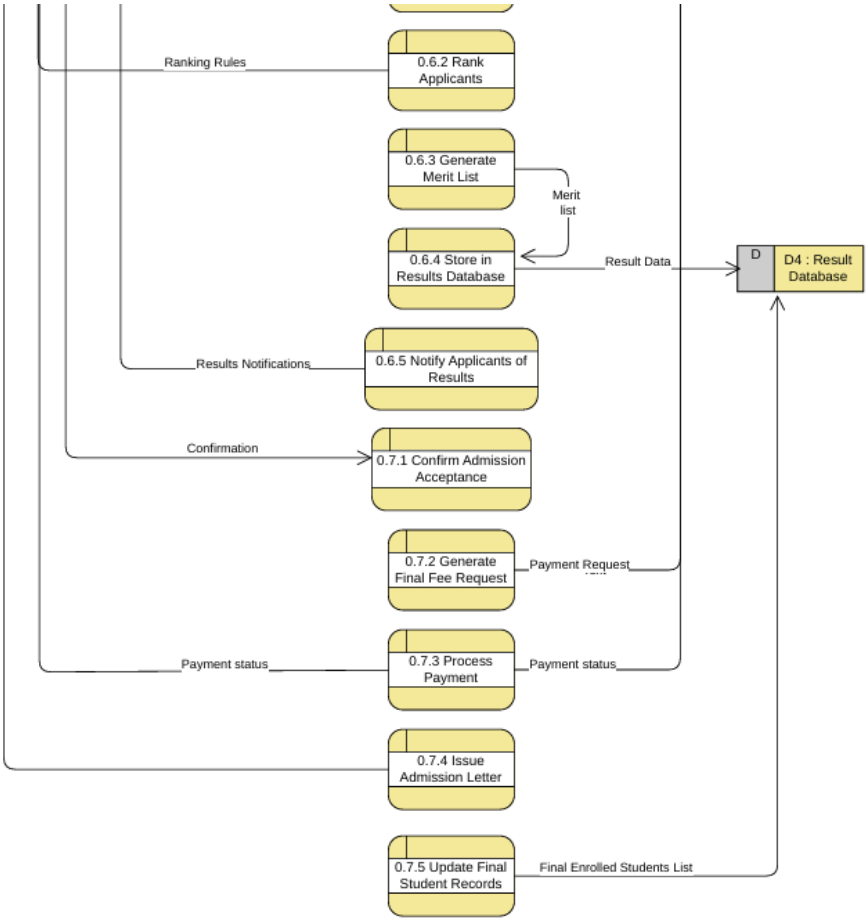

Email/SMS Notifications
Made with Visual Paradigm For non-commercial use
Notification Request
Notifications (Confirmation ,results, interview call)
Applicant
Registration Details, Login Credentials, Admission Form , Documents, Course Selection , Fee Payment Requests
Admission Rules , Publish Results Commands
Adminstrator
Confirmation , Notification , Application Status, Merit List ,Admission Letter
College Admission Management Portal
Reports, Final Merit List , Admission Summary
Verification Status , Request for Additional Documents, Interview Schedules , Merit list Criteria
Interview Feedback , Document Review Results
Admission Officer
Applicant Data , Submitted Documents , Applicants Lists
Applicant profiles , Interview Schedules
Faculty Reviewer
Payment Confirmation / Rejection
Payment Request
Payment Gateway

4.2.2 Context Level DFD (Level 1)

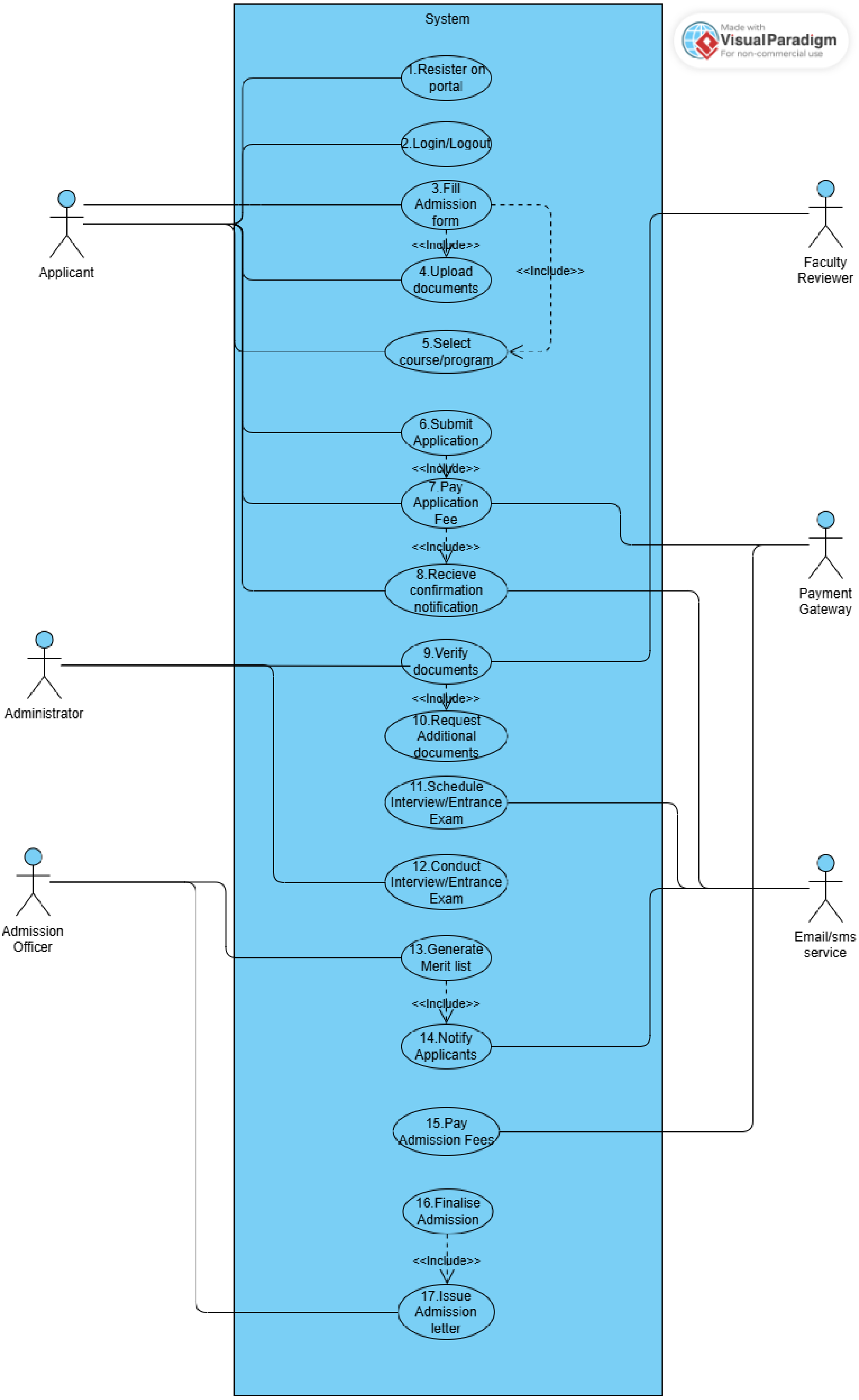

Registration /Login Details
Applicant
Confirmation/ credentials
0.1 Registration and Login
Registration Info , credentials
Made with Visual Paradigm For non-commercial use
credentials)
Results
Application form , documents ,course details
0.2 Application Submission
application forms , documents, course choices
D
D2 : Application Database (application forms , documents, course choices)
Payment Status
Trigger Payment
Fee Request
Fetch Applications
Interview Information
Payment Gateway
Confirmation/failure
0.3 Payment Processing
Payment Details
D
D3 : Payment Records
Admission Officer
Verification commands
Applicant Data
Request additional docs , updates
Reviewed Document
0.4 Document Verification and review
Faculty Reviewer
Applicant Data
Interview schedule
Interview Feedback
0.5 Interview/exam Scheduling and Review
Email/SMS Services
Interview Information
Results
fetch verified applications
Administrator
Publish Commands
0.6 Merit List and Result Publication
Generated Merit List
D
D4 : Admission Results Database
Issue Admission Letter
0.7 Admission Finalization
Final Fee Payment
Final enrolled student List

4.2.3 Context Level DFD (Level 2)

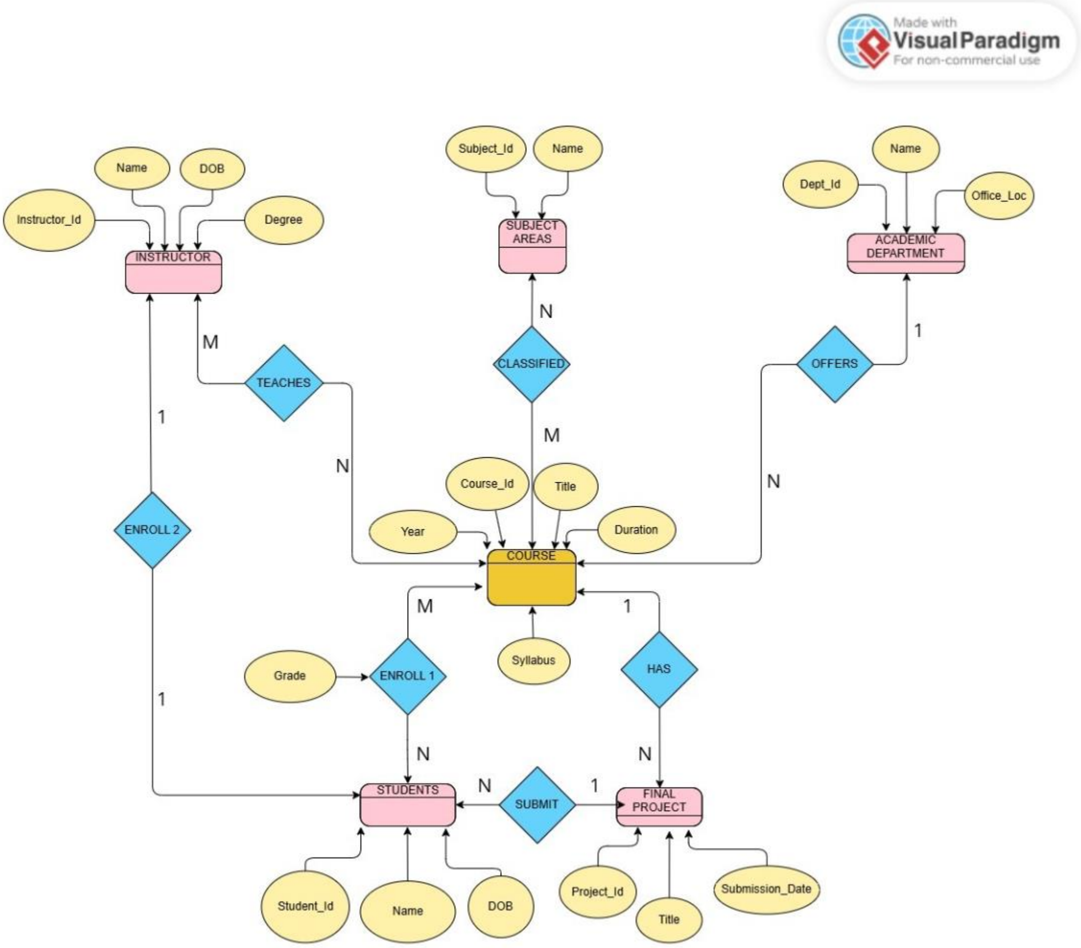

Applicant
Registration Info
0.1.1 Register New Applicant
Applicant Info
D
D1 : Applicant Database
Login Requests
0.1.2 Validate Credentials
Credentials
Returns Success/Failure
Profile Confirmation
0.1.3 Generate Applicant Profile
Form details
0.2.1 Fill Application Form
Form details
Documents
Documents
0.2.2 Upload Documents
Course Selection
D
D2 : Application Database
Course Selection
0.2.3 Select Course/Program
Validate Form / Docs
0.2.4 Validate Application
0.2.5 Submit Application to Database
Application Information
Fee Details
0.3.1 Generate Payment Request
Payment Details
0.3.2 Send Request to Payment Gateway
Payment Details
Payment Gateway
Made with Visual Paradigm For non-commercial use
Payment Confirmation / Failure
0.3.3 Receive Transaction Status
Response
Updated info
0.3.4 Update Payment Records
Payment Records
D
D3 : Payment Records
Admission Officer
Verification Request
0.4.1 Fetch Application Data
Verified Documents Request
0.4.2 Check Document Authenticity
Verified Documents
Faculty Reviewer
Request more Documents
0.4.3 Request Additional Documents
0.4.4 Approve/Reject Application
Updated Status
Scheduling Officer
Schedule details
0.5.1 Schedule Interview/Exam
Results Notifications
Schedule Details
Email/SMS Services
Schedule details
0.5.2 Notify Applicant
0.5.3 Collect Interview/Exam Feedback
Updated Info
Feedback
0.6.1 Fetch Verified Applications
Applications
Made with Visual Paradigm For non-commercial use


Ranking Rules
0.6.2 Rank Applicants
0.6.3 Generate Merit List
Merit list
0.6.4 Store in Results Database
Result Data
D
D4 : Result Database
Results Notifications
0.6.5 Notify Applicants of Results
Confirmation
0.7.1 Confirm Admission Acceptance
0.7.2 Generate Final Fee Request
Payment Request
Payment status
0.7.3 Process Payment
Payment status
0.7.4 Issue Admission Letter
0.7.5 Update Final Student Records
Final Enrolled Students List

4.2.4 Use Case Diagram



4.2.5 ER Diagram (Entity Relationship Model)



4.2.6 Sequence Diagram (Optional – Advanced)

Applicant Submits Admission Application

Actors & Objects:

1. **Applicant** (actor)
2. **CAP Frontend** (UI layer)
3. **CAP Application Server** (backend)
4. **Database** (DB layer)
5. **Payment Gateway** (external system)
6. **Email/SMS Service** (external system)

Flow of Interaction

1. **Applicant → CAP Frontend:** Login
   - Applicant enters credentials.
   - Frontend sends login request to Application Server.
2. **CAP Frontend → Application Server:** Authenticate User

   - Server validates credentials with Database.
   - Server returns login success/failure to Frontend.

3. **Applicant → CAP Frontend:** Fill Admission Form
   - Personal details, academic details, course selection.
4. **CAP Frontend → Application Server:** Submit Form Data
   - Server validates input.
   - Stores data in Database.
5. **Applicant → CAP Frontend:** Upload Documents
   - Upload PDFs/images.
6. **CAP Frontend → Application Server:** Save Documents
   - Server stores document links/metadata in Database.
7. **Applicant → CAP Frontend:** Make Payment
   - Select payment option (Credit Card, UPI, Wallet).
8. **CAP Application Server → Payment Gateway:** Process Payment
   - Payment Gateway processes and returns transaction status.
9. **CAP Application Server → Database:** Update Payment Status
   - Transaction ID, status, timestamp saved.
10. **CAP Application Server → Email/SMS Service:** Send Confirmation
    - Email/SMS sent to applicant confirming submission.
11. **CAP Frontend → Applicant:** Display Confirmation Message
    - “Application submitted successfully. Payment received.”

Index / Table of Contents