

Hamdard University
Department of Computing
Final Year Project



**Dakhla: AI-Assisted University Admissions Platform
(FYP-020/FA25)**

Software Design Specifications

Submitted by
Muhammad Nauman Arif (2514-2022)
Ismail Dad Khan (2927-2022)
Hamza Ali Khan (2508-2022)

Supervisor(s)
Dr. Umer Farooq

Fall 2025

Dakhla: AI-Assisted University Admissions Platform	Version: 1.0
Software Design Specifications	Date: 19/DEC/2025
FYP-020/FA25	

Document Sign off Sheet

Document Information

Project Title	Dakhla: AI-Assisted University Admissions Platform
Project Code	FYP-020/FA25
Document Name	Software Design Specifications
Document Version	1.0
Document Identifier	FYP-020/FA25-SDS
Document Status	Initial Draft
Author(s)	Muhammad Nauman Arif (2514-2022) Ismail Dad Khan (2927-2022) Hamza Ali Khan (2508-2022)
Approver(s)	Dr. Umer Farooq
Issue Date	19/DEC/2025

Name	Role	Signature	Date
Muhammad Nauman Arif (2514-2022)	Team Lead		19/DEC/2025
Ismail Dad Khan (2927-2022)	Team Member 2		19/DEC/2025
Hamza Ali Khan (2508-2022)	Team Member 3		19/DEC/2025
Dr. Umer Farooq	Supervisor		19/DEC/2025
	Co-Supervisor		
	Project Coordinator		

Dakhla: AI-Assisted University Admissions Platform	Version: 1.0
Software Design Specifications	Date: 19/DEC/2025
FYP-020/FA25	

Revision History

Date	Version	Description	Author
19/DEC/2025	1.0	Initial Draft	Hamza Ali Khan

Definition of Terms, Acronyms, and Abbreviations

Term	Description
API	Application Programming Interface - A set of protocols for building software applications
JWT	JSON Web Token - A compact, URL-safe means of representing claims to be transferred between two parties
ORM	Object-Relational Mapping - A programming technique for converting data between incompatible type systems
REST	Representational State Transfer - An architectural style for distributed hypermedia systems
RBAC	Role-Based Access Control - An approach to restricting system access to authorized users
CORS	Cross-Origin Resource Sharing - A mechanism that allows restricted resources to be requested from another domain
PostgreSQL	An open-source relational database management system
Flask	A micro web framework written in Python
React	A JavaScript library for building user interfaces
SQLAlchemy	Python SQL toolkit and Object-Relational Mapping library
Supabase	An open-source Firebase alternative providing PostgreSQL database hosting
Vercel	A cloud platform for static sites and Serverless Functions
Render	A unified cloud platform to build and run apps and websites
HEC	Higher Education Commission - Pakistan's regulatory body for higher education
DakhlaBot	AI-powered chatbot integrated into the Dakhla platform

Dakhla: AI-Assisted University Admissions Platform	Version: 1.0
Software Design Specifications	Date: 19/DEC/2025
FYP-020/FA25	

Table of Contents

Document Information	2
Table of Contents	4
1. Introduction	6
1.1 Purpose of Document	6
1.2 Intended Audience	6
1.3 Document Convention	6
1.4 Project Overview	6
1.5 Scope	7
2. Design Considerations	8
2.1 Assumptions and Dependencies	8
2.2 Risks and Volatile Areas	9
3. System Architecture	11
3.1 System Level Architecture	11
3.2 Frontend Architecture (React SPA)	13
3.3 Backend Architecture (Flask API)	14
3.4 Database Architecture	15
3.5 System Interaction Design	16
4. Design Strategy	17
4.1 Object-Oriented Design Approach	17
4.2 Separation of Concerns	17
4.3 API-First Design	17
4.4 Security-First Design	18
4.5 Scalability Strategy	18
4.6 Data Management Strategy	18
4.7 Concurrency and Synchronization	19
4.8 User Interface Paradigm	19
4.9 Future System Extension	19
4.10 System Reuse	19
5. Detailed System Design	20
5.1 Design Class Diagram	20
5.2 Database Design	30
5.3 Application Design	55

Dakhla: AI-Assisted University Admissions Platform	Version: 1.0
Software Design Specifications	Date: 19/DEC/2025
FYP-020/FA25	

5.3.1 Sequence Diagrams	55
5.3.2 State Diagrams	60
5.4 <i>GUI Design</i>	63
6. References	Error! Bookmark not defined.

Dakhla: AI-Assisted University Admissions Platform	Version: 1.0
Software Design Specifications	Date: 19/DEC/2025
FYP-020/FA25	

1. Introduction

1.1 Purpose of Document

The purpose of this Software Design Specification (SDS) document is to provide a detailed technical blueprint for the "Dakhla - Centralized, AI-Assisted University Admissions Platform." This document translates the requirements outlined in the Software Requirements Specification (SRS) [1] into a concrete architectural and component-level design. It will serve as the primary guide for the development team in implementing the system's modules, interfaces, and data structures.

This project follows an **Object-Oriented design methodology**, leveraging a modern, multi-layered web architecture. The design emphasizes modularity, scalability, and maintainability to support the system's core functionalities.

1.2 Intended Audience

This document is intended for the following personnel:

- **Development Team:** (Muhammad Nauman Arif, Ismail Dad Khan, Hamza Ali Khan) To use as the foundational guide for building, implementing, and integrating the system's components.
- **Project Supervisor:** (Dr. Umer Farooq) To review the technical design, architectural decisions, and ensure the proposed solution is sound and aligns with project objectives.
- **Evaluation Committee:** To assess the technical feasibility, completeness, and engineering soundness of the proposed design during project evaluations.
- **Future Developers:** To understand the system's architecture, data flow, and component interactions for future maintenance, scaling, or enhancements.

1.3 Document Convention

This document uses the **Times New Roman** font. Main headings (Heading 2) are 14pt (Bold), sub-headings (Heading 3) are 12pt (Bold), and all body text is 12pt.

1.4 Project Overview

Dakhla is a web-based, AI-assisted platform designed to centralize and streamline the fragmented university admissions process in Pakistan. The system's core functionality includes a centralized university directory, an AI-powered recommendation engine, an intelligent chatbot (DakhlaBot), the **Dakhla Assessment Module**, and a "DirectApply" feature enabling students to submit a unified application to multiple institutions.

The basic design approach is a **modular, multi-layered architecture** as referenced in the project proposal, ensuring scalability and separation of concerns:

Dakhla: AI-Assisted University Admissions Platform	Version: 1.0
Software Design Specifications	Date: 19/DEC/2025
FYP-020/FA25	

1. **Presentation Layer:** A responsive Single Page Application (SPA) built with **React.js (v18.2.0)** [2] and styled with **Tailwind CSS (v3.3.0)** [3]. This layer handles all user interaction and communicates with the backend via RESTful APIs.
2. **Application Layer:** A service layer that orchestrates business logic. This includes the **Authentication Service**, which uses **JWT (JSON Web Tokens)** [4] for secure, role-based access control (Student, University, Admin), and the **AI Layer**, which is planned for integrating services like OpenAI or Google Gemini for the chatbot, recommendation engine, and exam generation.
3. **Core Services Layer (Backend):** A **Python Flask** [5] backend that exposes secure REST APIs. It uses the **SQLAlchemy ORM** [6] to manage business logic and data operations. This layer is responsible for all core features, user management, assessment logic, and API endpoint provision. Future enhancements include a **Data Collection (Web Scraping Pipeline)**.
4. **Data Layer:** A **PostgreSQL** [7] relational database serves as the single source of truth. It manages all structured data, including user profiles, university listings, programs, application data, and assessment results using a normalized schema for data integrity.

1.5 Scope

The scope of the project defines the boundaries of the system to be developed.

1.5.1 In Scope

- **Centralized University Directory:** A searchable and filterable database of universities, including programs, eligibility criteria, deadlines, and fee structures.
- **Student Account and Dashboard:** Secure user registration, login, and a personalized dashboard to manage academic profiles, saved universities, and application status.
- **AI-Based Recommendation System:** A rule-based engine to suggest suitable universities and programs based on a student's academic background, location, and interests.
- **Intelligent Chatbot (DakhlaBot):** A conversational AI to provide instant responses to FAQs and assist with platform navigation.
- **DirectApply Module:** A single, reusable application form that students can use to apply to multiple universities through the platform.
- **Dakhla Assessment Module:** An AI-generated entrance exam module with automated grading and anti-cheating features.
- **University Dashboard:** A dedicated portal for university representatives to manage their program listings, view, and filter incoming applicant profiles.
- **Admin Panel:** A secure administrative interface for system management, user oversight, and database optimization.
- **Automated Data Scraping:** A backend pipeline using Python (BeautifulSoup, Scrapy) to periodically populate and update university information from public websites.

Dakhla: AI-Assisted University Admissions Platform	Version: 1.0
Software Design Specifications	Date: 19/DEC/2025
FYP-020/FA25	

1.5.2 Out of Scope

The following items are explicitly excluded from the scope of this project:

- **Native Mobile Applications:** No native apps for Android or iOS will be developed. The web platform will be mobile-responsive.
- **Payment Gateway Integration:** The system will not process application fees.
- **Official Database Integration:** No real-time API connections to government or HEC [8] databases for automatic credential verification will be implemented. All academic data will be self-reported by the student.
- **Real-time Video Interviews:** No video conferencing or interview scheduling features.
- **Document Verification Services:** No integration with third-party document verification services.

2. Design Considerations

This section outlines the high-level design decisions, assumptions, and risks that shape the system's architecture.

2.1 Assumptions and Dependencies

This section details assumptions and dependencies relevant to the design phase.

2.1.1 Assumptions

- **Modular Architecture:** The system is designed with a modular, service-oriented structure, featuring a separate React.js SPA frontend and a Python Flask backend API.
- **Stateless Backend:** The Flask backend is stateless. Session management is handled via client-side JWT (JSON Web Tokens), enabling horizontal scaling.
- **SPA Frontend:** The React.js frontend operates as a Single Page Application (SPA), handling all client-side routing and view rendering.
- **Normalized Schema:** The PostgreSQL database utilizes a highly normalized schema to ensure data integrity and minimize redundancy.
- **Cloud-Native Design:** The system is designed for and deployed on a cloud-native infrastructure: the frontend on Vercel, the backend configured for Render, and the database on Supabase.
- **RESTful API Communication:** All communication between frontend and backend follows REST principles using JSON data format.
- **Role-Based Access:** The system implements three distinct user roles: Student, University, and Admin, each with specific permissions and access levels.

Dakhla: AI-Assisted University Admissions Platform	Version: 1.0
Software Design Specifications	Date: 19/DEC/2025
FYP-020/FA25	

2.1.2 Dependencies

- **Third-Party AI APIs:** The DakhlaBot, recommendation, and assessment generation features are dependent on external APIs (e.g., OpenAI [9], Gemini, or Dialogflow). The design must account for API keys, rate limits, and potential latency.
- **Scraping Target Stability:** The data collection module is critically dependent on the HTML structure of the target university websites.
- **Cloud Hosting Environment:** The system is dependent on the services provided by Vercel (Frontend Hosting) [10], Render (Backend Hosting) [11], and Supabase (Managed PostgreSQL Database) [12].
- **Authentication Service:** The system relies on the Flask-JWT-Extended library [13] to implement self-managed, token-based authentication and Role-Based Access Control (RBAC) for Student, University, and Admin roles.
- **Email Service:** Password resets and notifications depend on an external SMTP service (currently configured for Gmail).
- **Core Libraries:** The application is built upon a specific set of libraries, including:
 - **Frontend:** React (18.2.0) [2], React Router, Axios [14], Tailwind CSS (3.3.0) [3], Heroicons
 - **Backend:** Flask, Flask-SQLAlchemy [15], Flask-JWT-Extended, Flask-CORS, Flask-Caching, Python-dotenv
 - **Database:** PostgreSQL via Supabase, SQLAlchemy ORM

2.2 Risks and Volatile Areas

This section discusses the most likely sources of change and risks that impact the system design, along with the mitigation strategies.

Risk 1: Third-Party AI API Volatility (High Risk)

Description: Key features (DakhlaBot, Assessment Module) are dependent on external AI providers (e.g., OpenAI, Gemini). These services pose a risk of changing pricing, rate limits, or API structure.

Design Mitigation: An **Adapter Pattern** [16] will be used to create an abstraction layer between the main application and the external API. This allows the AI provider to be switched with minimal code changes, insulating the core logic from provider-specific volatility.

Dakhla: AI-Assisted University Admissions Platform	Version: 1.0
Software Design Specifications	Date: 19/DEC/2025
FYP-020/FA25	

Risk 2: Web Scraping Target Stability (High Risk)

Description: The automated data collection module is critically dependent on the HTML structure of target university websites, which change frequently and without notice.

Design Mitigation: A **Strategy Pattern** will be implemented. A central "manager" will coordinate scraping tasks, while individual, pluggable "parser" scripts for each university will be developed. This isolates failures and allows a single parser to be updated and redeployed without affecting the entire system. The currently implemented manual data entry system serves as a robust fallback.

Risk 3: University Adoption of "DirectApply" (Medium Risk)

Description: Universities may be slow to adopt the platform and may not actively log in to the University Dashboard to check for new applicants.

Design Mitigation: The "DirectApply" module will be designed with flexible "sinks":

- **Primary Sink:** Save the application data to the PostgreSQL database for viewing on the University Dashboard.
- **Fallback Sink:** Generate a formatted email (or PDF) summary of the application and automatically send it to the university's official admissions email address.

Risk 4: Database Performance at Scale (Medium Risk)

Description: As the number of users and universities grows, database queries may become slow, affecting user experience.

Design Mitigation:

- Implementation of **Flask-Caching** for frequently accessed data
- Database indexing on commonly queried fields
- Query optimization using SQLAlchemy's lazy loading and eager loading strategies
- Potential implementation of read replicas for scaling read operations

Risk 5: Security Vulnerabilities (High Risk)

Description: Web applications are susceptible to various security threats including SQL injection, XSS attacks, and unauthorized access.

Design Mitigation:

- Use of SQLAlchemy ORM to prevent SQL injection
- JWT-based authentication with secure token management
- Input validation and sanitization on both frontend and backend
- CORS configuration to restrict API access
- Password hashing using bcrypt
- Environment variables for sensitive configuration data

Dakhla: AI-Assisted University Admissions Platform	Version: 1.0
Software Design Specifications	Date: 19/DEC/2025
FYP-020/FA25	

3. System Architecture

This section provides a high-level overview of how the functionality and responsibilities of the system are partitioned and assigned to subsystems or components.

3.1 System Level Architecture

The Dakhla platform follows a **three-tier client-server architecture** with clear separation of concerns:

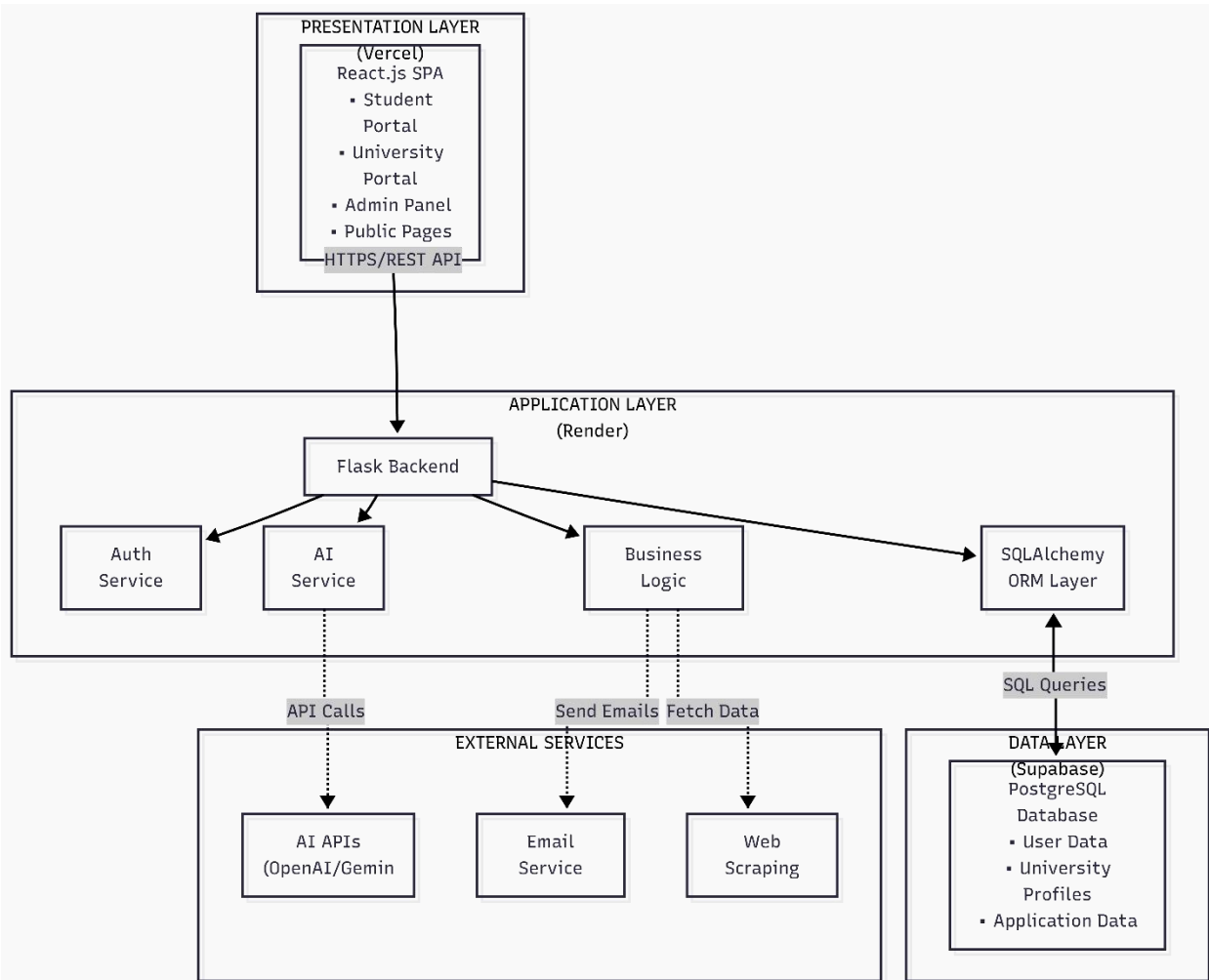


Figure 3.1: System Level Architecture showing the three-tier structure with Presentation Layer (React Frontend on Vercel), Application Layer (Flask Backend on Render), and Data Layer (PostgreSQL on Supabase), along with external services (AI APIs, SMTP, University Websites).

Dakhla: AI-Assisted University Admissions Platform	Version: 1.0
Software Design Specifications	Date: 19/DEC/2025
FYP-020/FA25	

Key Architectural Components:

1. Presentation Layer (Frontend)

- **Technology:** React.js 18.2.0, Tailwind CSS 3.3.0
- **Hosting:** Vercel (CDN-enabled, global distribution)
- **Responsibilities:**
 - User interface rendering
 - Client-side routing
 - Form validation
 - API consumption
 - State management

2. Application Layer (Backend)

- **Technology:** Python Flask, SQLAlchemy ORM
- **Hosting:** Render (scalable cloud platform)
- **Responsibilities:**
 - Business logic execution
 - Authentication & authorization (JWT)
 - API endpoint provision
 - Data validation
 - Integration with external services

Dakhla: AI-Assisted University Admissions Platform	Version: 1.0
Software Design Specifications	Date: 19/DEC/2025
FYP-020/FA25	

3. Data Layer (Database)

- **Technology:** PostgreSQL 12+
- **Hosting:** Supabase (managed database service)
- **Responsibilities:**
 - Data persistence
 - Data integrity enforcement
 - Query optimization
 - Backup and recovery

4. External Services

- AI APIs for chatbot and recommendations
- SMTP service for email notifications
- University websites for data scraping

3.2 Frontend Architecture (React SPA)

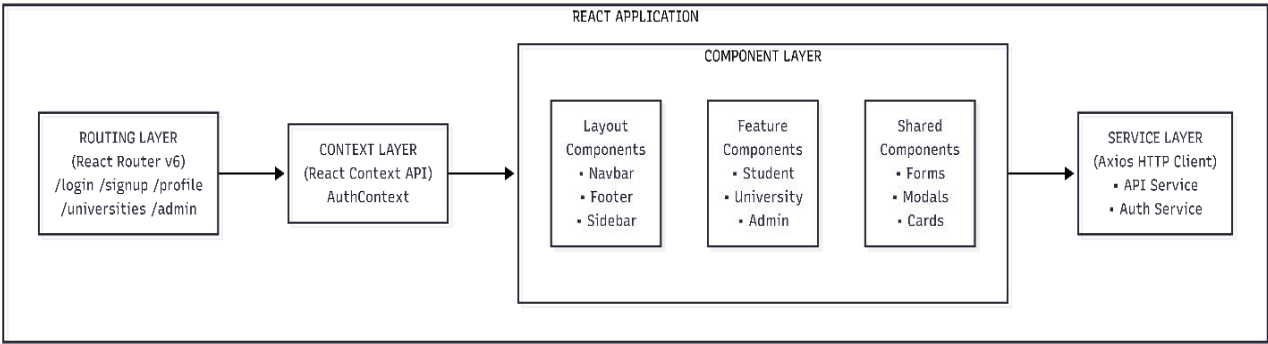


Figure 3.2: React Frontend Architecture showing the layered structure from routing to service layer.

Dakhla: AI-Assisted University Admissions Platform	Version: 1.0
Software Design Specifications	Date: 19/DEC/2025
FYP-020/FA25	

3.3 Backend Architecture (Flask API)

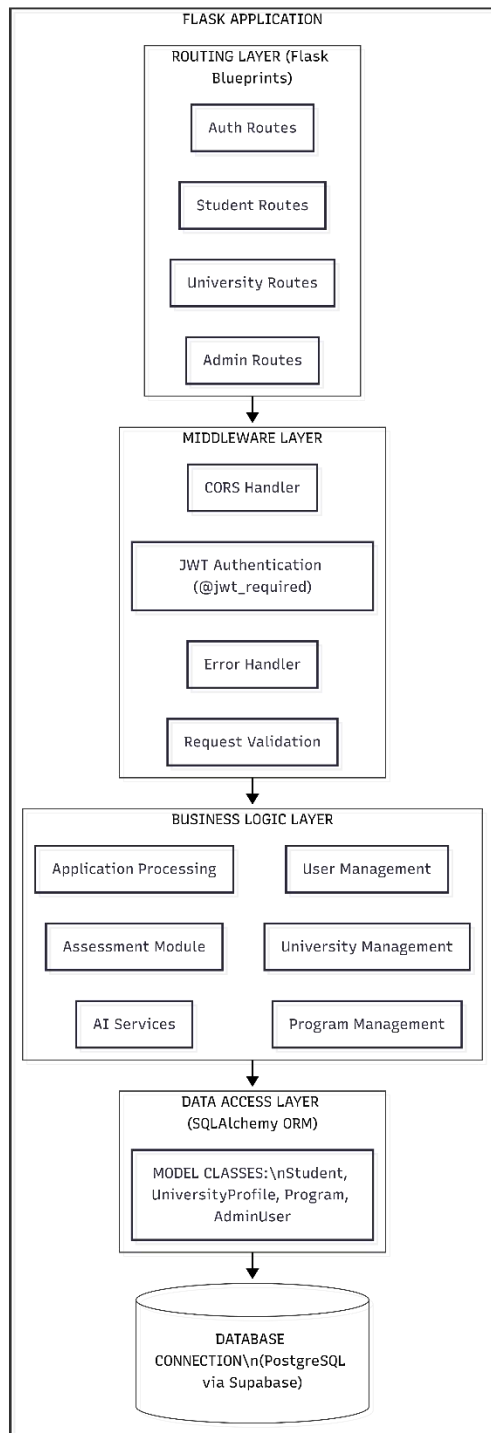


Figure 3.3: Flask Backend Architecture showing the layered structure from routing to database connection.

Dakhla: AI-Assisted University Admissions Platform	Version: 1.0
Software Design Specifications	Date: 19/DEC/2025
FYP-020/FA25	

3.4 Database Architecture

The database follows a **highly normalized relational schema** to ensure data integrity and minimize redundancy. The schema is organized into the following logical groups:

1. User Management Tables

- students_new - Core student records
- university_profiles - Core university records
- admin_users - Administrative users
- user_auth, uni_auth - Authentication data

2. Profile Information Tables

- personal_info - Student personal details
- contact_info, uni_contact_info - Contact information
- address_info, uni_address_info - Address details
- academic_info - Student academic records

3. University Data Tables

- programs - Academic programs offered
- campus_facility - University facilities
- uni_profile_picture, uni_cover_photo - Media assets
- static_universities - Pre-populated university data

4. Lookup/Reference Tables

- countries, cities - Geographic data
- fields_of_study - Academic fields
- degree_types - Degree classifications
- university_types - University classifications
- genders - Gender options

5. Security & Admin Tables

- password_reset_tokens - Password recovery
- admin_invites - Admin invitation system

Dakhla: AI-Assisted University Admissions Platform	Version: 1.0
Software Design Specifications	Date: 19/DEC/2025
FYP-020/FA25	

3.5 System Interaction Design

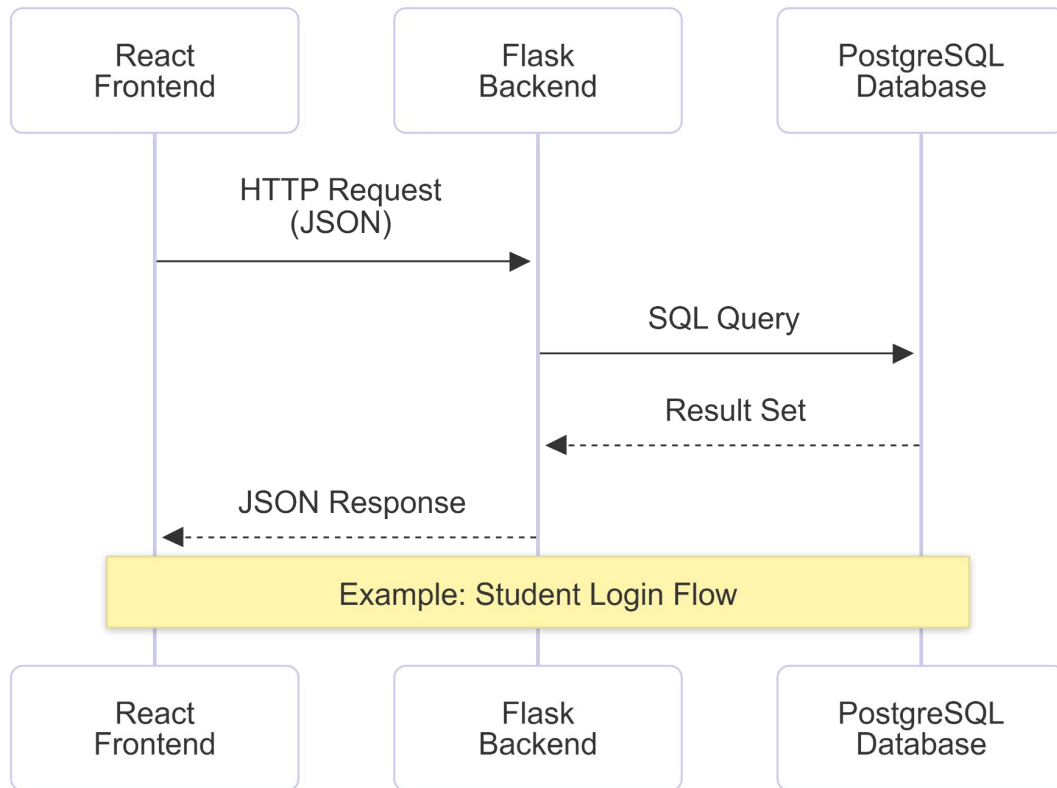


Figure 3.5: Sequence Diagram illustrating the HTTP request/response cycle between frontend, backend, and database.

Example Flow: Student Login

1. User enters credentials in React login form
2. Frontend sends POST request to `/api/auth/student/login`
3. Flask backend validates credentials against `students_new` table
4. Backend generates JWT token
5. Token returned to frontend and stored in `localStorage`
6. Subsequent requests include JWT in Authorization header
7. Backend validates JWT before processing protected routes

Dakhla: AI-Assisted University Admissions Platform	Version: 1.0
Software Design Specifications	Date: 19/DEC/2025
FYP-020/FA25	

4. Design Strategy

This section describes the design strategies and decisions that impact the overall organization of the system and its high-level structures.

4.1 Object-Oriented Design Approach

The system follows **Object-Oriented Programming (OOP) principles** [17] throughout both frontend and backend:

- **Encapsulation:** Data and methods are encapsulated within classes (Python models, React components)
- **Inheritance:** Common functionality is inherited (SQLAlchemy base models, React component hierarchy)
- **Polymorphism:** Different user types (Student, University, Admin) share common interfaces but implement specific behaviors
- **Abstraction:** Complex operations are abstracted behind simple interfaces (API endpoints, React hooks)

4.2 Separation of Concerns

The architecture strictly separates:

- **Presentation Logic** (React components) from **Business Logic** (Flask services)
- **Business Logic** from **Data Access** (SQLAlchemy ORM)
- **Authentication** from **Authorization** (JWT tokens + role-based permissions)

This separation enables:

- Independent development and testing of each layer
- Easy replacement of components without affecting others
- Better maintainability and scalability

4.3 API-First Design

The system follows an **API-first approach**:

- Backend exposes RESTful APIs as the primary interface
- Frontend consumes these APIs exclusively
- Clear API contracts defined before implementation
- Enables future mobile app development without backend changes

Dakhla: AI-Assisted University Admissions Platform	Version: 1.0
Software Design Specifications	Date: 19/DEC/2025
FYP-020/FA25	

4.4 Security-First Design

Security is embedded at every layer:

- **Authentication:** JWT-based stateless authentication
- **Authorization:** Role-Based Access Control (RBAC) for all protected routes
- **Data Protection:** Password hashing with bcrypt, SQL injection prevention via ORM
- **Communication Security:** HTTPS for all API calls, CORS restrictions
- **Input Validation:** Both client-side and server-side validation

4.5 Scalability Strategy

The design supports horizontal and vertical scaling:

- **Stateless Backend:** No server-side sessions, enabling load balancing
- **Database Optimization:** Indexing, caching, query optimization
- **CDN Integration:** Static assets served via Vercel's global CDN
- **Caching Layer:** Flask-Caching for frequently accessed data
- **Modular Architecture:** Independent scaling of frontend and backend

4.6 Data Management Strategy

Storage Strategy

- **Relational Database:** PostgreSQL for structured data with ACID compliance
- **Normalized Schema:** Reduces redundancy, ensures data integrity
- **Foreign Key Constraints:** Maintains referential integrity
- **Soft Deletes:** is_active flags instead of hard deletes for audit trails

Distribution Strategy

- **Centralized Database:** Single source of truth on Supabase
- **Read Replicas:** Potential future implementation for read-heavy operations
- **Caching:** In-memory caching for frequently accessed lookup tables

Persistence Strategy

- **Automatic Backups:** Managed by Supabase
- **Transaction Management:** SQLAlchemy session management
- **Migration Support:** Alembic for database schema versioning

Dakhla: AI-Assisted University Admissions Platform	Version: 1.0
Software Design Specifications	Date: 19/DEC/2025
FYP-020/FA25	

4.7 Concurrency and Synchronization

- **Database Level:** PostgreSQL's Multi-Version Concurrency Control (MVCC)
- **Application Level:** SQLAlchemy's session management and transaction isolation
- **API Level:** Stateless design eliminates most concurrency issues
- **Optimistic Locking:** updated_at timestamps for conflict detection

4.8 User Interface Paradigm

- **Responsive Design:** Mobile-first approach using Tailwind CSS
- **Single Page Application:** Smooth navigation without page reloads
- **Component-Based:** Reusable React components
- **Accessibility:** Web Content Accessibility Guidelines (WCAG 2.1) compliance considerations
- **Progressive Enhancement:** Core functionality works without JavaScript

4.9 Future System Extension

The design facilitates future enhancements:

- **Plugin Architecture:** For AI service providers (Adapter pattern)
- **Microservices Migration:** Modular design enables gradual migration
- **Mobile Apps:** API-first design supports native mobile clients
- **Third-Party Integrations:** Webhook support for external systems
- **Multi-Language Support:** Internationalization (i18n) ready structure

4.10 System Reuse

Reusable components and patterns:

- **React Component Library:** Shared UI components across all portals
- **API Client Library:** Centralized Axios configuration
- **Database Models:** Reusable SQLAlchemy models
- **Authentication Middleware:** Reusable JWT decorators
- **Validation Schemas:** Shared validation logic

Dakhla: AI-Assisted University Admissions Platform	Version: 1.0
Software Design Specifications	Date: 19/DEC/2025
FYP-020/FA25	

5. Detailed System Design

This section provides detailed design specifications including class diagrams, database design, application design, and GUI mockups.

5.1 Design Class Diagram

This section presents the object-oriented design of the Dakhla platform through comprehensive class diagrams. The system follows a highly normalized database design with clear separation of concerns and strong object-oriented principles.

5.1.1 Overview of Class Structure

The Dakhla platform's class structure is organized into seven main categories:

1. **Student Domain Model** - Normalized student data structure with separate classes for personal info, contact info, address, authentication, and academic information
2. **University Domain Model** - Normalized university profile structure with modular components for authentication, contact, address, and media management
3. **Program and Campus Facilities** - Academic program management and university facility tracking
4. **Admin and Authentication** - Administrative user management with Role-Based Access Control (RBAC)
5. **Lookup and Reference Tables** - Standardized reference data for countries, cities, fields of study, degree types, etc.
6. **Static University Data** - Pre-populated university information for institutions not yet registered
7. **Complete System Overview** - High-level view of all domain model interactions

5.1.2 Key Design Patterns

The class design implements several important design patterns:

1. Normalized Data Pattern

- Student and University data are split into multiple related classes
- Each class has a single responsibility (e.g., ContactInfo only handles contact details)
- Reduces data redundancy and ensures consistency
- Example: Student class references PersonalInfo, ContactInfo, AddressInfo, UserAuth, and AcademicInfo

Dakhla: AI-Assisted University Admissions Platform	Version: 1.0
Software Design Specifications	Date: 19/DEC/2025
FYP-020/FA25	

2. Computed Properties Pattern

- Classes expose computed properties for backward compatibility
- Example: Student.full_name property retrieves data from PersonalInfo.full_name
- Provides a clean API while maintaining normalized structure

3. Soft Delete Pattern

- Most classes include an is_active boolean flag
- Records are marked inactive instead of being deleted
- Maintains data integrity and audit trails

4. Timestamp Pattern

- All classes include created_at and updated_at timestamps
- Automatic tracking of record creation and modification
- Essential for auditing and debugging

5. Role-Based Access Control (RBAC) Pattern

- AdminUser class includes granular permission flags
- has_permission() method centralizes permission checking
- Super admins automatically have all permissions

Dakhla: AI-Assisted University Admissions Platform	Version: 1.0
Software Design Specifications	Date: 19/DEC/2025
FYP-020/FA25	

5.1.3 Class Diagrams

Diagram 1: Student Domain Model

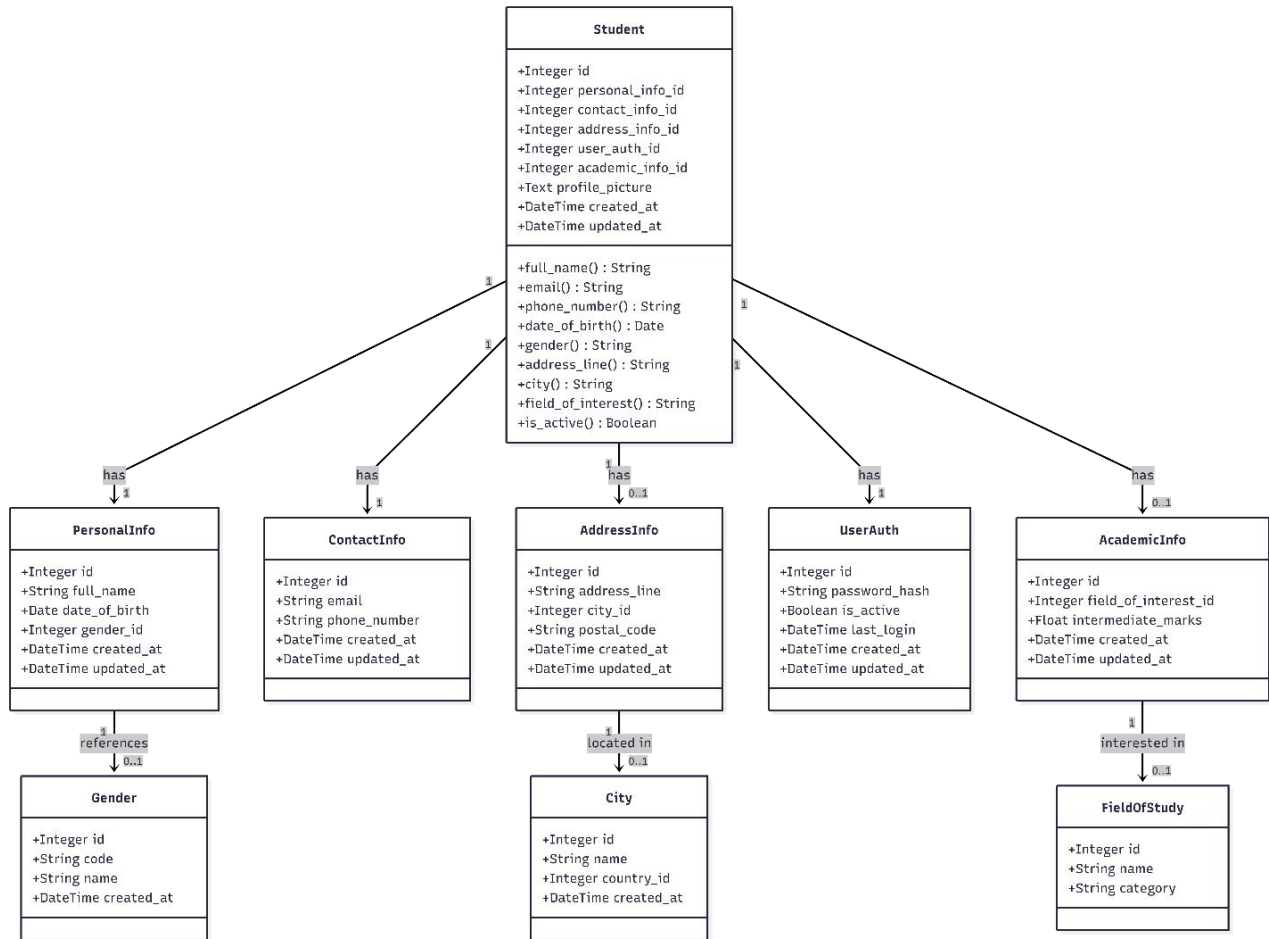


Figure 5.1: Student Domain Model showing the Student class and its relationships with normalized data tables (PersonalInfo, ContactInfo, AddressInfo, UserAuth, AcademicInfo).

This diagram illustrates the Student class and its relationships with normalized data tables. The Student class acts as an aggregate root, coordinating access to PersonalInfo, ContactInfo, AddressInfo, UserAuth, and AcademicInfo. Each related class has a specific responsibility:

- **PersonalInfo:** Stores name, date of birth, and gender
- **ContactInfo:** Manages email and phone number
- **AddressInfo:** Handles address details and city reference
- **UserAuth:** Manages password hash, active status, and last login
- **AcademicInfo:** Tracks field of interest and academic marks

Dakhla: AI-Assisted University Admissions Platform	Version: 1.0
Software Design Specifications	Date: 19/DEC/2025
FYP-020/FA25	

Diagram 2: University Domain Model

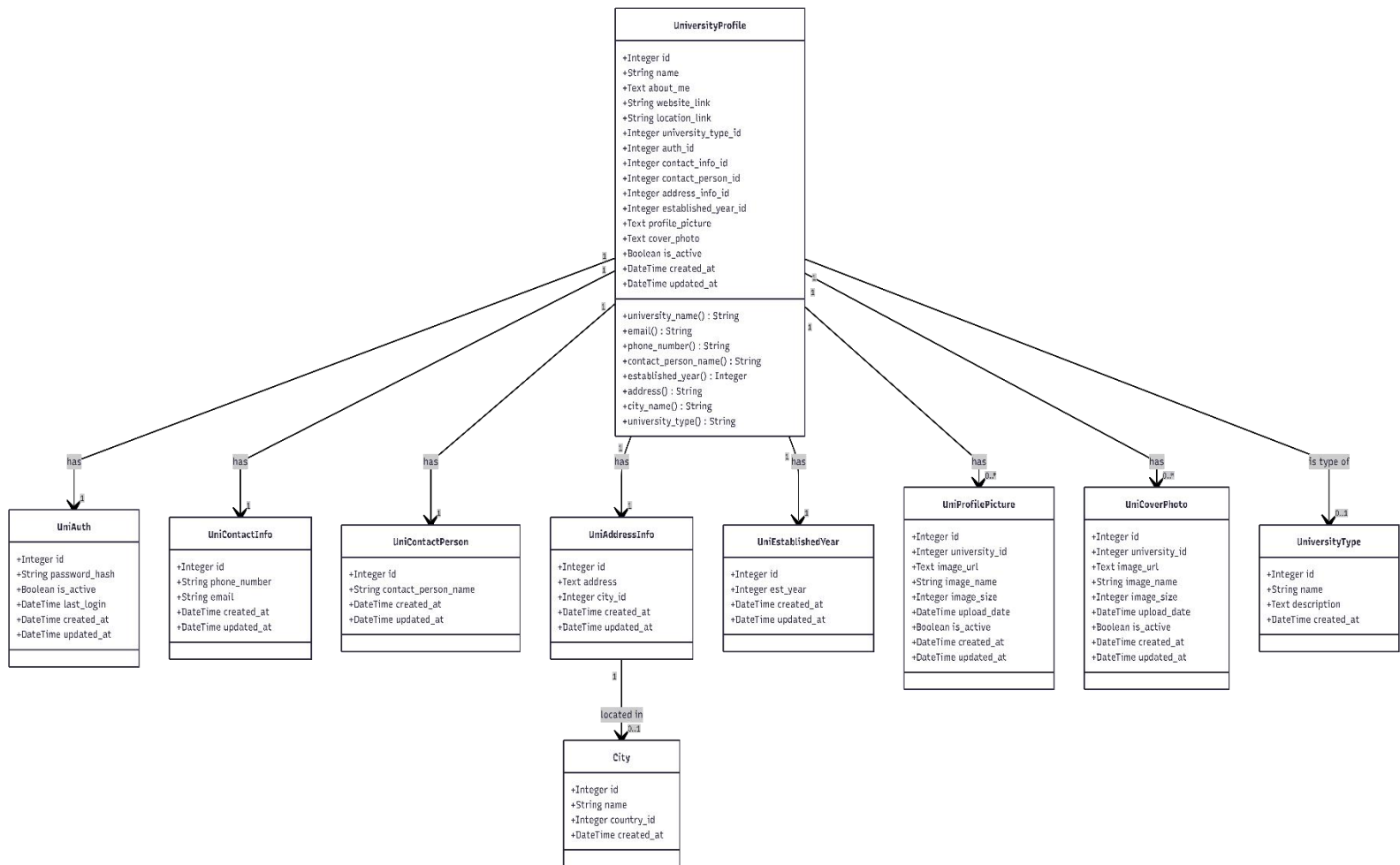


Figure 5.2: University Domain Model showing the UniversityProfile class structure with its normalized components.

This diagram shows the UniversityProfile class structure with its normalized components. Similar to the Student model, the university data is split into focused classes:

- **UniAuth:** Authentication credentials and status
- **UniContactInfo:** Email and phone number
- **UniContactPerson:** Contact person details
- **UniAddressInfo:** Address and location information
- **UniEstablishedYear:** Year of establishment
- **UniProfilePicture:** Profile image management
- **UniCoverPhoto:** Cover photo management

Dakhla: AI-Assisted University Admissions Platform	Version: 1.0
Software Design Specifications	Date: 19/DEC/2025
FYP-020/FA25	

Diagram 3: Program and Campus Facilities

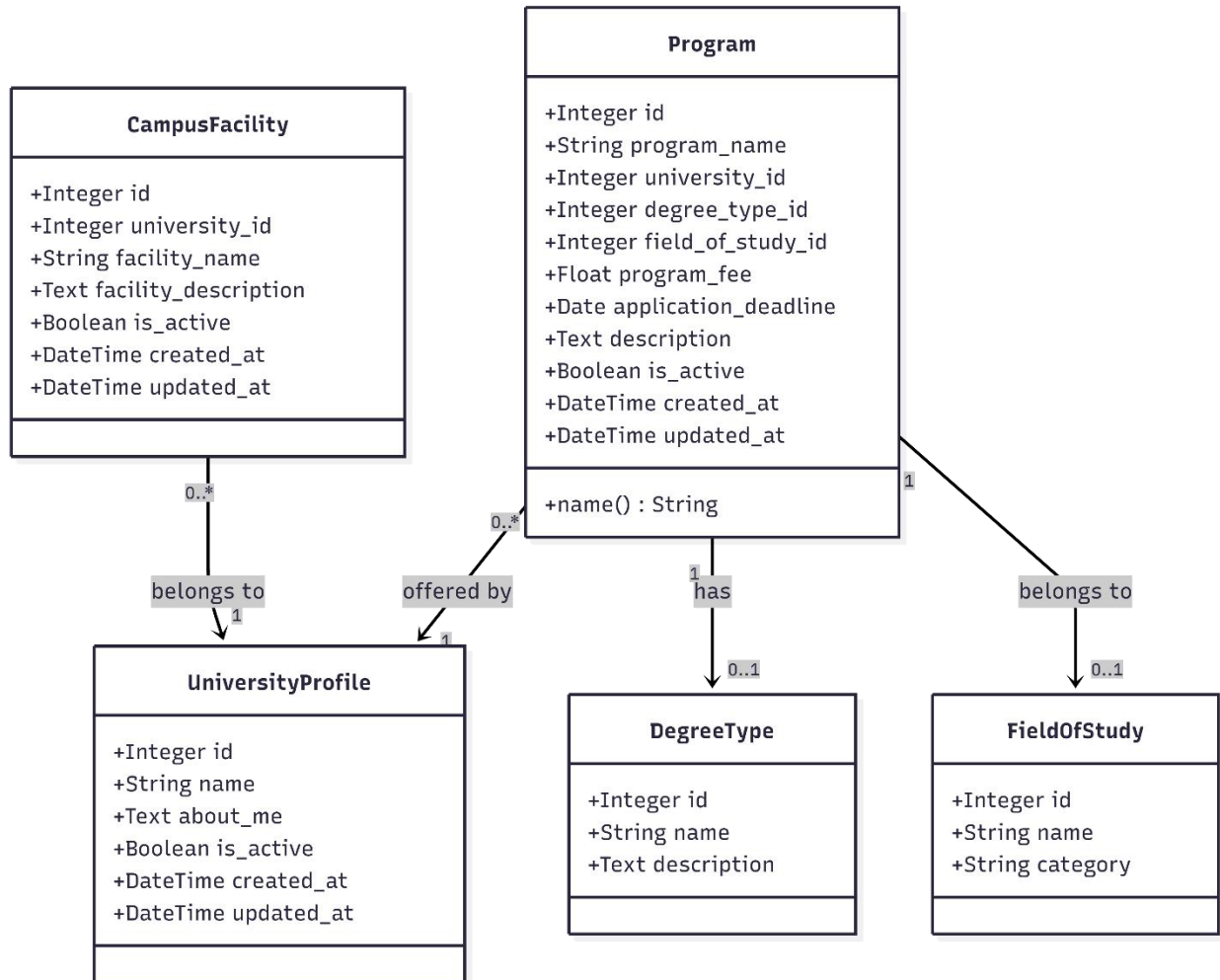


Figure 5.3: Program and Campus Facilities showing how academic programs are structured and linked to universities.

This diagram demonstrates how academic programs are structured and linked to universities. The **Program** class includes:

- Reference to the offering **UniversityProfile**
- Link to **DegreeType** (Bachelor's, Master's, etc.)
- Link to **FieldOfStudy** (Computer Science, Engineering, etc.)
- Program-specific details (fee, deadline, description)

The **CampusFacility** class represents university facilities and amenities.

Dakhla: AI-Assisted University Admissions Platform	Version: 1.0
Software Design Specifications	Date: 19/DEC/2025
FYP-020/FA25	

Diagram 4: Admin and Authentication

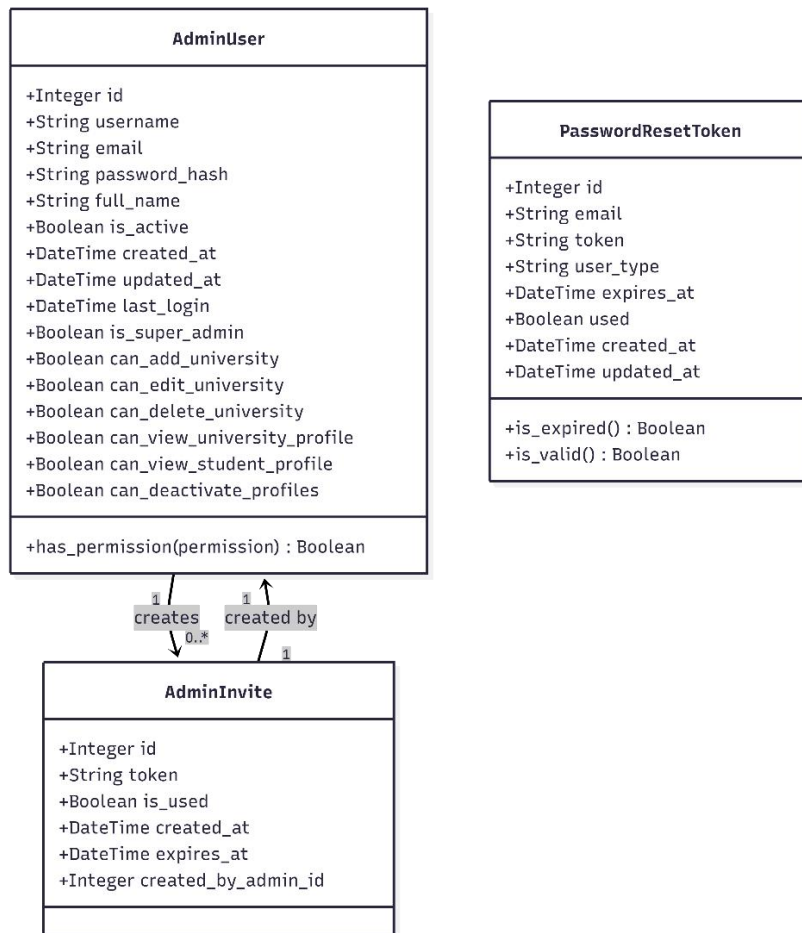


Figure 5.4: Admin and Authentication showing the administrative user management system.

This diagram shows the administrative user management system:

- **AdminUser:** Core admin account with RBAC permissions
- **AdminInvite:** Invitation-based registration system
- **PasswordResetToken:** Secure password reset mechanism

The AdminUser class includes granular permissions:

- can_add_university
- can_edit_university
- can_delete_university
- can_view_university_profile
- can_view_student_profile
- can_deactivate_profiles

Dakhla: AI-Assisted University Admissions Platform	Version: 1.0
Software Design Specifications	Date: 19/DEC/2025
FYP-020/FA25	

Diagram 5: Static University Data Model

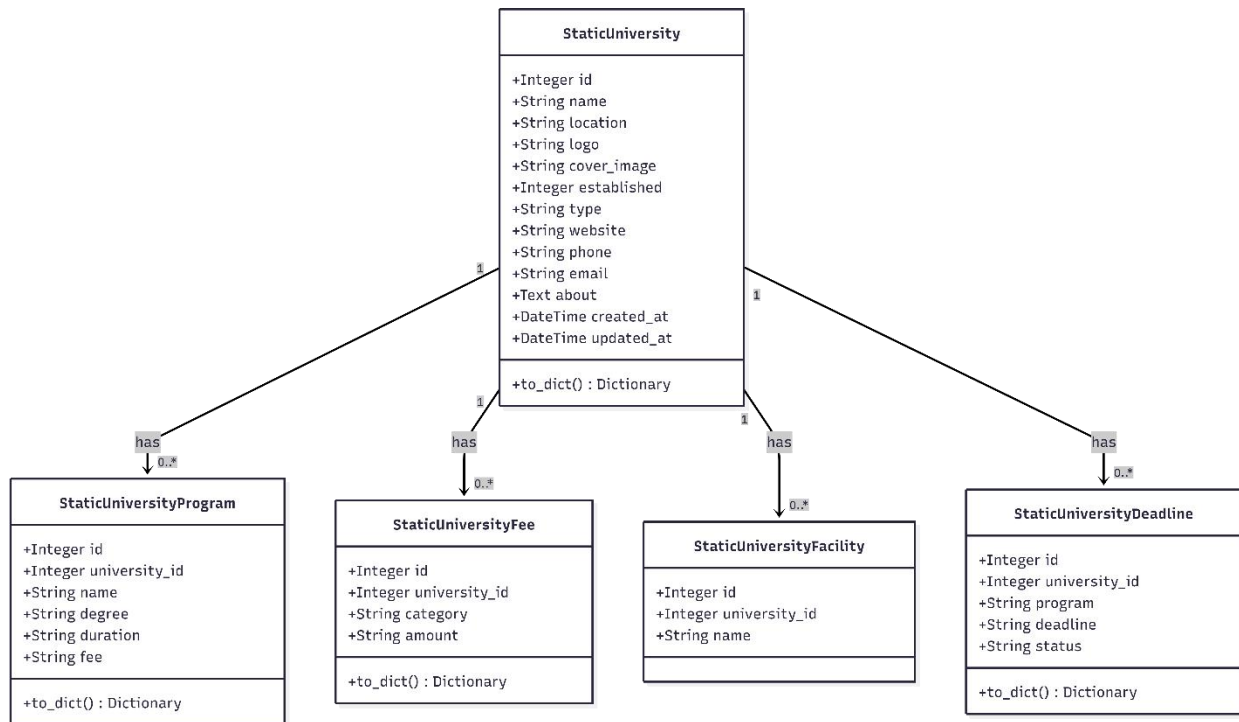


Figure 5.5: Static University Data Model showing the structure for pre-populated university data.

This diagram shows the structure for pre-populated university data:

- **StaticUniversity:** Core university information
- **StaticUniversityProgram:** Program offerings
- **StaticUniversityFee:** Fee structure
- **StaticUniversityFacility:** Available facilities
- **StaticUniversityDeadline:** Application deadlines

This structure is used for universities that haven't registered on the platform yet, allowing students to browse and get information.

Dakhla: AI-Assisted University Admissions Platform	Version: 1.0
Software Design Specifications	Date: 19/DEC/2025
FYP-020/FA25	

Diagram 6: Complete System Overview

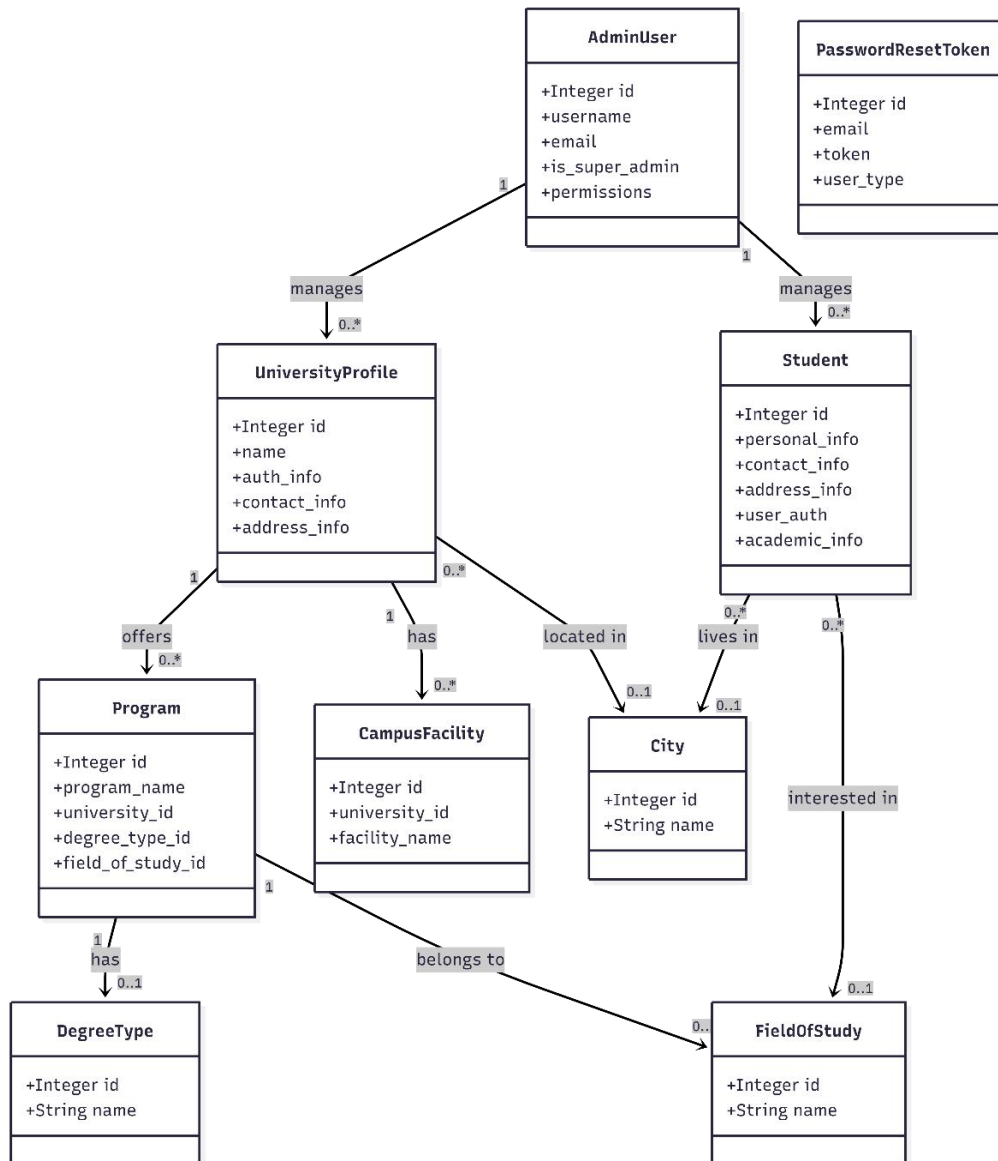


Figure 5.6: Complete System Overview showing how all major domain models interact.

This high-level diagram shows how all major domain models interact:

- Students are linked to FieldOfStudy (interest) and City (location)
- Universities offer Programs and have CampusFacilities
- Programs are categorized by DegreeType and FieldOfStudy
- AdminUsers manage both Students and UniversityProfiles
- PasswordResetToken supports both user types

Dakhla: AI-Assisted University Admissions Platform	Version: 1.0
Software Design Specifications	Date: 19/DEC/2025
FYP-020/FA25	

5.1.4 Key Relationships

One-to-One Relationships:

- Student ↔ PersonalInfo
- Student ↔ ContactInfo
- Student ↔ UserAuth
- UniversityProfile ↔ UniAuth
- UniversityProfile ↔ UniContactInfo

One-to-Many Relationships:

- UniversityProfile → Programs (one university offers many programs)
- UniversityProfile → CampusFacilities (one university has many facilities)
- Country → Cities (one country contains many cities)
- AdminUser → AdminInvites (one admin creates many invites)

Many-to-One Relationships:

- Students → City (many students live in one city)
- Students → FieldOfStudy (many students interested in one field)
- Programs → DegreeType (many programs have one degree type)
- Programs → FieldOfStudy (many programs belong to one field)

Dakhla: AI-Assisted University Admissions Platform	Version: 1.0
Software Design Specifications	Date: 19/DEC/2025
FYP-020/FA25	

5.1.5 Class Responsibilities

Student Class Responsibilities:

- Aggregate root for student data
- Coordinate access to normalized tables
- Provide computed properties for easy data access
- Maintain backward compatibility with legacy code

UniversityProfile Class Responsibilities:

- Aggregate root for university data
- Manage university profile information
- Coordinate relationships with programs and facilities
- Provide computed properties for profile data

Program Class Responsibilities:

- Represent academic programs
- Link programs to universities
- Categorize programs by degree type and field
- Manage program-specific details (fees, deadlines)

AdminUser Class Responsibilities:

- Manage administrative accounts
- Implement Role-Based Access Control
- Track admin permissions and activities
- Support invitation-based registration

5.1.6 Design Principles Applied

Single Responsibility Principle (SRP):

Each class has a single, well-defined responsibility. For example, ContactInfo only handles contact details, not authentication or personal information.

Open/Closed Principle (OCP):

Classes are open for extension but closed for modification. Computed properties allow extending functionality without changing the core class structure.

Dependency Inversion Principle (DIP):

High-level classes (Student, UniversityProfile) depend on abstractions (relationships) rather than concrete implementations.

Dakhla: AI-Assisted University Admissions Platform	Version: 1.0
Software Design Specifications	Date: 19/DEC/2025
FYP-020/FA25	

Don't Repeat Yourself (DRY):

Common patterns (timestamps, soft deletes, active flags) are consistently applied across all classes.

Separation of Concerns:

Authentication, contact information, and personal data are separated into distinct classes, making the system more maintainable and testable.

5.2 Database Design

The Dakhla platform uses a **highly normalized PostgreSQL relational database** hosted on Supabase. The database schema is designed to ensure data integrity, minimize redundancy, and support efficient querying.

5.2.1 Entity-Relationship Diagram

The Entity-Relationship diagram below illustrates the complete database schema with all entities, attributes, and relationships:

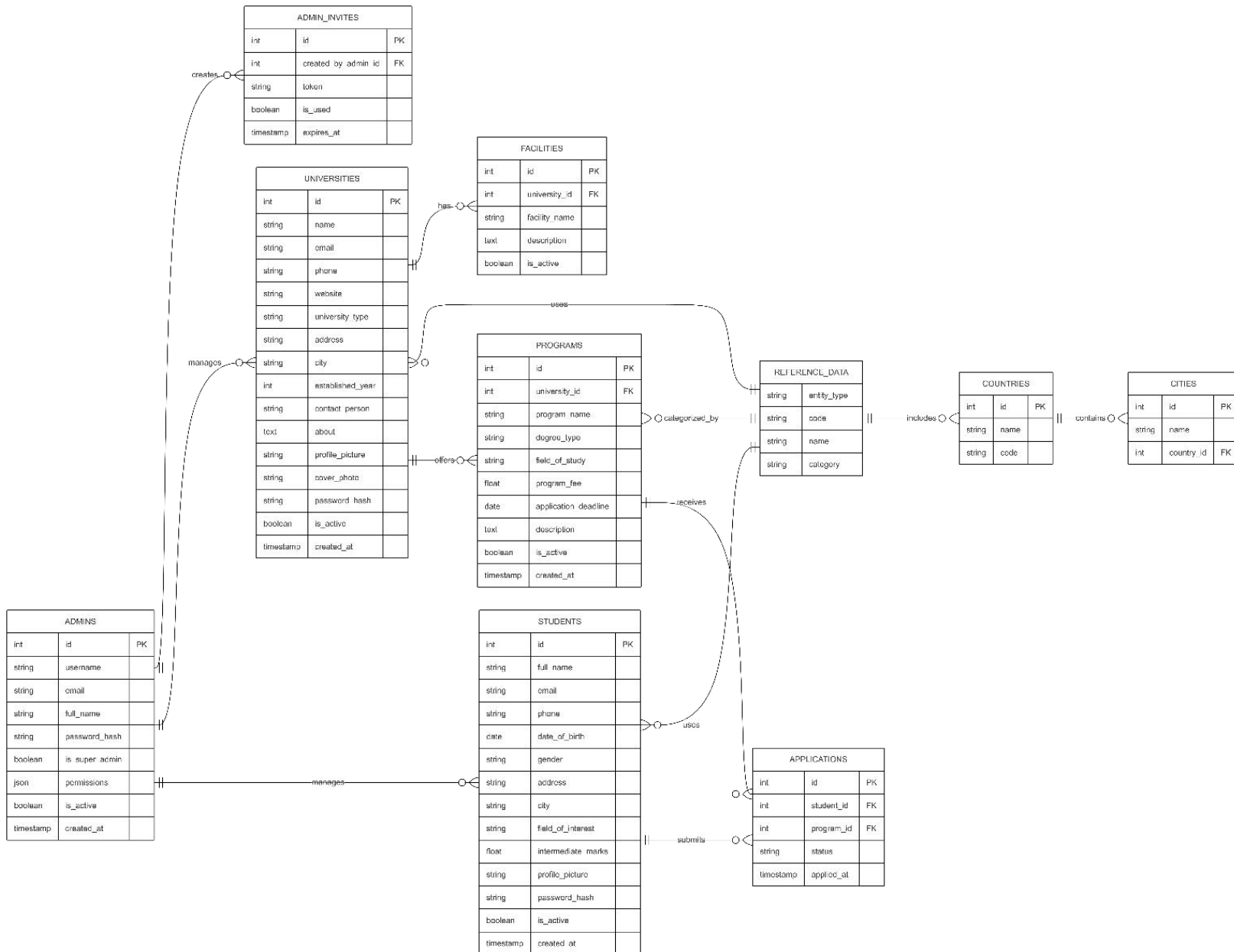


Figure 5.8: Complete Entity-Relationship Diagram showing all 17 tables with their relationships, primary keys, and foreign keys.

Dakhla: AI-Assisted University Admissions Platform	Version: 1.0
Software Design Specifications	Date: 19/DEC/2025
FYP-020/FA25	

Key Design Decisions:

1. **Normalization:** The schema is in **Third Normal Form (3NF)** to eliminate redundancy
2. **Separation of Concerns:** Personal, contact, address, and academic information are stored in separate tables
3. **Referential Integrity:** Foreign key constraints ensure data consistency
4. **Soft Deletes:** `is_active` flags allow logical deletion without data loss
5. **Audit Trail:** `created_at` and `updated_at` timestamps on all major tables
6. **Lookup Tables:** Normalized reference data (countries, cities, fields, etc.)

ERD Legend:

- **PK** = Primary Key
- **FK** = Foreign Key
- **UK** = Unique Key
- **||--||** = One-to-One relationship
- **||--o{** = One-to-Many relationship
- **{o--||** = Many-to-One relationship

Dakhla: AI-Assisted University Admissions Platform	Version: 1.0
Software Design Specifications	Date: 19/DEC/2025
FYP-020/FA25	

5.2.2 Data Dictionary

This section provides detailed specifications for each table in the database, including column definitions, data types, constraints, and relationships.

5.2.2.1 *students_new*

Description: Core table storing student user accounts with references to normalized profile information tables.

Alias: Student, StudentProfile

Where-used/how-used:

- Used by Student Registration process (input)
- Used by Student Login process (input)
- Used by Student Profile Management (input/output)
- Used by Application Submission process (input)

Content Description: `students_new = id + personal_info_id + contact_info_id + [address_info_id] + user_auth_id + [academic_info_id] + [profile_picture] + created_at + updated_at`

Column Name	Description	Type	Length	Nullable	Default Value	KeyType
id	Unique identifier for student record	integer	int4	No	nextval('students_new_id_seq')	PK
personal_info_id	Foreign key to personal_info table	integer	int4	No	-	FK
contact_info_id	Foreign key to contact_info table	integer	int4	No	-	FK
address_info_id	Foreign key to address_ table	integer	int4	Yes	NULL	FK

Dakhla: AI-Assisted University Admissions Platform	Version: 1.0
Software Design Specifications	Date: 19/DEC/2025
FYP-020/FA25	

	info table					
user_auth_id	Foreign key to user_auth table	integer	int4	No	-	FK
academic_info_id	Foreign key to academic_info table	integer	int4	Yes	NULL	FK
profile_picture	URL or path to student's profile picture	text	-	Yes	NULL	-
created_at	Timestamp when record was created	timestamp	-	Yes	CURRENT_TIMESTAMP	-
updated_at	Timestamp when record was last updated	timestamp	-	Yes	CURRENT_TIMESTAMP	-

Relationships:

- students_new.personal_info_id → personal_info.id (1:1)
- students_new.contact_info_id → contact_info.id (1:1)
- students_new.address_info_id → address_info.id (1:1)
- students_new.user_auth_id → user_auth.id (1:1)
- students_new.academic_info_id → academic_info.id (1:1)

Dakhla: AI-Assisted University Admissions Platform	Version: 1.0
Software Design Specifications	Date: 19/DEC/2025
FYP-020/FA25	

5.2.2.2 *personal_info*

Description: Stores personal information for students including name, date of birth, and gender.

Alias: StudentPersonalInfo

Where-used/how-used:

- Used by Student Registration process (output)
- Used by Student Profile Management (input/output)
- Used by Student Profile Display (input)

Content Description: *personal_info* = *id* + *full_name* + [*date_of_birth*] + [*gender_id*] + *created_at* + *updated_at*

Column Name	Description	Type	Length	Nullabl e	Default Value	Key Type
id	Unique identifier for personal info record	integer	int4	No	nextval('personal_info_id_seq')	PK
full_name	Student's complete name	varchar	100	No	-	-
date_of_birth	Student's date of birth	date	-	Yes	NULL	-
gender_id	Foreign key to genders table	integer	int4	Yes	NULL	FK
created_at	Timestamp when record was created	timestamp	-	Yes	CURRENT_TIMESTAMP	-
updated_at	Timestamp when record was last updated	timestamp	-	Yes	CURRENT_TIMESTAMP	-

Relationships:

- *personal_info.gender_id* → *genders.id* (N:1)

Dakhla: AI-Assisted University Admissions Platform	Version: 1.0
Software Design Specifications	Date: 19/DEC/2025
FYP-020/FA25	

5.2.2.3 *contact_info*

Description: Stores contact information for students including email and phone number.

Alias: StudentContactInfo

Where-used/how-used:

- Used by Student Registration process (output)
- Used by Student Login process (input)
- Used by Password Reset process (input)
- Used by Notification System (input)

Content Description: contact_info = id + email + phone_number + created_at + updated_at

Column Name	Description	Type	Length	Nullable	Default Value	Key Type
id	Unique identifier for contact info record	integer	int4	No	nextval('contact_info_id_seq')	PK
email	Student's email address (unique)	varchar	120	No	-	UNIQUE
phone_number	Student's phone number	varchar	11	No	-	-
created_at	Timestamp when record was created	timestamp	-	Yes	CURRENT_TIMESTAMP	-
updated_at	Timestamp when record was last updated	timestamp	-	Yes	CURRENT_TIMESTAMP	-

Dakhla: AI-Assisted University Admissions Platform	Version: 1.0
Software Design Specifications	Date: 19/DEC/2025
FYP-020/FA25	

5.2.2.4 address_info

Description: Stores address information for students including street address, city, and postal code.

Alias: StudentAddressInfo

Where-used/how-used:

- Used by Student Profile Management (input/output)
- Used by Location-based Search (input)
- Used by University Recommendations (input)

Content Description: address_info = id + [address_line] + [city_id] + [postal_code] + created_at + updated_at

Column Name	Description	Type	Length	Nullable	Default Value	Key Type
id	Unique identifier for address info record	integer	int4	No	nextval('address_info_id_seq')	PK
address_line	Street address Or detailed location	varchar	255	Yes	NULL	-
city_id	Foreign key to cities table	integer	int4	Yes	NULL	FK
postal_code	Postal or ZIP code	varchar	20	Yes	NULL	-
created_at	Timestamp when record was created	timestamp	-	Yes	CURRENT_TIMESTAMP	-
updated_at	Timestamp when record was last updated	timestamp	-	Yes	CURRENT_TIMESTAMP	-

Relationships:

- address_info.city_id → cities.id (N:1)

Dakhla: AI-Assisted University Admissions Platform	Version: 1.0
Software Design Specifications	Date: 19/DEC/2025
FYP-020/FA25	

5.2.2.5 user_auth

Description: Stores authentication credentials and status for student users.

Alias: StudentAuth, StudentCredentials

Where-used/how-used:

- Used by Student Login process (input)
- Used by Student Registration process (output)
- Used by Password Reset process (input/output)
- Used by Authentication Middleware (input)

Content Description: user_auth = id + password_hash + [is_active] + [last_login] + created_at + updated_at

Column Name	Description	Type	Length	Nullable	Default Value	Key Type
id	Unique identifier for auth record	integer	int4	No	nextval('user_auth_id_seq')	PK
password_hash	Bcrypt hashed password	varchar	255	No	-	-
is_active	Account active status flag	boolean	-	Yes	TRUE	-
last_login	Timestamp of last successful login	timestamp	-	Yes	NULL	-
created_at	Timestamp when record was created	timestamp	-	Yes	CURRENT_TIMESTAMP	-
updated_at	Timestamp when record was last updated	timestamp	-	Yes	CURRENT_TIMESTAMP	-

Dakhla: AI-Assisted University Admissions Platform	Version: 1.0
Software Design Specifications	Date: 19/DEC/2025
FYP-020/FA25	

5.2.2.6 *academic_info*

Description: Stores academic information for students including field of interest and intermediate marks.

Alias: StudentAcademicInfo

Where-used/how-used:

- Used by Student Profile Management (input/output)
- Used by University Recommendation Engine (input)
- Used by Program Matching Algorithm (input)

Content Description: *academic_info* = id + [field_of_interest_id] + [intermediate_marks] + created_at + updated_at

Column Name	Description	Type	Length	Nullable	Default Value	Key Type
id	Unique identifier for academic info record	integer	int4	No	nextval('academic_info_id_seq')	PK
field_of_interest_id	Foreign key to fields_of_study table	integer	int4	Yes	NULL	FK
intermediate_marks	Student's intermediate/12th grade marks	float	float8	Yes	NULL	-
created_at	Timestamp when record was created	timestamp	-	Yes	CURRENT_TIMESTAMP	-
updated_at	Timestamp when record was last updated	timestamp	-	Yes	CURRENT_TIMESTAMP	-

Relationships:

- *academic_info.field_of_interest_id* → *fields_of_study.id* (N:1)

Dakhla: AI-Assisted University Admissions Platform	Version: 1.0
Software Design Specifications	Date: 19/DEC/2025
FYP-020/FA25	

5.2.2.7 *university_profiles*

Description: Core table storing university accounts with references to normalized profile information tables.

Alias: University, UniversityAccount

Where-used/how-used:

- Used by University Registration process (input/output)
- Used by University Login process (input)
- Used by University Profile Management (input/output)
- Used by University Search (input)
- Used by Program Management (input)

Content Description: university_profiles = id + name + [about_me] + [website_link] + [location_link] + [university_type_id] + auth_id + contact_info_id + contact_person_id + address_info_id + established_year_id + [is_active] + created_at + updated_at

Column Name	Description	Type	Length	Nullable	Default Value	Key Type
id	Unique identifier for university record	integer	int4	No	nextval('university_profiles_id_seq')	PK
name	Official name of the university	varchar	200	No	-	-
about_me	Description/about section for university	text	-	Yes	NULL	-
website_link	University's official website URL	varchar	255	Yes	NULL	-
location_link	Google Maps or location link	varchar	255	Yes	NULL	-
university_type_id	Foreign key to university_types table	integer	int4	Yes	NULL	FK
auth_id	Foreign key to uni_auth table	integer	int4	No	-	FK
contact_info_id	Foreign key to uni_contact_info table	integer	int4	No	-	FK
contact_person_id	Foreign key to uni_contact person	integer	int4	No	-	FK

Dakhla: AI-Assisted University Admissions Platform	Version: 1.0
Software Design Specifications	Date: 19/DEC/2025
FYP-020/FA25	

	table					
address_info_id	Foreign key to uni_address_info table	integer	int4	No	-	FK
established_year_id	Foreign key to uni_established_year table	integer	int4	No	-	FK
is_active	University account active status	boolean	-	Yes	TRUE	
created_at	Timestamp when record was created	timestamp	-	Yes	CURRENT_TIMESTAMP	
updated_at	Timestamp when record was last updated	timestamp	-	Yes	CURRENT_TIMESTAMP	

Relationships:

- university_profiles.university_type_id → university_types.id (N:1)
- university_profiles.auth_id → uni_auth.id (1:1)
- university_profiles.contact_info_id → uni_contact_info.id (1:1)
- university_profiles.contact_person_id → uni_contact_person.id (1:1)
- university_profiles.address_info_id → uni_address_info.id (1:1)
- university_profiles.established_year_id → uni_established_year.id (1:1)

Dakhla: AI-Assisted University Admissions Platform	Version: 1.0
Software Design Specifications	Date: 19/DEC/2025
FYP-020/FA25	

5.2.2.8 programs

Description: Stores academic programs offered by universities including degree type, field of study, fees, and deadlines.

Alias: AcademicProgram, UniversityProgram

Where-used/how-used:

- Used by Program Management (input/output)
- Used by Program Search (input)
- Used by University Recommendation Engine (input)
- Used by Application Submission (input)

Content Description: programs = id + program_name + university_id + [degree_type_id] + [field_of_study_id] + [program_fee] + [application_deadline] + [description] + [is_active] + created_at + updated_at

Column Name	Description	Type	Length	Nullable	Default Value	Key Type
id	Unique identifier for program record	integer	int4	No	nextval('programs_id_seq')	PK
program_name	Name of the academic program	varchar	200	No	-	-
university_id	Foreign key to university_profiles table	integer	int4	No	-	FK
degree_type_id	Foreign key to degree_types table	integer	int4	Yes	NULL	FK
field_of_study_id	Foreign key to fields_of_study table	integer	int4	Yes	NULL	FK
program_fee	Program tuition fee amount	float	float8	Yes	NULL	-

Dakhla: AI-Assisted University Admissions Platform	Version: 1.0
Software Design Specifications	Date: 19/DEC/2025
FYP-020/FA25	

application_deadline	Last date to apply for program	date	-	Yes	NULL	-
description	Detailed program description	text	-	Yes	NULL	-
is_active	Program active status flag	boolean	-	Yes	TRUE	-
created_at	Timestamp when record was created	timestamp	-	Yes	CURRENT_TIMESTAMP	-
updated_at	Timestamp when record was last updated	timestamp	-	Yes	CURRENT_TIMESTAMP	-

Relationships:

- programs.university_id → university_profiles.id (N:1)
- programs.degree_type_id → degree_types.id (N:1)
- programs.field_of_study_id → fields_of_study.id (N:1)

Dakhla: AI-Assisted University Admissions Platform	Version: 1.0
Software Design Specifications	Date: 19/DEC/2025
FYP-020/FA25	

5.2.2.9 admin_users

Description: Stores administrator user accounts with role-based access control permissions.

Alias: Admin, Administrator, SystemAdmin

Where-used/how-used:

- Used by Admin Login process (input)
- Used by Admin Registration process (output)
- Used by Admin Dashboard (input)
- Used by Permission Checking Middleware (input)

Content Description: admin_users = id + username + email + password_hash + full_name + [is_active] + created_at + updated_at + [last_login] + [is_super_admin] + [can_add_university] + [can_edit_university] + [can_delete_university] + [can_view_university_profile] + [can_view_student_profile] + [can_deactivate_profiles]

Column Name	Description	Type	Length	Nullable	Default Value	Key Type
id	Unique identifier for admin record	integer	int4	No	nextval('admin_users_id_seq')	PK
username	Admin's unique username	varchar	50	No	-	UNIQUE
email	Admin's email address	varchar	120	No	-	UNIQUE
password_hash	Bcrypt hashed password	varchar	255	No	-	-
full_name	Admin's complete name	varchar	100	No	-	-
is_active	Admin account active	boolean	-	Yes	TRUE	-

Dakhla: AI-Assisted University Admissions Platform	Version: 1.0
Software Design Specifications	Date: 19/DEC/2025
FYP-020/FA25	

	status					
created_at	Timestamp when record was created	timestamp	-	Yes	CURRENT_TIMESTAMP	-
updated_at	Timestamp when record was last updated	timestamp	-	Yes	CURRENT_TIMESTAMP	-
last_login	Timestamp of last successful login	timestamp	-	Yes	NULL	-
is_super_admin	Super admin flag (all permissions)	boolean	-	Yes	FALSE	-
can_add_university	Permission to add universities	boolean	-	Yes	FALSE	-
can_edit_university	Permission to edit universities	boolean	-	Yes	FALSE	-
can_delete_university	Permission to delete universities	boolean	-	Yes	FALSE	-
can_view_university_profile	Permission to view university profiles	boolean	-	Yes	FALSE	-

Dakhla: AI-Assisted University Admissions Platform	Version: 1.0
Software Design Specifications	Date: 19/DEC/2025
FYP-020/FA25	

can_view_student_profile	Permission to view student profiles	boolean	-	Yes	FALSE	-
can_deactivate_profiles	Permission to deactivate user profiles	boolean	-	Yes	FALSE	-

Dakhla: AI-Assisted University Admissions Platform	Version: 1.0
Software Design Specifications	Date: 19/DEC/2025
FYP-020/FA25	

5.2.2.10 cities

Description: Lookup table storing city names and their associated countries.

Alias: City, CityList

Where-used/how-used:

- Used by Address Information (input)
- Used by Location-based Search (input)
- Used by University Filtering (input)

Content Description: cities = id + name + country_id + created_at

Column Name	Description	Type	Length	Nullable	Default Value	Key Type
id	Unique identifier for country record	integer	int4	No	nextval('countries_id_seq')	PK
name	Name of the country	varchar	100	No	-	UNIQUE
code	ISO 3166-1 alpha-3 country code	varchar	3	No	-	UNIQUE
created_at	Timestamp when record was created	timestamp	-	Yes	CURRENT_TIMESTAMP	-

Dakhla: AI-Assisted University Admissions Platform	Version: 1.0
Software Design Specifications	Date: 19/DEC/2025
FYP-020/FA25	

5.2.2.12 *fields_of_study*

Description: Lookup table storing academic fields and disciplines.

Alias: Field, AcademicField, StudyField

Where-used/how-used:

- Used by Academic Info Management (input)
- Used by Program Management (input)
- Used by Program Search and Filtering (input)
- Used by Recommendation Engine (input)

Content Description: fields_of_study = id + name + [category]

Column Name	Description	Type	Length	Nullable	Default Value	Key Type
id	Unique identifier for field record	integer	int4	No	nextval('fields_of_study_id_seq')	PK
name	Name of the field of study	varchar	100	No	-	UNIQUE
category	Broad category classification	varchar	50	Yes	NULL	-

Dakhla: AI-Assisted University Admissions Platform	Version: 1.0
Software Design Specifications	Date: 19/DEC/2025
FYP-020/FA25	

5.2.2.13 *degree_types*

Description: Lookup table storing degree classifications (Bachelor's, Master's, PhD, etc.).

Alias: Degree, DegreeLevel

Where-used/how-used:

- Used by Program Management (input)
- Used by Program Search and Filtering (input)

Content Description: degree_types = id + name + [description]

Column Name	Description	Type	Length	Nullable	Default Value	Key Type
id	Unique identifier for degree type record	integer	int4	No	nextval('degree_types_id_seq')	PK
name	Name of the degree type	varchar	100	No	-	UNIQUE
description	Detailed description of degree type	text	-	Yes	NULL	-

Dakhla: AI-Assisted University Admissions Platform	Version: 1.0
Software Design Specifications	Date: 19/DEC/2025
FYP-020/FA25	

5.2.2.14 *university_types*

Description: Lookup table storing university classifications (Public, Private, etc.).

Alias: UniversityCategory

Where-used/how-used:

- Used by University Profile Management (input)
- Used by University Search and Filtering (input)

Content Description: university_types = id + name + [description] + created_at

Column Name	Description	Type	Length	Nullable	Default Value	Key Type
id	Unique identifier for university type record	integer	int4	No	nextval('university_types_id_seq')	PK
name	Name of the university type	varchar	50	No	-	UNIQUE
description	Detailed description of university type	text	-	Yes	NULL	-
created_at	Timestamp when record was created	timestamp	-	Yes	CURRENT_TIMESTAMP	-

Dakhla: AI-Assisted University Admissions Platform	Version: 1.0
Software Design Specifications	Date: 19/DEC/2025
FYP-020/FA25	

5.2.2.15 genders

Description: Lookup table storing gender options.

Alias: Gender

Where-used/how-used:

- Used by Personal Info Management (input)
- Used by Student Registration (input)

Content Description: genders = id + code + name + created_at

Column Name	Description	Type	Length	Nullable	Default Value	Key Type
id	Unique identifier for gender record	integer	int4	No	nextval('genders_id_seq')	PK
code	Single character gender code (M/F)	varchar	1	No	-	UNIQUE
name	Full name of gender (Male/Female)	varchar	10	No	-	UNIQUE
created_at	Timestamp when record was created	timestamp	-	Yes	CURRENT_TIMESTAMP	-

Dakhla: AI-Assisted University Admissions Platform	Version: 1.0
Software Design Specifications	Date: 19/DEC/2025
FYP-020/FA25	

5.2.2.16 password_reset_tokens

Description: Stores temporary tokens for password reset functionality.

Alias: ResetToken, PasswordToken

Where-used/how-used:

- Used by Password Reset Request process (output)
- Used by Password Reset Verification process (input)
- Used by Password Reset Completion process (input/output)

Content Description: password_reset_tokens = id + email + token + user_type + expires_at + [used] + created_at + updated_at

Column Name	Description	Type	Length	Nullable	Default Value	Key Type
id	Unique identifier for token record	integer	int4	No	nextval('password_reset_tokens_id_seq')	PK
email	Email address of user requesting reset	varchar	255	No	-	-
token	Unique reset token (UUID or hash)	varchar	255	No	-	UNIQUE
user_type	Type of user (student/university)	varchar	20	No	-	-
expires_at	Token expiration timestamp	timestamp	-	No	-	-
used	Flag indicating if token has been used	boolean	-	Yes	FALSE	-
created_at	Timestamp when record was created	timestamp	-	Yes	CURRENT_TIMESTAMP	-

Dakhla: AI-Assisted University Admissions Platform	Version: 1.0
Software Design Specifications	Date: 19/DEC/2025
FYP-020/FA25	

updated_at	Timestamp when record was last updated	timestamp	-	Yes	CURRENT_TIMESTAMP	-
-------------------	--	-----------	---	-----	-------------------	---

5.2.2.17 *campus_facility*

Description: Stores facilities and amenities available at university campuses.

Alias: Facility, UniversityFacility

Where-used/how-used:

- Used by University Profile Management (input/output)
- Used by University Detail Display (input)
- Used by University Search Filtering (input)

Content Description: campus_facility = id + university_id + facility_name + [facility_description] + [is_active] + created_at + updated_at

Column Name	Description	Type	Length	Nullable	Default Value	Key Type
id	Unique identifier for facility record	integer	int4	No	nextval('campus_facility_id_seq')	PK
university_id	Foreign key to university_profiles table	integer	int4	No	-	FK
facility_name	Name of the facility	varchar	255	No	-	-
facility_description	Detailed description of facility	text	-	Yes	NULL	-
is_active	Facility active status flag	boolean	-	Yes	TRUE	-
created_at	Timestamp when record was created	timestamp	-	Yes	CURRENT_TIMESTAMP	-

Dakhla: AI-Assisted University Admissions Platform	Version: 1.0
Software Design Specifications	Date: 19/DEC/2025
FYP-020/FA25	

updated_at	Timestamp when record was last updated	timestamp	-	Yes	CURRENT_TIMESTAMP	-
-------------------	--	-----------	---	-----	-------------------	---

Relationships:

- *campus_facility.university_id* → *university_profiles.id* (N:1)

Data Construct Notation Reference

Data Construct	Notation	Meaning
Composition	=	is composed of
Sequence	+	And
Selection	[]	either-or
Repetition	{ }n	n repetitions of
Optional	()	optional data
Comments	* ... *	delimits comments

Dakhla: AI-Assisted University Admissions Platform	Version: 1.0
Software Design Specifications	Date: 19/DEC/2025
FYP-020/FA25	

5.3 Application Design

This section provides detailed application design including sequence diagrams and state diagrams that illustrate the dynamic behaviour of the system.

5.3.1 Sequence Diagrams

Sequence diagrams show the interactions between objects in a sequential order, emphasizing the time ordering of messages.

5.3.1.1 Student Registration Sequence Diagram

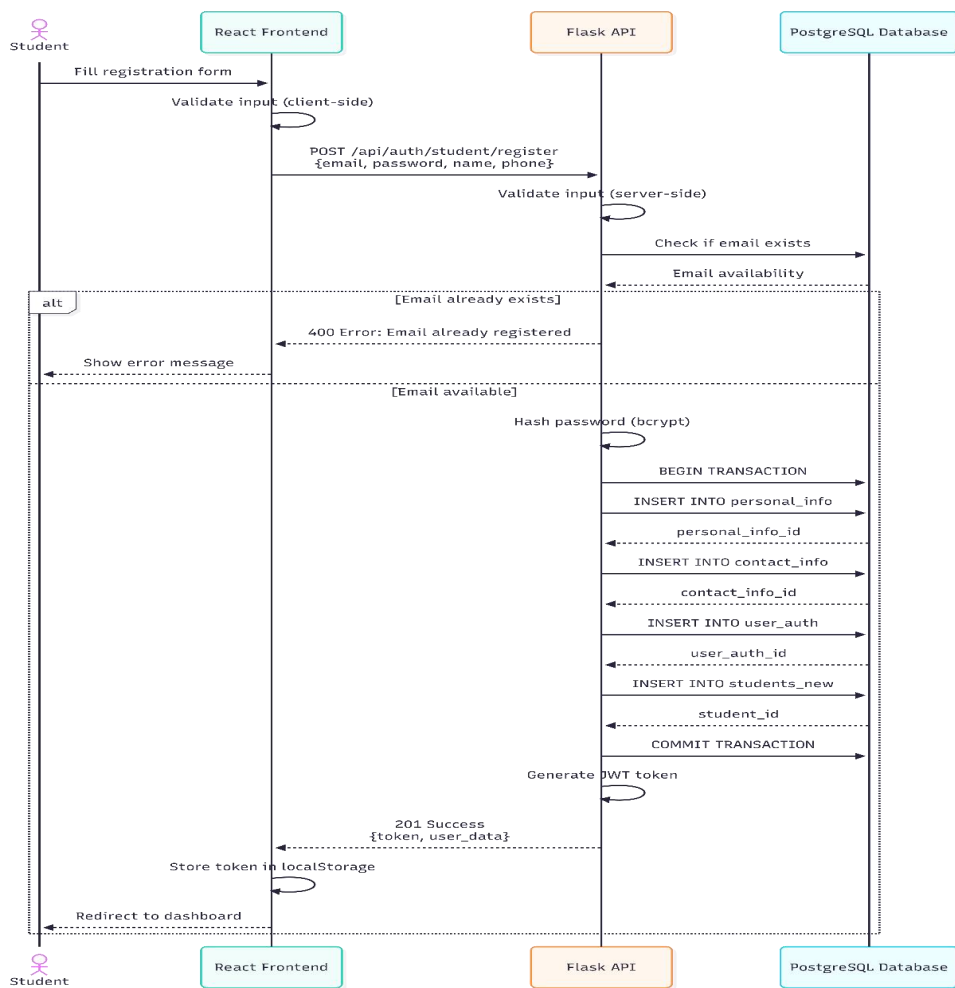


Figure 5.9: Student Registration Sequence Diagram

Description: This diagram illustrates the complete flow of student registration including validation, database operations, and JWT token generation.

Dakhla: AI-Assisted University Admissions Platform	Version: 1.0
Software Design Specifications	Date: 19/DEC/2025
FYP-020/FA25	

5.3.1.2 Student Login Sequence Diagram

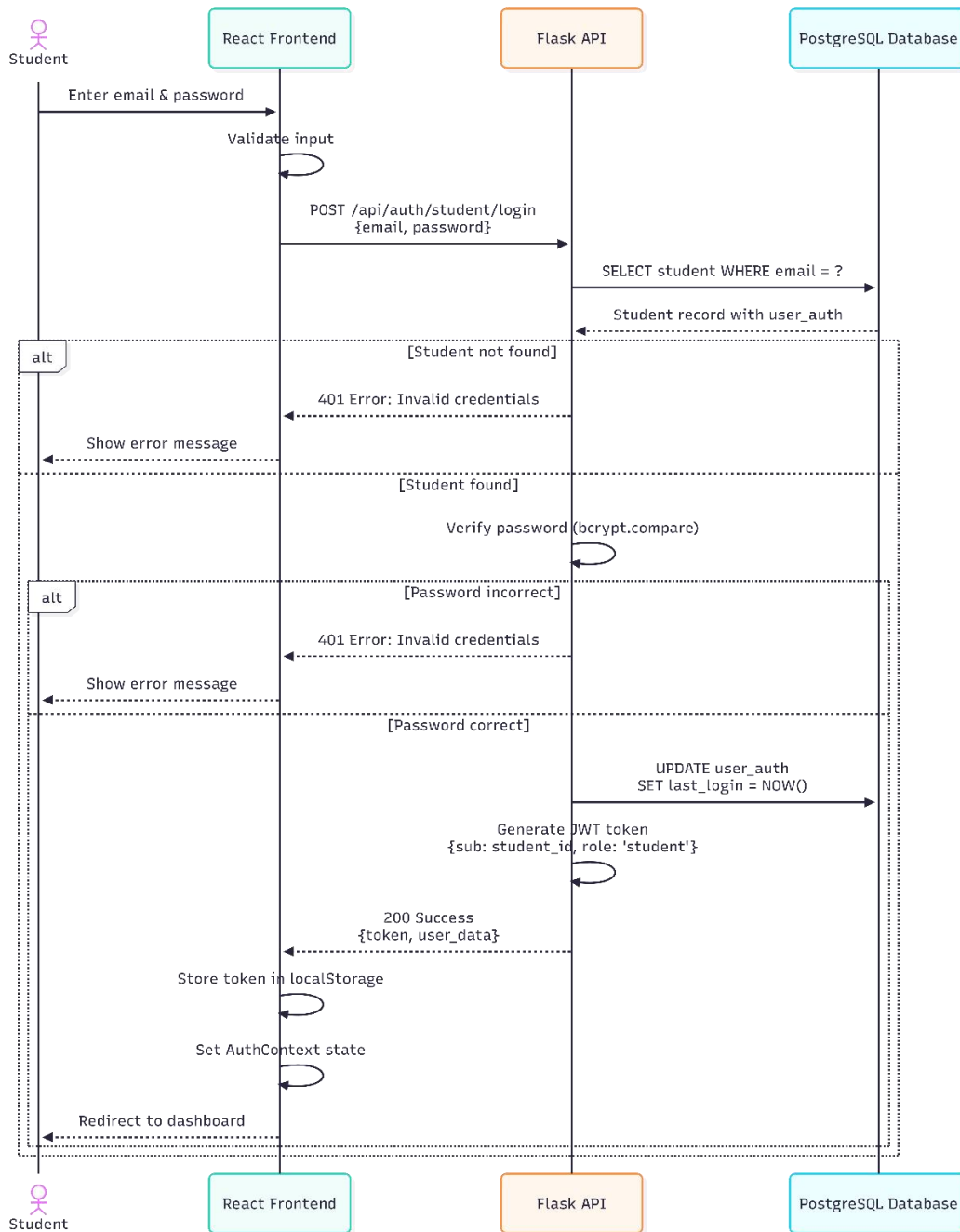


Figure 5.10: Student Login Sequence Diagram

Description: Shows the authentication process from credential submission to JWT token generation and storage.

Dakhla: AI-Assisted University Admissions Platform	Version: 1.0
Software Design Specifications	Date: 19/DEC/2025
FYP-020/FA25	

5.3.1.3 University Profile Creation Sequence Diagram

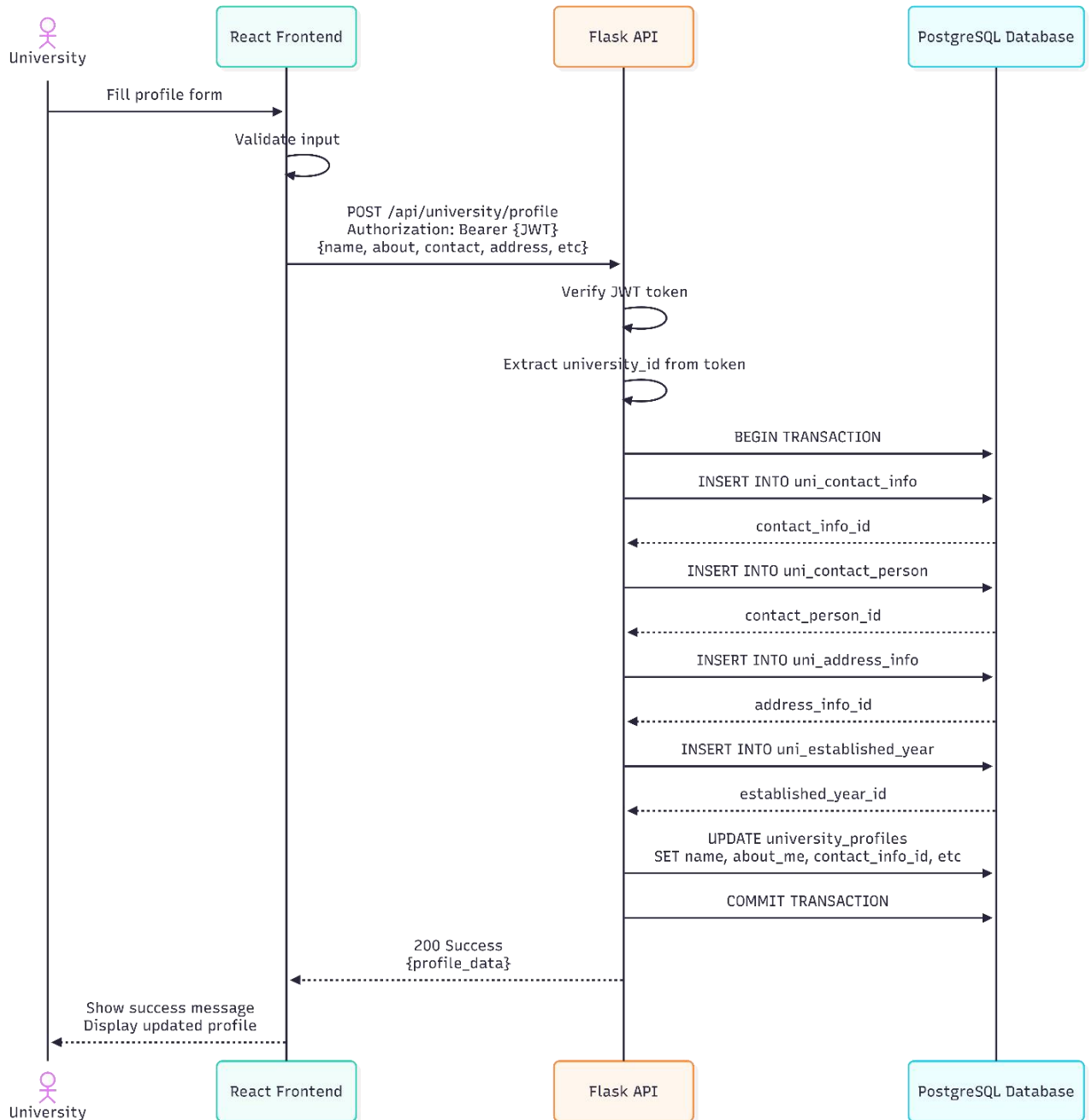


Figure 5.11: University Profile Creation Diagram

Description: This diagram illustrates the complete flow of University registration including validation, database operations, and JWT token generation.

Dakhla: AI-Assisted University Admissions Platform	Version: 1.0
Software Design Specifications	Date: 19/DEC/2025
FYP-020/FA25	

5.3.1.4 Program Search and Filter Sequence Diagram

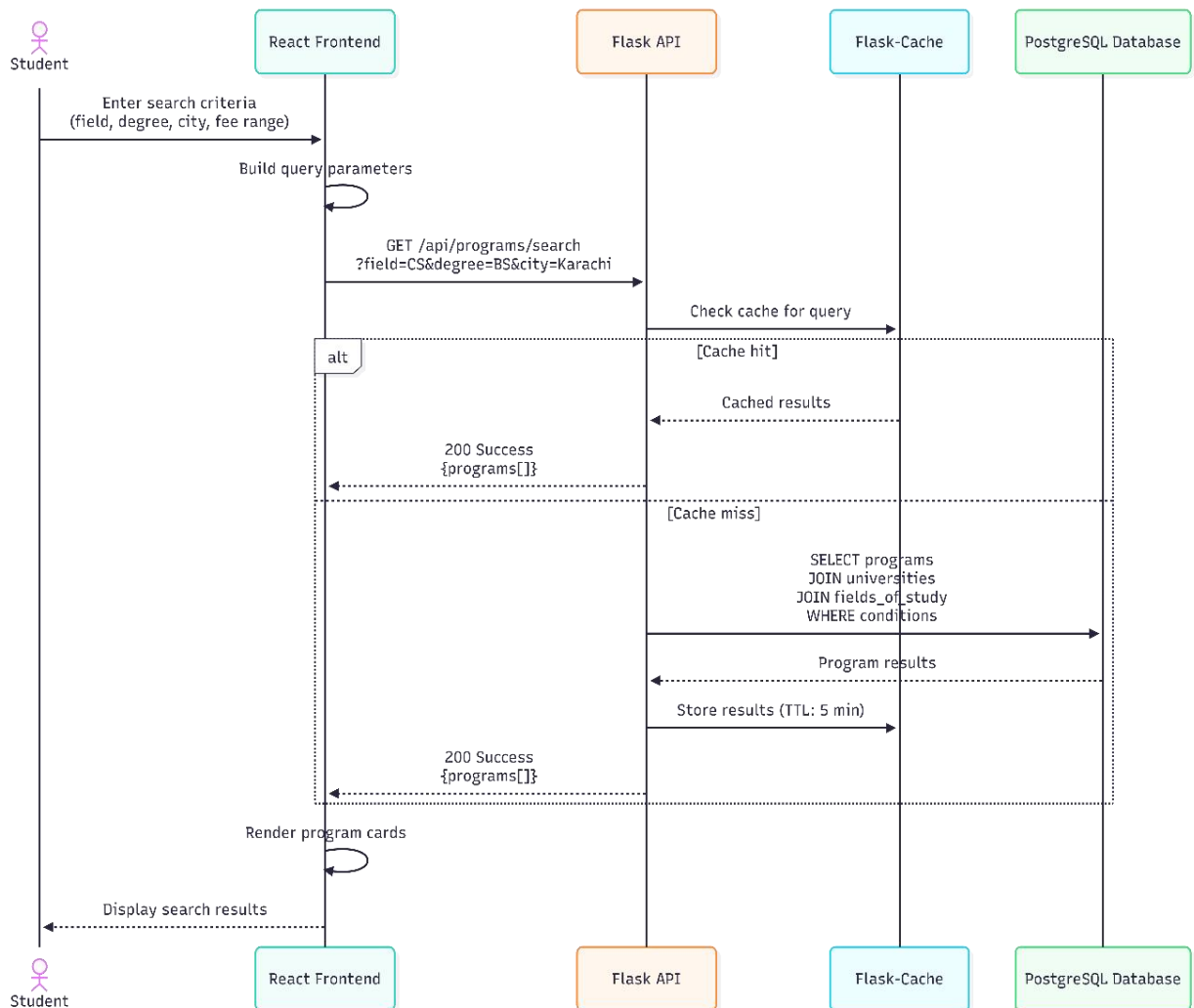


Figure 5.12: DirectApply Submission Sequence Diagram

Description: Demonstrates how search queries are processed, filtered, and results are returned to the user interface.

Dakhla: AI-Assisted University Admissions Platform	Version: 1.0
Software Design Specifications	Date: 19/DEC/2025
FYP-020/FA25	

5.3.1.5 Password Reset Sequence Diagram

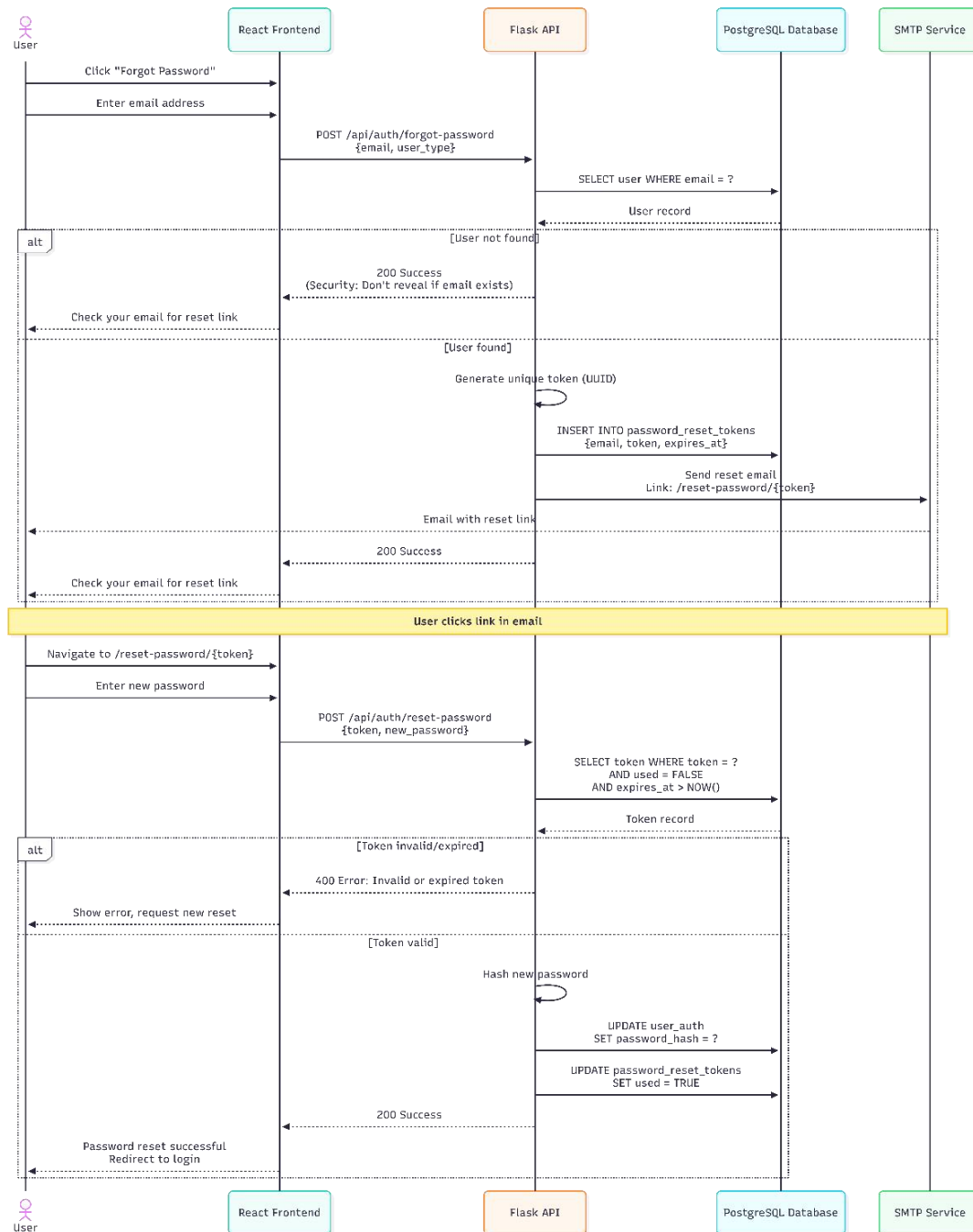


Figure 5.13: Password Reset Sequence Diagram

Description: Illustrates how student profiles are analyzed and matched with appropriate university programs.

Dakhla: AI-Assisted University Admissions Platform	Version: 1.0
Software Design Specifications	Date: 19/DEC/2025
FYP-020/FA25	

5.3.2 State Diagrams

State diagrams show the different states of an object and the transitions between those states.

5.3.2.1 Student Account State Diagram

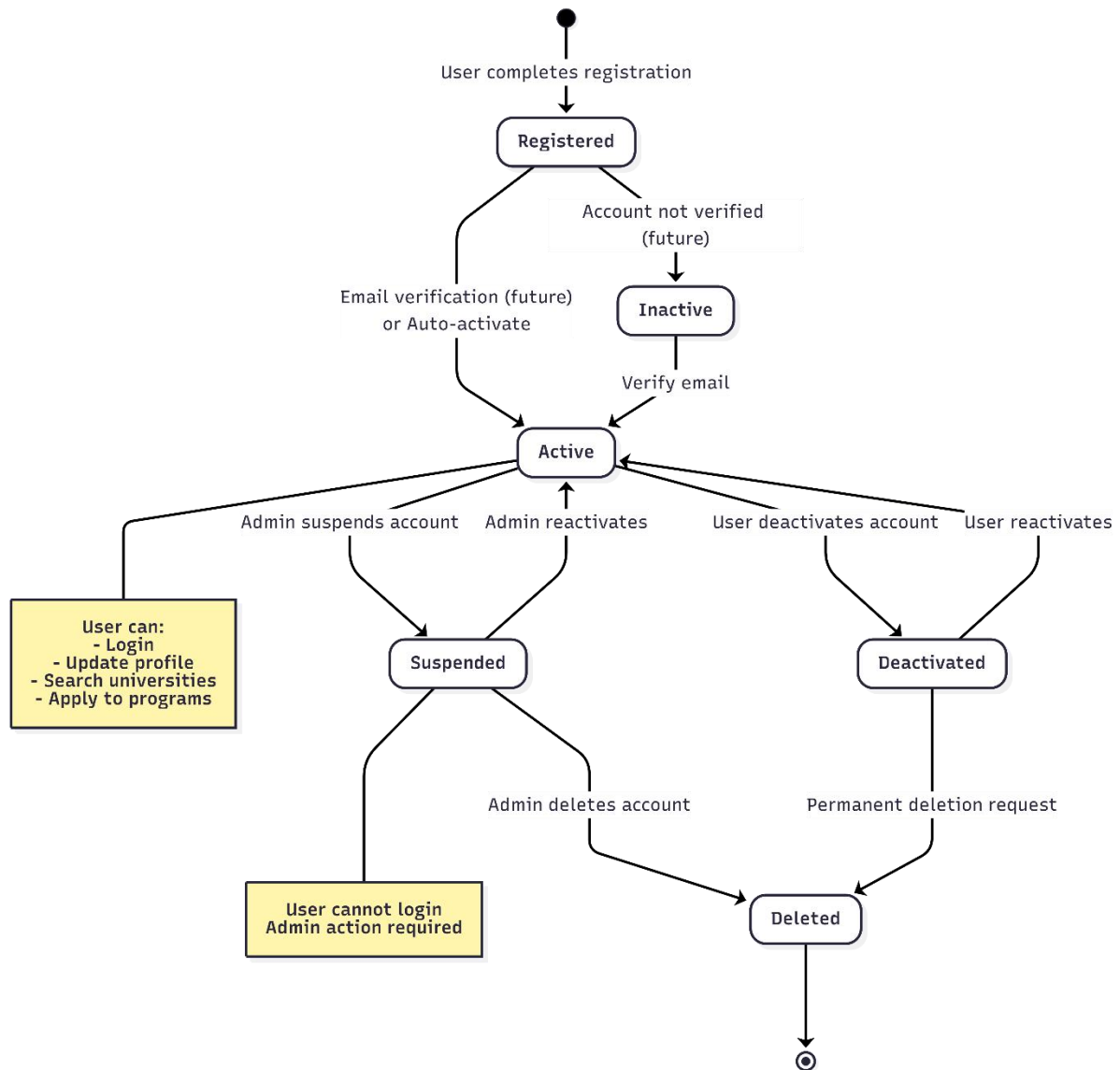


Figure 5.14: Student Account State Diagram

Description: Shows the various states of a student account (Unregistered, Registered, Active, Inactive, Suspended) and the transitions between them.

Dakhla: AI-Assisted University Admissions Platform	Version: 1.0
Software Design Specifications	Date: 19/DEC/2025
FYP-020/FA25	

5.3.2.3 Admin Profile State Diagram

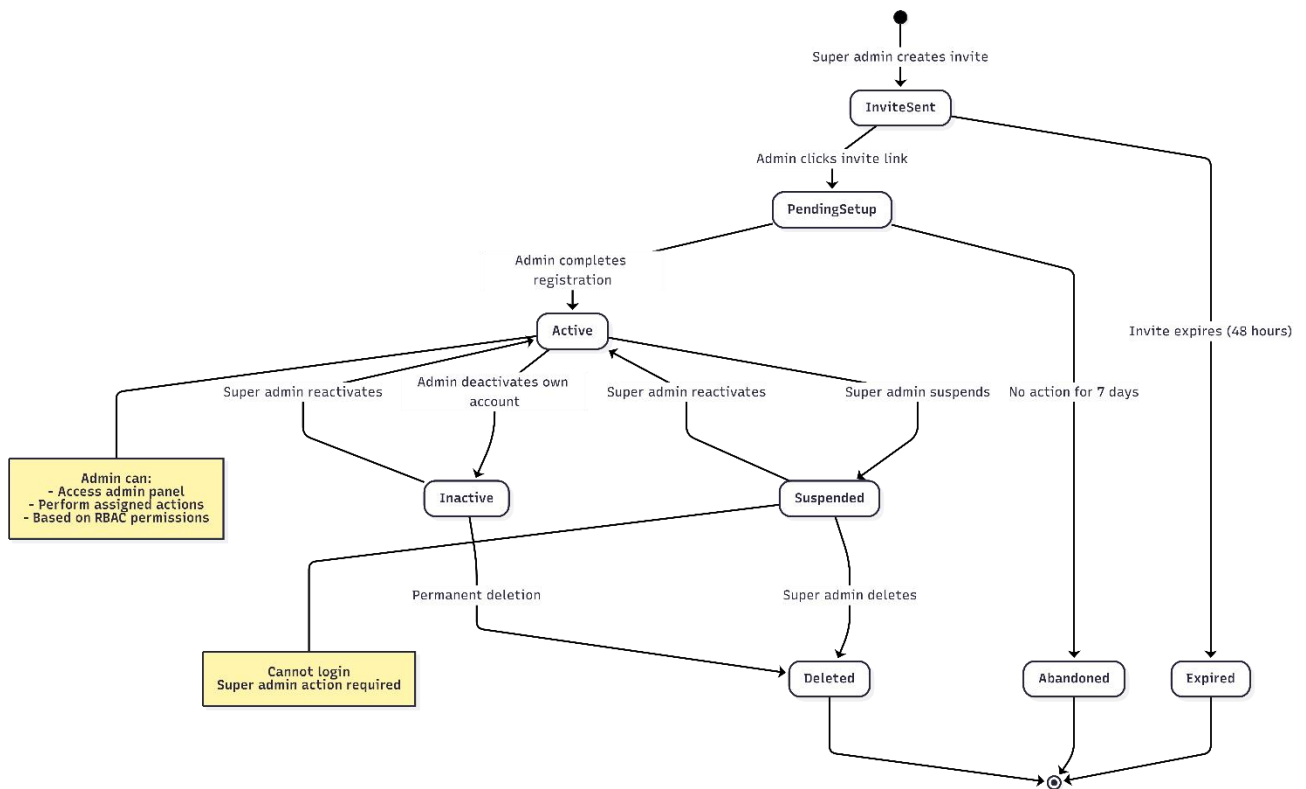


Figure 5.16: University Profile State Diagram

Description: Shows the states of a Admin Profile (Pending Verification, Active, Inactive, Suspended) and allowed transitions.

Dakhla: AI-Assisted University Admissions Platform	Version: 1.0
Software Design Specifications	Date: 19/DEC/2025
FYP-020/FA25	

5.4 GUI Design

This section presents the graphical user interface mockups for key use cases and screens in the Dakhla platform.

5.4.1 Student Landing Page

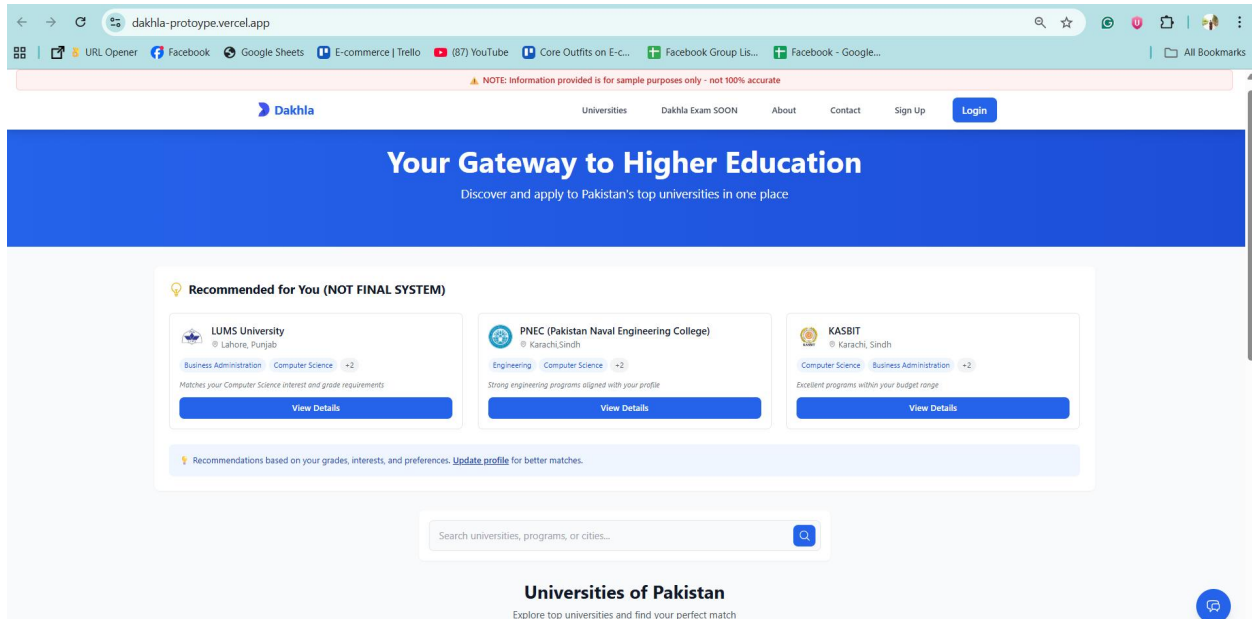


Figure 5.17: Student Landing Page Mockup

Description: The main landing page featuring hero section, search bar, featured universities, and call-to-action buttons.

Dakhla: AI-Assisted University Admissions Platform	Version: 1.0
Software Design Specifications	Date: 19/DEC/2025
FYP-020/FA25	

5.4.2 Student Registration Form

NOTE: Information provided is for sample purposes only - not 100% accurate

Dakhla Universities Dakhla Exam SDOH About Contact Sign Up **Login**

Create your account

Or sign in to your existing account

I am a

Student

Personal Information

Full Name * ismail

Email Address * ismaildakhlan@gmail.com
Must be a valid email format

Phone Number * 03202095258
11 digits starting with 03

Date of Birth * 18/06/2002

Gender * Male

Address Information

Address * Hino R 230, A one city, sector 1A/4, North karachi

City * Karachi

Postal Code * 75210
Exactly 5 digits

Academic Information

Field of Interest * Computer Science

Intermediate Marks (%) * 64

Security

Password *

Confirm Password *


Create Student Account



Figure 5.18: Student Registration Form Mockup


Description: Multi-step registration form collecting personal information, contact details, and academic background.

Dakhla: AI-Assisted University Admissions Platform	Version: 1.0
Software Design Specifications	Date: 19/DEC/2025
FYP-020/FA25	

5.4.3 Student Dashboard



Hamza Ali Khan
Student Dashboard


 Profile
 My Applications





Hamza Ali Khan
Student
Click the pencil icon to upload a profile picture (180x180px recommended)


Personal Information


 Full Name
Hamza Ali Khan


 Phone
03302888875

 Gender
Male


 Preferred Field of Study
Computer Science


 Email
hkalikhan12@gmail.com


 Date of Birth
April 26, 2001



 Intermediate Marks
51%

Address Information


 Address
R-75 SHAZ BANGLOWS GULZAR-E-HIJRI SCHEME-33

 City
Karachi

 Postal Code
75330

 Profile
 My Applications

My Applications



No Applications Yet
You haven't applied to any universities yet.
Browse universities and apply to programs that interest you!

Browse Universities

Figure 5.19: Student Dashboard

Description: Personalized dashboard showing saved universities and quick actions.

Dakhla: AI-Assisted University Admissions Platform	Version: 1.0
Software Design Specifications	Date: 19/DEC/2025
FYP-020/FA25	

5.4.4 University Search and Filter Page

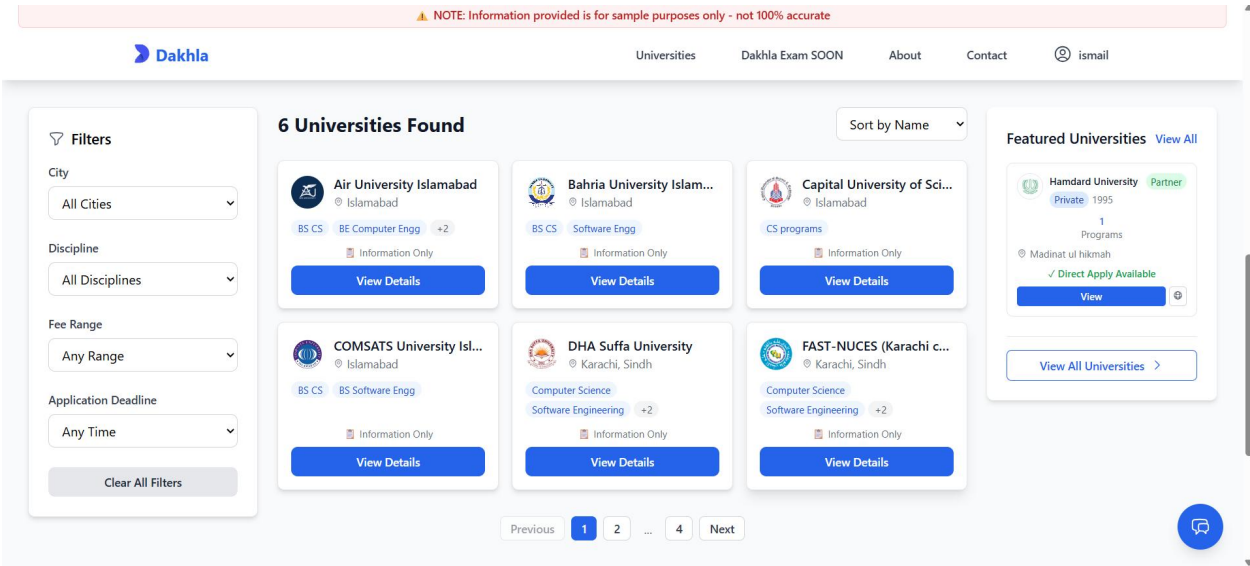


Figure 5.20: University Search and Filter Page

Description: Advanced search interface with multiple filter options (location, field, degree type, fees) and result grid.

Dakhla: AI-Assisted University Admissions Platform	Version: 1.0
Software Design Specifications	Date: 19/DEC/2025
FYP-020/FA25	

5.4.5 University Detail Page

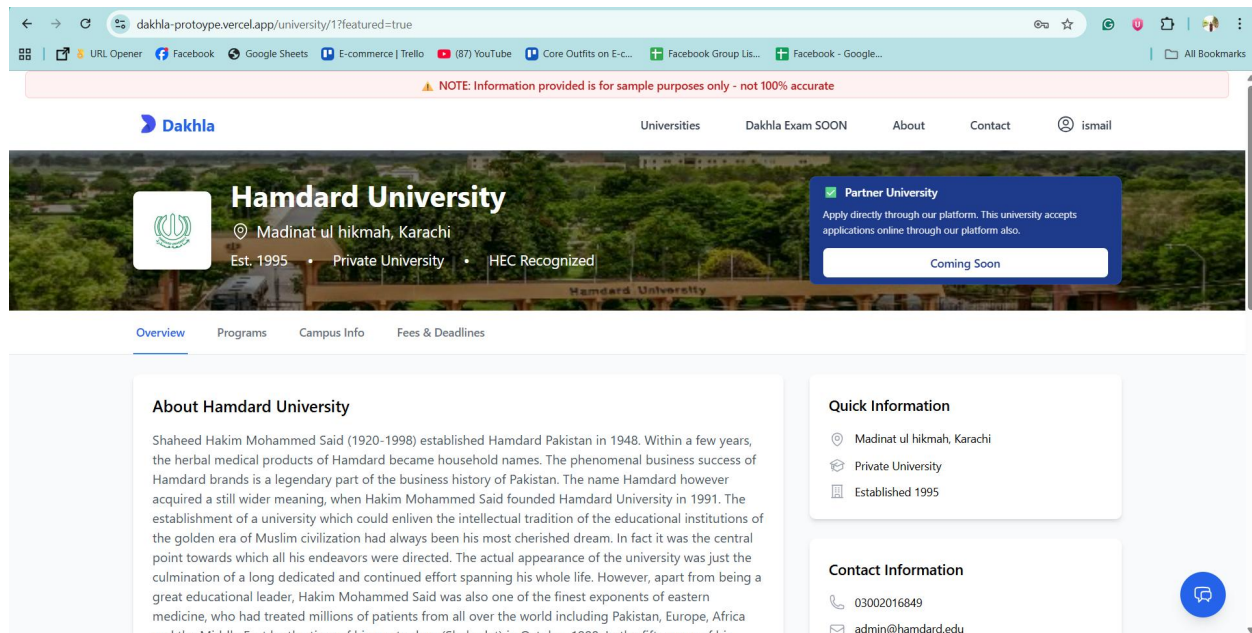


Figure 5.21: University Detail Page

Description: Comprehensive university profile displaying about section, programs offered, facilities, contact information, and application button.

Dakhla: AI-Assisted University Admissions Platform	Version: 1.0
Software Design Specifications	Date: 19/DEC/2025
FYP-020/FA25	

5.4.6 DakhlaBot Chatbot Interface

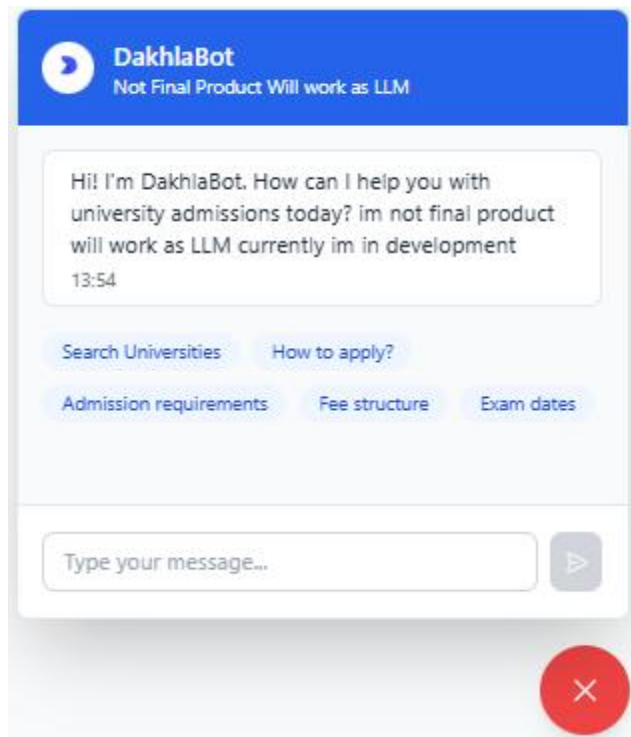



Figure 5.22: DakhlaBot Chatbot Interface

Description: AI-powered chat widget providing instant answers to admissions queries with suggested questions.

Dakhla: AI-Assisted University Admissions Platform	Version: 1.0
Software Design Specifications	Date: 19/DEC/2025
FYP-020/FA25	

5.4.7 University Dashboard




Hamdard University
University Dashboard
Contact: Hamza



University Info
Programs
Applicants

Programs Offered
Manage your university's academic programs

Advanced
+ Add Program

Current Programs (1)



(BS) Computer Science

A Computer Science degree focuses on the principles of computation and the design, development, and application of computer systems

Fee: PKR 600,000
Deadline: 30/09/2025

Active Available for applications




Hamdard University
University Dashboard
Contact: Hamza

University Info
Programs
Applicants

Student Applications
Review and manage student applications


All Programs
All Status



No Applications Received

When students apply to your programs, they will appear here.
You'll be able to review applications, approve or reject them.

Preview: How applications will appear



Student Name
Applied for: Program Name

Accept
Reject

Figure 5.23: University Dashboard

Description: University portal displaying incoming applications, program management, and analytics.

Page 69 of 71

Dakhla: AI-Assisted University Admissions Platform	Version: 1.0
Software Design Specifications	Date: 19/DEC/2025
FYP-020/FA25	

5.4.8 Admin Panel Dashboard

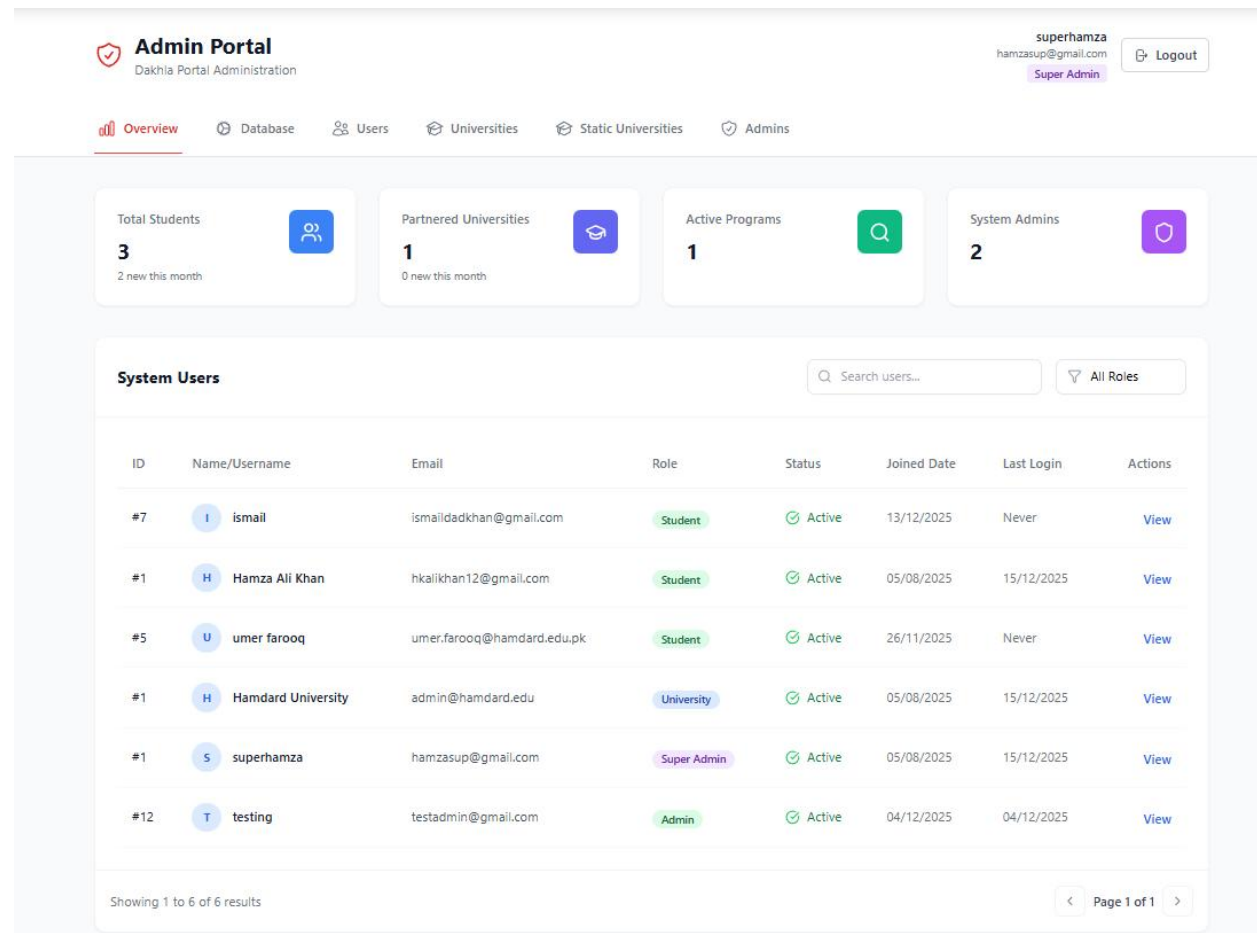


Figure 5.24: Admin Panel Dashboard

Description: Administrative interface for system management, user oversight, and platform analytics.

Dakhla: AI-Assisted University Admissions Platform	Version: 1.0
Software Design Specifications	Date: 19/DEC/2025
FYP-020/FA25	

References

- [1] Dakhla Project Team, "Software Requirements Specification (SRS) Document," 2025.
- [2] Meta Platforms, Inc., *React - A JavaScript Library for Building User Interfaces*, 2024.
- [3] Tailwind Labs Inc., *Tailwind CSS - Rapidly Build Modern Websites*, 2024.
- [4] Auth0, *JSON Web Tokens - Introduction*, 2024.
- [5] Pallets Projects, *Flask Web Development Framework*, 2024.
- [6] Pallets Projects, *Flask-SQLAlchemy Extension*, 2024.
- [7] The PostgreSQL Global Development Group, *PostgreSQL: The World's Most Advanced Open Source Relational Database*, 2024.
- [8] Higher Education Commission Pakistan, *HEC Official Website*, 2024.
- [9] OpenAI, *OpenAI API Reference*, 2024.
- [10] Vercel Inc., *Vercel Platform Documentation*, 2024.
- [11] Render Services, Inc., *Render Cloud Platform Documentation*, 2024.
- [12] Supabase, *Supabase - The Open Source Firebase Alternative*, 2024.
- [13] Flask-JWT-Extended, *Flask-JWT-Extended Documentation*, 2024.
- [14] Axios, *Axios - Promise Based HTTP Client*, 2024.
- [15] SQLAlchemy, *SQLAlchemy - The Database Toolkit for Python*, 2024.
- [16] M. Fowler, *Patterns of Enterprise Application Architecture*, Addison-Wesley Professional, 2002.
- [17] I. Sommerville, *Software Engineering*, 10th ed., Pearson Education Limited, 2016.

End of Document

Dakhla - Centralized, AI-Assisted University Admissions Platform
Software Design Specifications (SDS) - Version 1.0
© 2025 Hamdard University - Department of Computing