|  |  |  |
| --- | --- | --- |
| amblem | **ÇANKAYA UNIVERSITY**  **Software Engineering Department** | **A circular logo with colorful arrows  Description automatically generated** |

**SENG 491 – 492 Graduation Project**

**Software Design**

**Description**

**UAPCPS**

**Mehmet Deniz Kaya**

**Halil Berkay Güngör**

**Emircan Güneş**

**Osman Türker Köse**

**Version <<1.0>>**

İçindekiler

[1. Introduction 3](#_Toc186299151)

[1.1 Purpose 3](#_Toc186299152)

[1.2 Scope 3](#_Toc186299153)

[1.3 Definitions, Acronyms, and Abbreviations 3](#_Toc186299154)

[1.4 References 3](#_Toc186299155)

[1.5 Overview 3](#_Toc186299156)

[1.6 Version History 3](#_Toc186299157)

[2. Architecture 3](#_Toc186299158)

[2.1 Clients 3](#_Toc186299159)

[2.1.1 Personal Computers 3](#_Toc186299160)

[2.2 Application Server 4](#_Toc186299161)

[2.3 Database Server 4](#_Toc186299162)

[3. User Interface 5](#_Toc186299163)

[3.1 Login Screens 5](#_Toc186299164)

[3.1.1 Student Login 5](#_Toc186299165)

[3.1.2 University Login 6](#_Toc186299166)

[3.1.3 YÖK Administrator Login 7](#_Toc186299167)

[3.2 Dashboard Screens 7](#_Toc186299168)

[3.2.1 Student Dashboard 7](#_Toc186299169)

[3.2.2 University Dashboard 8](#_Toc186299170)

[3.2.3 YÖK Administrator Dashboard 9](#_Toc186299171)

[4. High-Level Design 9](#_Toc186299172)

[4.1 Module Descriptions 9](#_Toc186299173)

[4.1.1 Authentication Module 9](#_Toc186299174)

[4.1.2 Prediction Module 10](#_Toc186299175)

[4.1.3 Analytics Module 10](#_Toc186299176)

[4.2 System Architecture 10](#_Toc186299177)

[5. Low-Level Design 11](#_Toc186299178)

[5.1 Module Flows 11](#_Toc186299179)

[5.1.1 Authentication Flow 11](#_Toc186299180)

[5.1.2 Prediction Flow 12](#_Toc186299181)

[5.1.3 Analystic Flow 13](#_Toc186299182)

[6. Database Design 15](#_Toc186299183)

[6.1 E-R Diagram 15](#_Toc186299184)

[6.2 Tables 18](#_Toc186299185)

[6.2.1 User Table 18](#_Toc186299186)

[6.2.2 University Table 19](#_Toc186299187)

[6.2.3 Faculty Table 19](#_Toc186299188)

[6.2.4 Department Table 19](#_Toc186299189)

[6.2.5 City Table 19](#_Toc186299190)

[6.2.6 District Table 19](#_Toc186299191)

[7. References 20](#_Toc186299192)

# 1. Introduction

## 1.1 Purpose

The purpose of this SDD is to outline the design of the University Admission Prediction and Capacity Planning System (UAPCPS). This document is intended for developers, stakeholders, and administrators to understand the system’s architecture, user interface, and data management.

## 1.2 Scope

This SDD focuses on the architectural design, database structure, and user interface of the UAPCPS, ensuring that the system meets functional and non-functional requirements specified in the SRS. The system will operate primarily as a web-based platform for students, universities, and YÖK administrators.

## 1.3 Definitions, Acronyms, and Abbreviations

* **UAPCPS:** University Admission Prediction and Capacity Planning System.
* **SQLite:** Lightweight Database Management System.
* **YÖK:** Yükseköğretim Kurulu (Council of Higher Education in Turkey).

## 1.4 References

1. SRS for UAPCPS, Version 1.0.
2. Django Framework Documentation.
3. SQLite Official Documentation.

## 1.5 Overview

This document provides the system’s architectural design, user interface details, and database design. Each section describes a critical component of the system to ensure seamless implementation.

## 1.6 Version History

| **Version** | **Description** | **Date** |
| --- | --- | --- |
| 1.0 | Initial Release | 28.12.2024 |

# 2. Architecture

## 2.1 Clients

### 2.1.1 Personal Computers

* **Minimum Requirements:**
  + **Operating System:** Windows, macOS, or Linux.
  + **Browser:** Latest versions of Chrome, Firefox, or Safari.
  + **Hardware:** Dual-core processor, 4 GB RAM, 100 MB free disk space.

## 2.2 Application Server

* **Framework:** Django Web Framework.
* **Programming Language:** Python 3.12+.
* **Key Components:** Handles business logic, user authentication, and communication with the database.

## 2.3 Database Server

* **Database Management System:** SQLite.
  + **Advantages:**
    - Lightweight and serverless.
    - Ideal for development and small-scale production environments.
  + **Storage Format:** Single file (db.sqlite3).
  + **Limitations:**
    - Limited concurrent write operations.
    - Not recommended for high-scale, high-traffic environments.

# 3. User Interface

## 3.1 Login Screens

### 3.1.1 Student Login

* metin, ekran görüntüsü, ağaç, web sayfası içeren bir resim

  Açıklama otomatik olarak oluşturuldu**Description:** Students log in using email and password or via Google.

**Title Student Sign-In Screen**

1. **Email Text input field for the user's email address.**
2. **Password Secure text input field for the user's password.**
3. **Remember Me Checkbox option to remember the login credentials on the current device.**
4. **Sign In Button to authenticate the user's credentials and redirect to the student dashboard upon successful login.**
5. **Forgot Your Password? A link for students to reset their password if forgotten.**
6. **Sign In with Google Button for single sign-on (SSO) with Google.**
7. **Sign Up Link to the registration page for students who do not yet have an account.**

### 3.1.2 University Login

* **metin, ekran görüntüsü, gökyüzü içeren bir resim

  Açıklama otomatik olarak oluşturulduDescription:** Universities log in with their institutional email and password.

**Title University Sign-In Screen**

1. **Email Text input field for the university's registered email address.**
2. **Password Secure text input field for the university's password.**
3. **Sign In Button to authenticate the university's credentials and redirect to the university dashboard upon successful login.**
4. **Sign Up Button or link to direct new universities to the registration page for account creation.**

### 3.1.3 YÖK Administrator Login

* **Description:** YÖK administrators log in using a pre-assigned unique ID and password.

**Title Sign In**

1. **ID Field Text input field for YÖK ID. The initial value is empty.**
2. **Password Field Password input field. The initial value is empty.**
3. **Sign In Button Submits the ID and password for authentication.**

**metin, ekran görüntüsü, yazı tipi, logo içeren bir resim

Açıklama otomatik olarak oluşturuldu**

## 3.2 Dashboard Screens

### metin, ekran görüntüsü, yazılım, grafik tasarım içeren bir resim Açıklama otomatik olarak oluşturuldu3.2.1 Student Dashboard

**1. Student Information Input Section (Left Side):**

**Score Type: The user selects the type of score used for predictions (e.g., Science (SAY), Equally Weighted (EA), Verbal (SÖZ)).**

**Score: The user enters their exam score in this field.**

**Ranking: The user specifies their exam ranking.**

**Year: The user selects the year for which they want to make predictions.**

**University and Department: The user chooses the university and department they are interested in.**

**Predict Button: After filling in all the fields, the user clicks this button to see the prediction results.**

**2. Result Section (Right Side):**

**Graph: Displays the user's probability of being admitted to the selected department as a pie chart.**

**Recommendations: The system provides suggestions for similar programs or alternative choices based on the user's preferences and ranking.**

**Feedback Section: Users can share their feedback about the predictions:**

**Successful/Unsuccessful: Users can indicate whether they found the results satisfactory or not by clicking the respective buttons.**

### 3.2.2 University Dashboard

metin, yazı tipi, öykü gelişim çizgisi; kumpas; grafiğini çıkarma, çizgi içeren bir resim

Açıklama otomatik olarak oluşturuldu

**1. Department Overview Table (Left Section):**

**Displays all departments associated with the university.**

**Columns:**

**Department ID: Unique identifier for each department.**

**Department Name: Name of the department (e.g., Computer Engineering).**

**Score Type: Indicates the score category required for admission (e.g., SAY).**

**Capacity: Shows the total number of seats available for the department.**

**Enrollment: Number of students currently enrolled.**

**Gender Enrollment: Breakdown of student enrollment by gender (e.g., Male/Female).**

**Field Rate: Displays the fill rate percentage for the department.**

**The table allows administrators to quickly review department-specific data, providing a summary of capacity, enrollment trends, and other relevant metrics.**

**2. Capacity and Admission Analysis Chart (Right Section):**

**A bar chart visualizes the projected gender-based admission ratios for the next academic year for selected departments.**

**Each bar represents a department with the distribution of male and female admission rates.**

**The chart helps university administrators:**

**Analyze gender trends in specific departments.**

**Identify departments with underutilized or overutilized capacities.**

**Plan adjustments to quotas based on projected trends.**

### 3.2.3 YÖK Administrator Dashboard

* **Features:**
  + Aggregate analytics across universities.
  + Recommendations for quota adjustments.
  + Program performance evaluations.

# 4. High-Level Design

## 4.1 Module Descriptions

### 4.1.1 Authentication Module

**Description: This module manages user authentication and authorization. It ensures secure login, registration, and role-based access control for different user types.**

**Key Features:**

**User login and registration (via email or Google).**

**Password reset functionality.**

**Role-based access control (Students, Universities, YÖK Administrators).**

**Relationships: Shares authenticated user information with the Prediction Module and the Analytics Module.**

### 4.1.2 Prediction Module

**Description: Implements AI models to predict university admission probabilities and recommend suitable programs based on the user's preferences and ranking.**

**Key Features:**

**Processes user input (ranking, program preferences) and historical data.**

**Uses machine learning models to predict admission chances.**

**Provides program recommendations for students with lower rankings.**

**Relationships: Receives user input from the Authentication Module and shares prediction results with the Analytics Module.**

### 4.1.3 Analytics Module

**Description: Offers advanced visualizations and insights into data trends, program popularity, and capacity utilization rates.**

**Key Features:**

**Visualizes trends using charts, graphs, and tables.**

**Provides insights for university administrators on enrollment and program performance.**

**Displays interactive dashboards for students and universities.**

**Relationships: Retrieves prediction data from the Prediction Module and user access data from the Authentication Module.**

## 4.2 System Architecture

**System Layers:**

**Frontend (User Interface Layer):**

**Provides interfaces for Students, Universities, and YÖK Administrators.**

**Contains dashboards for users to view predictions, trends, and reports.**

**Backend (Business Logic Layer):**

**Handles data processing, prediction algorithms, and access control.**

**Manages interactions between modules and ensures smooth data flow.**

**Database Layer:**

**Stores historical admission data, user credentials, program details, and prediction results.**

**Uses SQLite as the database management system.**

# 5. Low-Level Design

## 5.1 Module Flows

### 5.1.1 Authentication Flow

**Explanation:**

**Start:**

**The authentication process begins when a user interacts with the login or registration interface.**

**User Type Selection:**

**Users are categorized into three groups:**

**Students**

**Universities**

**YÖK Administrators**

**Student Authentication:**

**Students can log in with their email and password or register to create a new account.**

**For login:**

**Credentials are validated. If valid, the student is redirected to their dashboard. If invalid, an error message is shown, and they can retry.**

**For registration:**

**New account details are saved, and the student is redirected to their dashboard.**

**University Authentication:**

**Universities follow a similar flow, with login and registration options.**

**Credentials are validated for login. If invalid, the process allows retries. Upon registration, new account details are saved.**

**YÖK Authentication:**

**YÖK administrators can only log in using predefined credentials. There is no registration option for this user group.**

**If the credentials are valid, the admin is redirected to the YÖK dashboard. Otherwise, an error message is displayed, and retries are allowed.**

**End:**

**The process concludes when the user is authenticated and redirected to their respective dashboard.**

ekran görüntüsü, diyagram, 3B modelleme içeren bir resim

Açıklama otomatik olarak oluşturuldu

### 5.1.2 Prediction Flow

**Explanation**

**User Type Decision:**

**Users are first divided into Students, Universities, or YÖK admins based on their role.**

**Students:**

**Focused on inputting details for predictions, validating input, processing predictions, and displaying results with suggestions.**

**Universities:**

**Involves analyzing historical data, validating input, running capacity analysis models, and generating insights.**

**YÖK:**

**Aggregates data from universities, processes it, performs trend analysis, and generates national-level reports.**

**Unified Endpoints:**

**Each workflow ends in a corresponding "End" node, keeping the structure modular and comprehensive.**

ekran görüntüsü, metin, diyagram, 3B modelleme içeren bir resim

Açıklama otomatik olarak oluşturuldu

### 5.1.3 Analystic Flow

**Key Steps in the Analytics Flow:**

**Data Collection:**

**Inputs are collected separately for Students, Universities, and YÖK to serve their respective analytics needs.**

**Validation and Preprocessing:**

**Each type of data is validated and preprocessed to ensure the analytics engine receives clean and usable inputs.**

**Central Analytics Engine:**

**A single analytics engine processes all the data, applying relevant models and calculations based on the user type and context.**

**Insight Generation:**

**Student insights focus on admission probabilities and program suggestions.**

**University insights target capacity trends, popular programs, and program performance.**

**YÖK insights revolve around national trends and strategic recommendations.**

**Display and Feedback:**

**Results are displayed to each user type in a tailored manner, with feedback mechanisms to refine the analytics process.**

metin, ekran görüntüsü, diyagram, 3B modelleme içeren bir resim

Açıklama otomatik olarak oluşturuldu

# 6. Database Design

## 6.1 E-R Diagram

**Entity-Relationship Diagram Description**

**The Entity-Relationship Diagram (ERD) represents the database structure for the University Admission Prediction and Capacity Planning System (UAPCPS). It defines the relationships between various entities and their attributes to support functionalities such as student predictions, university capacity analysis, and scholarship planning. Below are the detailed descriptions of the entities and relationships:**

**1. Entities and Attributes:**

**myapp\_university:**

**Represents universities in the system.**

**Attributes:**

**id: Unique identifier for each university.**

**name: Name of the university.**

**type: Type of university (e.g., public or private).**

**myapp\_faculty:**

**Represents faculties within universities.**

**Attributes:**

**id: Unique identifier for each faculty.**

**name: Name of the faculty.**

**university\_id: Foreign key linking to the associated university.**

**myapp\_department:**

**Represents departments within faculties.**

**Attributes:**

**id: Unique identifier for each department.**

**name: Name of the department.**

**education\_language: Language of instruction in the department.**

**city\_id: Foreign key linking to the city where the department is located.**

**district\_id: Foreign key linking to the district.**

**faculty\_id: Foreign key linking to the associated faculty.**

**myapp\_year:**

**Represents academic years in the system.**

**Attributes:**

**id: Unique identifier for each year.**

**myapp\_scoretype:**

**Represents the types of scores used for university admissions.**

**Attributes:**

**id: Unique identifier for each score type.**

**name: Name of the score type (e.g., SAY, EA, SOZ).**

**myapp\_scholarshiptype:**

**Represents different types of scholarships offered.**

**Attributes:**

**id: Unique identifier for each scholarship type.**

**name: Name of the scholarship type (e.g., full, partial).**

**myapp\_quotatype:**

**Represents the types of quotas available for departments.**

**Attributes:**

**id: Unique identifier for each quota type.**

**name: Name of the quota type (e.g., general, special).**

**myapp\_placementanalysis:**

**Core table representing the analysis of placements.**

**Attributes:**

**id: Unique identifier for each analysis.**

**department\_id: Foreign key linking to the associated department.**

**faculty\_id: Foreign key linking to the associated faculty.**

**university\_id: Foreign key linking to the associated university.**

**quota\_type\_id: Foreign key linking to the type of quota.**

**scholarship\_type\_id: Foreign key linking to the type of scholarship.**

**score\_type\_id: Foreign key linking to the type of score.**

**year\_id: Foreign key linking to the academic year.**

**2. Relationships:**

**University to Faculty:**

**A one-to-many relationship where a university can have multiple faculties.**

**Faculty to Department:**

**A one-to-many relationship where a faculty can have multiple departments.**

**Department to Placement Analysis:**

**A one-to-many relationship where a department can be analyzed for placement in various contexts.**

**City and District:**

**Represents the geographic attributes of departments, linking them to their locations.**

**Quota, Scholarship, and Score Types:**

**Placement analysis is enriched by associating each entry with a specific quota type, scholarship type, and score type.**

**Year:**

**Each placement analysis entry is tied to a specific academic year, enabling temporal analysis.**

**Purpose of the ERD:**

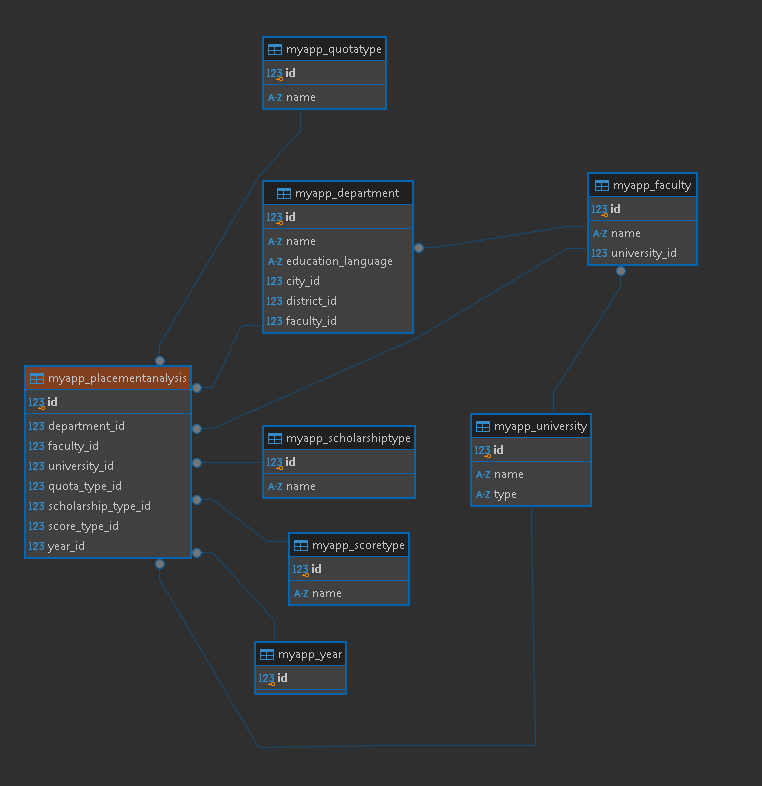
**This ERD ensures that the system:**

**Supports detailed analysis of university admissions and capacity planning.**

**Facilitates predictions based on historical data, including score types, quotas, and scholarships.**

**Manages hierarchical relationships between universities, faculties, and departments.**

**Enables tracking of student placement and capacity trends across years.**



## 6.2 Tables

### 6.2.1 User Table

| **Attribute** | **Type** | **Description** |
| --- | --- | --- |
| id | Integer (PK) | Unique user ID. |
| email | Text | User email. |
| password | Text | Encrypted password. |
| role | Text | Role (student/university/YÖK). |

### 6.2.2 University Table

| **Attribute** | **Type** | **Description** |
| --- | --- | --- |
| id | Integer (PK) | Unique university ID. |
| name | Text | University name. |
| type | Text | Public/Private/Other. |

### 6.2.3 Faculty Table

| **Attribute** | **Type** | **Description** |
| --- | --- | --- |
| id | Integer (PK) | Unique faculty ID. |
| university | Foreign Key | Links to University. |
| name | Text | Faculty name. |

### 6.2.4 Department Table

| **Attribute** | **Type** | **Description** |
| --- | --- | --- |
| id | Integer (PK) | Unique department ID. |
| faculty | Foreign Key | Links to Faculty. |
| name | Text | Department name. |
| education\_lang | Text | Language of instruction. |

### 6.2.5 City Table

| **Attribute** | **Type** | **Description** |
| --- | --- | --- |
| id | Integer (PK) | Unique city ID. |
| name | Text | City name. |

### 6.2.6 District Table

| **Attribute** | **Type** | **Description** |
| --- | --- | --- |
| id | Integer (PK) | Unique district ID. |
| city | Foreign Key | Links to City. |
| name | Text | District name. |

# 7. References

1. Django Documentation - <https://docs.djangoproject.com/>
2. SQLite Documentation - <https://www.sqlite.org/>
3. [Home | Mermaid Chart](https://www.mermaidchart.com/)
4. LLM Tools
5. [DBeaver Community | Free Universal Database Tool](https://dbeaver.io/)