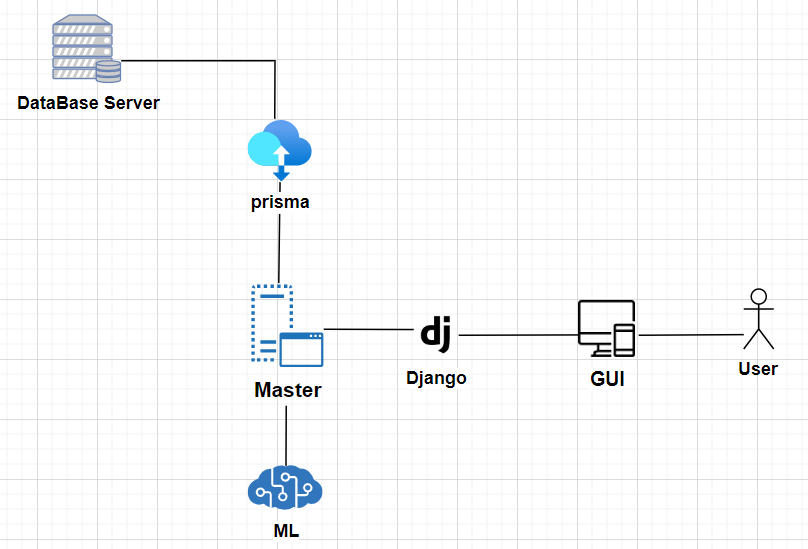
**Detailed Design**

**Physical Architecture**

****

**GUI/ Frontend**

The GUI is the visual gateway to our website, it is a seamless and user-friendly experience. It allows accepted students to submit their academic achievements via a user-friendly form and allows interested students to search for admission requirements based on school, degree, and academic grades by using a search toolbar. Through intuitive design and interactive features, the GUI empowers users to navigate the website effortlessly, access relevant information, and interact with the system's functionalities effectively.

**Django**

Django serves as the backend framework for our website, it powers the server-side logic. Within our Django application, we can implement the necessary API endpoints for handling form submissions from accepted students, processing admission requirement queries, and interfacing with the machine learning component to retrieve predictive insights.

**Master Server**

The master server serves as the central coordinating component of our website, it manages various tasks such as user authentication, data routing, communication between different components, and database interactions. It handles the flow of data between the frontend GUI, backend Django application, machine learning components, and the database server, ensuring smooth operations and synchronization of all system functionalities.

**Machine Learning (ML) Server**

The machine learning component plays a key part in our website's functionality by leveraging predictive analytics to forecast acceptance rates for prospective students. By training ML models on historical data collected from accepted students, the system can analyze patterns and correlations to predict the likelihood of admission for new applicants.

**Prisma**

Prisma acts as the intermediary between our backend application and the database server, providing a convenient tool for accessing and manipulating database records. This streamlines database interactions within our backend code, simplifies tasks such as data retrieval, insertion, and updates. This enables smooth data processing for our machine learning tasks and admission requirement queries.

**Database Server**

The database server serves as the foundational storage component for our website. It houses all the data collected from accepted students, including their personal information, academic records, and application details. This data is crucial for training the machine learning models and for providing personalized admission requirements based on school, degree, or grades.

**Component Diagram**

GUI/ Frontend

Master Server

Database Server

Frontend Requests

Queries & commands

Prediction Requests

ML Server

**Frontend**

1. Submission Page - This page allows past students to submit their academic details.
2. Search Tool Page - Users can utilize this page to search for acceptance requirements based on various criteria such as degree type, institution, or required grades.
3. Results Page - Upon submitting a search query, users are directed to this page, where they can view a list of degree programs that match their criteria.
4. Detailed Results Page - This page provides additional details about the admission requirements for the school or degree selected from the search results page and the probability of acceptance.

**Master Server**

1. Submission Handling - The backend will be responsible for processing, authenticating, and storing user submissions.
2. Search Query Processing – The backend will execute search queries received from the frontend, retrieve relevant data from the database, and return search results to the user interface.
3. Data Retrieval for Further Information - When users request additional information about the admission requirements for specific degrees or schools, the backend will retrieve the relevant data from the database and Machine Learning Server and format it for its presentation on the frontend.

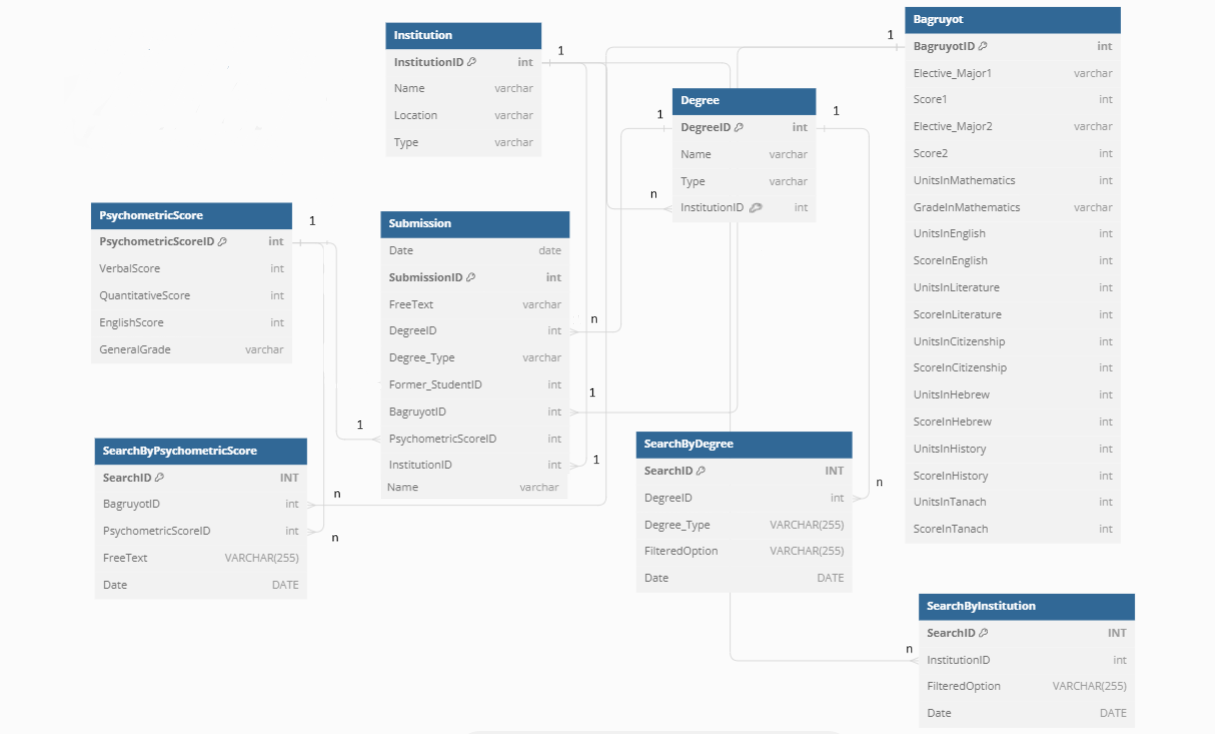
**Database Server**

1. Submission Tables - These tables store user submissions.
2. Institution Table - Contains basic information about institutions.
3. Degree Table - Stores data about degree programs offered by institutions.
4. Search History Tables - These tables record user search activity.

**Machine Learning Server**

1. Model Hosting & Execution - The ML server will host and execute the machine learning models responsible for predicting admission probabilities and extracting important features in the admission process.
2. Integration with the Master server - It will receive data requests from the master server for prediction or feature extraction tasks and return the results back to the master.

**ERD**



**ERD Explanation**

In our database, we'll have several key tables to manage our data. Our student submissions table will house vital information related to each student's submission, including their bagruyot scores, psychometric scores, and additional details contributing to their admission to a school and degree program. To streamline our database structure, we've opted to store bagruyot scores and psychometric scores in separate, more detailed tables.

Additionally, we'll maintain separate tables for institution data and information about the degrees offered by each institution. These tables will contain basic information and ensure ease of access when retrieving relevant data.

We will track user searches in three distinct tables. This approach involves recording the search date, specific information sought, and any applied filters. By doing so, we can effectively monitor searched data and trends over time, providing valuable insights into consistently sought-after topics. This information will guide us in optimizing our website to better meet user needs and preferences.

**UML** Diagrams of Use Cases

1.

Prospective student

Search for admission requirements

Views Search Results

Views Detailed Search Results Information

Refines Search

Finish

2.

Previously Accepted Student

Fills Submission Form with Admission Criteria

Information Validated

Data Analysis &

Displayed

Stores Submitted Data

3.

Admission Compass Admin

Retrieves Submitted Data from the Database

Identifies areas for improvement

Implements Improvements to the Website

Verifies Improvements

**Algorithms**

**Predicting Acceptance Algorithm**

This algorithm utilizes machine learning techniques such as cosine similarity, clustering, and linear regression to predict the probability of acceptance into a program based on a potential student’s grades. It starts by using the submission data of past applicants. Then, it preprocesses the data, which includes cleaning and transformation, the machine learning models are then trained using the techniques mentioned. Once trained, the models will predict the probability of acceptance for new applicants. Then we will calculate a weighted average from the predictions and generate an overall prediction score. This algorithm will aid potential students in making informed decisions about their application success rates.

**Academic Feature Importance Algorithm**

This algorithm utilizes the trained models from the previous algorithm. It derives the most important features for each school or degree, by using feature selection like SHAP values. Then we will combine the SHAP results from each model by using a voting algorithm to assess the most important features. These insights will help prospective students understand which requirements are most important to each school and degree, and advise students on which academic achievements they preferably need in order to be accepted.

**Functionality (continuation of functionality requirements)**

**Student Authentication Function**

This function authenticates users who are submitting their data through our accepted students form, and verifies that they are indeed students. Initially, it checks if the domain of the provided email address corresponds to one of our accepted institutions. If validated, a randomly generated code is sent to the email address for confirmation. Upon receiving the code, the student submits it on our website. Once successfully submitted, the student gains access to input their previously accepted grades into our form. This process ensures that our dataset, which will be utilized for predicting future student admission chances, is composed of genuine student submissions.

**Submission Tracking Function**

This function uses the collected submissions data from past students. It will calculate the frequency of submissions over time and evaluate the influxes of user engagement. These insights will help us enhance our user engagement and overall optimize our business.

**Search Behavior Analysis Function**

This function analyzes past searches from the database to better understand our user behavior. It analyzes search dates, filter usage, institutions and degrees searched. We will calculate the frequency of each item to determine the popular choices. This analysis will provide insights into how we can enhance the search functionality and user experience continually.

When a user makes a search based on institution, degree, or grades, we will present their results in various formats on the detailed results pages, so that the information and insights are digestible and clear for users. To do this we will implement the following algorithms:

**Calculating Averages, Ranges, and Distribution Functions**

This function computes the average, range, and distribution of different scores for each academic criteria for each institution or degree. To calculate the averages, it aggregates scores of particular criteria by summing them up and dividing them by the total number of submissions. Score ranges are determined by finding the minimum and maximum values within each criterion. For the distribution we get the frequency of scores falling within specific intervals or bins for each criterion.

תמונה שמכילה צילום מסך, טקסט, קו, גופן

התיאור נוצר באופן אוטומטי

**Aggregating Acceptance Requirements Function**

This function is designed to showcase the academic requirements that lead to the acceptance of past students by institutions and specific degrees. It starts by identifying all possible combinations of acceptance criteria. Next, it tallies the occurrences of each combination among the accepted submissions and computes the average scores for each combination. The results are then presented to the user via a pie chart, illustrating the distribution of the various combinations of acceptance requirements.

תמונה שמכילה טקסט, צילום מסך, גופן, תרשים

התיאור נוצר באופן אוטומטי