GTA Application Portal

Architecture/Design Document

**Table of Contents**

[**1 Introduction**](#) **2**

[**2 Architecture Style**](#) **2**

[2.1 Architecture Diagram](#_3p0luhepf6cy) 2

[2.2 Architecture Description](#_gmcdagj0neqf) 2

[**3 Tier Classification**](#) **2**

[3.1 User Interface Tier](#_dhqhfa2h7h8o) 2

[3.2 Logic Tier](#_v1jj16oci2ur) 3

[3.3 Database Tier](#_my89d06ccmh7) 3

[**4 Class Diagrams**](#) **4**

[**5 Database Schema**](#_wtk6tnqi7gyp) **5**

# **Introduction**

The GTA Application Portal Architecture Document is written to explain the architecture and design techniques being utilized in the GTA Application Portal. The document contains an overall view of this architecture style, the classification of the architecture tiers, class diagrams, and the database schema.

# Architecture Style

## Architecture Diagram

## Architecture Description

The architecture style for the GTA Application Portal is a 3-tiered architecture. It follows the simple structure of having a tier for user-interface, business logic, and data. Information can flow between these tiers but never skip over the middle tier. This provides proper information hiding and modularity throughout the system. All of the business logic will be handled using Python’s Flask framework. The database level will be made with SQLite. The user interface will be created using HTML, CSS, JavaScript, and Python within the Flask framework.

# Tier Classification

## User Interface Tier

**Purpose:** To display information, including input forms, navigation controls, and images, to the user in an understandable way.

**Specific Nature:** This tier will be responsible for displaying information in an easy to understand and usable way. It will also be responsible for page navigation across the web page. When a user clicks one of the tabs in the navigation bar or specific buttons on the web page, the code corresponding to those elements will be called. This tier is also responsible for getting the user input through the use of forms and passing it to the logic tier.

**Important Subcomponent(s):** Navigation Bar, Pages

**Navigation Bar -** The navigation bar is a UI element that will be present in all pages of the website at the top of the web page. It will contain different navigation buttons depending on the type of user logged in. Clicking on these buttons will send the user to the corresponding page.

**Pages -** The different interfaces to be displayed to the user. These pages are formatted and present the user with input form fields or buttons to be interacted with as well as information relevant to the type of page.

## Logic Tier

**Purpose:** This tier is responsible for handling data and processes between the user interface and database tiers, also contains the classes of the program, and algorithms.

**Specific Nature:** This tier functions as the backend of the program that contains the classes, algorithms, and functions necessary for the program to run. It is also the tier that will be used to perform queries and transactions with the database. The routing is also created in this tier, which is what is used to display the different pages of the websites. Whenever a new page is to be opened, the program retrieves the code contained in the given route and runs it before displaying the page needed.

**Important Subcomponent(s):** Routes, Classes, Database Transactions

**Routes -** Routes function similarly to generic program loops. These are sections of the code that are retrieved and run when the user navigates to a new page. These sections contain algorithms to determine what needs to be displayed to the user based on the system database, the current user type, and the current users information in the database. Error handling for the different web pages also occurs here.

**Classes -** There are two main types of classes being used. Classes that resemble tables in the database and classes for input forms.

**Database Transactions -** Methods to add and retrieve data within the database are also held in this tier. These transactions can include adding a new user, creating a new job listing, etc.

## Database Tier

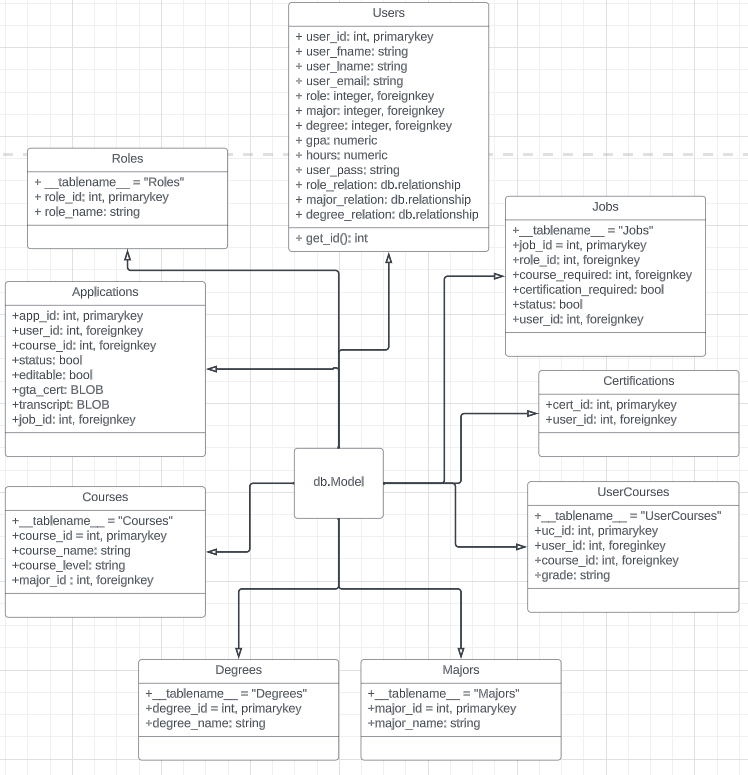
**Purpose:** This tier is in charge of storing all relevant data of the program.

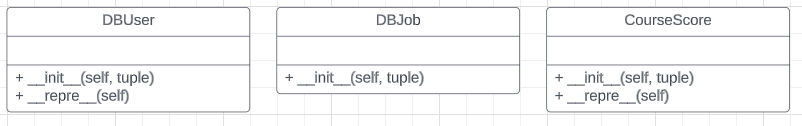
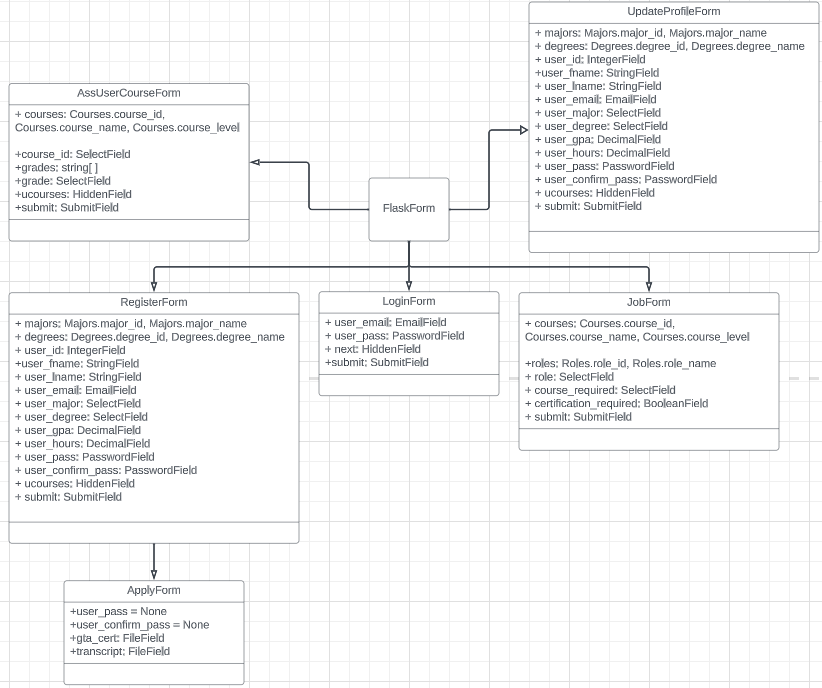
**Specific Nature:** This tier consists of a single database file composed of multiple tables storing information about users, job listings, and applications. User information will be stored with references to their corresponding applications submitted and classes taken.

**Important Subcomponent(s):** Tables

**Tables -** A database table with multiple fields relating to a specific element of the program. For example the user table will store the user’s first name, last name, email, password, major, degree, gpa, and all other relevant information about the user.

# Class Diagrams





# Database Schema

