جامعة القاهرة
كلية الدراسات العليا للبحوث الإحصائية

CAIRO UNIVERSITY

SOFTWARE ENGINEERING DIPLOMA

2024

A Proposed website for Faculty of Graduate Studies for Statistical Research

Supervised by: Dr Atef Raslan

Supervised by: Dr AbdelMoniem Helmy

Student number	Team Member
202301420	احمد معتز عبد الفتاح عبد اللطيف سليم
202301608	نجلاء أحمد محمود أحمد
202301563	سامح طه شرف هاشم

ACKNOWLEDGMENTS

At the beginning and at the end we all thank Allah for helping us to achieve this work and ask Allah to benefit us with what he taught us and teach us what will benefit us.

Also, the achievement of this piece of work wouldn’t be to see the light without the support, help and contribution of Doctor: Abdelmoniem Helmy supervisor of our graduation project and all the faculty members who taught us during our study in the degree program.

Our greetings for the software engineering department management from the professors, staff and employees' Warm greetings for doctors and teaching assistants

Finally, we thank our families for supporting us and helped us to reach so far and to have the tolerance to face difficulties and keep up.

ABSTRACT

Designing an Engaging University Website In the digital era,

a well-designed university website plays a crucial role in connecting with students, faculty, and the broader community. Our project aims to create an informative and user-friendly platform that reflects the institution's values and enhances the online experience for all stakeholders.

Objectives:

1. User-Centric Design: Our primary focus is on user experience. We design intuitive navigation, clear information architecture, and accessible content.

2. Showcasing Diplomas: The website will highlight academic programs, professional programs, faculty expertise, faculty News, and student achievements.

3. Admissions and Enrollment: Prospective students should find admission requirements, application procedures, and enrollment details easily.

4. Campus Life and Events: We feature campus events, student organizations, and news updates to foster a sense of community.

5. Responsive and Accessible: The website will adapt seamlessly to various devices (desktop, tablet, mobile) and comply with accessibility standards.

Key Features: 1. Homepage: A dynamic homepage with concise information about the university's mission, vision, and core values.

2. Diplomas Departments: Dedicated pages for each department, including faculty profiles, course offerings, and research areas.

3. Admissions Portal: Clear instructions for prospective students, application deadlines.

4. Events Calendar: An interactive calendar showcasing lectures, workshops, and cultural events.

5. Contact Information: Easily accessible contact details.

DEDICATION

We dedicate this documentation to the esteemed Dean, our dedicated staff, and the entire community of the Faculty of Graduate Studies for Statistical Research. Their tireless efforts, commitment to excellence, and passion for academic and research pursuits are the cornerstone of our institution's success.

To the Dean, DR. Sayed khater:

We extend our heartfelt gratitude to the Dean of the Faculty of Graduate Studies for Statistical Research for their visionary leadership and unwavering dedication to fostering a culture of academic excellence and innovation. Under their guidance, our institution continues to thrive and uphold its reputation as a center of excellence in statistical research and education.

To the Staff:

We salute the exemplary efforts of our esteemed staff professors who serve as pillars of knowledge and inspiration for our students. Their expertise, dedication, and mentorship empower our students to excel in their academic and research endeavors, shaping the future of statistical research and scholarship.

To the College Community:

Lastly, we extend our appreciation to the entire college community for their collective contributions to the advancement of statistical research and education. Together, we strive to uphold the highest standards of academic integrity, foster a culture of collaboration and innovation, and make meaningful contributions to the field of statistics.

This documentation stands as a testament to the dedication, hard work, and collective achievements of our dean, staff professors, and the entire college community. We are proud to be part of the Faculty of Graduate Studies for Statistical Research and remain committed to advancing the frontiers of knowledge and research in statistics.

TABLE OF CONTENTS

CHAPTER 1

INTRODUCTION

1. Introduction

The Faculty of Graduate Studies Statistical Website aims to provide a comprehensive platform for faculty members, students, and researchers to access and analyze statistical data related to graduate studies. This website will serve as a centralized hub for storing, managing, and visualizing various types of data, including enrollment statistics, graduation rates, research output, and faculty demographics.

The primary objective of this SRS document is to outline the requirements, functionalities, and constraints of the Faculty of Graduate Studies Statistical Website. It will define the scope of the project, identify the target audience, and specify the features and functionalities that the website should support.

This document is intended for stakeholders, including faculty members, IT professionals, and project managers involved in the development, implementation, and maintenance of the Faculty of Graduate Studies Statistical Website. It will serve as a reference guide throughout the development process, ensuring that the final product meets the needs and expectations of its users.

Adopting Agile XP Methodology

In the development of our college website, our team members embraced the Agile XP (Extreme Programming) methodology to ensure efficient collaboration, rapid iteration, and high-quality results. By following the Agile principles of communication, simplicity, feedback, and courage, we aimed to deliver a website that meets the diverse needs of our college community while maintaining flexibility to adapt to evolving requirements. Through continuous communication, frequent releases, and a focus on teamwork, we strived to maximize value and deliver a website that reflects our commitment to excellence and innovation in higher education.

1.1 Purpose

The purpose of the Faculty of Graduate Studies Statistical Website is to:

- Provide easy access to accurate and up-to-date statistical data related to graduate studies.
- Enable users to analyze and visualize data through interactive charts, graphs, and tables.
- Facilitate data-driven decision-making processes for faculty members, administrators, and researchers.
- Enhance transparency and accountability by making key performance indicators and metrics publicly available.

1.2 Scope

The scope of the Faculty of Graduate Studies Statistical Website includes:

- Data collection, storage, and management of various statistical data related to graduate studies.
- Development of user-friendly interfaces for data input, retrieval, and analysis.
- Implementation of interactive data visualization tools, such as charts, graphs, and dashboards.
- Integration with existing university systems and databases, where applicable.
- Security measures to protect sensitive and confidential data.

1.3 Target Audience

The primary target audience for the Faculty of Graduate Studies Statistical Website includes:

- Faculty members and academic staff involved in graduate studies.
- Graduate students seeking information on enrollment, program statistics, and research opportunities.
- Researchers and administrators interested in analyzing and interpreting graduate studies data.
- Prospective students and external stakeholders, such as funding agencies and accreditation bodies.

1.4 Definitions, Acronyms, and Abbreviations

- **SRS**: Software Requirements Specification
- **IT**: Information Technology
- **KPIs**: Key Performance Indicators
- **UI**: User Interface
- **UX**: User Experience

Objectives:

1. Define Clear and Measurable Requirements:

- Clearly articulate the functional and non-functional requirements of the website to ensure a common understanding among stakeholders.
- Establish measurable criteria for evaluating the performance, usability, and reliability of the website.

2. Identify Stakeholders and Users:

- Clearly identify and define the roles, responsibilities, and needs of all stakeholders involved in the development, implementation, and maintenance of the website.
- Understand the requirements and expectations of the website's primary users, including faculty members, students, researchers, and administrators.

3. Establish Scope and Boundaries:

- Clearly define the scope of the website, including the types of data to be collected, stored, and managed, as well as the functionalities and features to be implemented.
- Establish boundaries to ensure that the website's capabilities and limitations are clearly understood and communicated to stakeholders.

4. Ensure Data Accuracy and Integrity:

- Implement robust data validation and verification mechanisms to ensure the accuracy, consistency, and integrity of the statistical data presented on the website.
- Define data governance policies and procedures to govern the collection, storage, and management of data in compliance with relevant regulations and standards.

5. Enhance User Experience and Accessibility:

- Design intuitive and user-friendly interfaces that facilitate easy navigation, data retrieval, and analysis for users with varying levels of technical expertise.
- Ensure that the website is accessible to users with disabilities by adhering to web accessibility standards and best practices.

6. Implement Security and Compliance Measures:

- Implement stringent security measures to protect sensitive and confidential data from unauthorized access, disclosure, and misuse.
- Ensure compliance with relevant data protection, privacy, and security regulations

7. Facilitate Data Analysis and Visualization:

- Implement interactive data visualization tools, such as charts, graphs, and dashboards, to enable users to analyze and interpret statistical data effectively.
- Provide advanced analytics capabilities, such as data filtering, sorting, and drill-down, to facilitate in-depth data analysis and exploration.

8. Support Integration and Interoperability:

- Ensure seamless integration with existing university systems, databases, and third-party applications to facilitate data sharing and interoperability.
- Define clear interface requirements and specifications for data exchange, API integration, and system interoperability.

9. Establish Performance and Scalability Requirements:

- Define performance metrics and benchmarks to evaluate the website's responsiveness, throughput, and scalability under varying load conditions.
- Identify potential performance bottlenecks and scalability constraints and propose mitigation strategies and optimization techniques.

10. Ensure Maintainability and Extensibility:

- Adopt modular and scalable design principles to facilitate future enhancements, updates, and modifications to the website.
- Define coding standards, documentation requirements, and version control practices to ensure maintainability, extensibility, and code quality.

System Requirements:

Development Environment

1. **Operating System**:
   - Windows 10 or later
   - macOS 10.14 or later
   - Linux (various distributions like Ubuntu, Debian, Fedora, etc.)
2. **Hardware Requirements**:
   - Processor: 1.4 GHz or faster
   - RAM: 4 GB minimum (8 GB or more recommended)
   - Disk Space: At least 1 GB of free space for the .NET SDK and tools
3. **Software Requirements**:
   - .NET SDK: .NET Core 3.1 or .NET 5.0/6.0/7.0/8.0 (depending on your project's target framework)
   - IDE:
     - Visual Studio 2019 or later (Windows)
     - Visual Studio for Mac (macOS)
   - Microsoft SQL Server 2016 or later
4. **Additional Tools**:
   - Database Management System (SQL Server, PostgreSQL, MySQL, etc.)

Deployment Environment

1. **Hardware Requirements for Servers**:
   - Varies based on the expected load. A typical web server might start with:
     - CPU: Multi-core processor
     - RAM: 4 GB or more
     - Storage: SSD preferred for better performance
2. **Software Requirements for Servers**:
   - .NET Runtime: Install the appropriate .NET Core runtime version (matching your application)
   - Web Server Configuration:
     - For IIS: ASP.NET Core Module (ANCM) installed
     - For Linux: Proper configuration of reverse proxy (Nginx/Apache) to forward requests to Kestrel
   - Database Server: SQL Server 2016 or later.

Security and Maintenance

1. **Security**:
   - SSL/TLS for secure communication (obtain and install SSL certificates)
   - Regular updates and patches for the OS, web server, and application dependencies
   - Proper firewall and network security configurations
2. **Backup and Recovery**:
   - Regular backups of databases and application data
   - Disaster recovery plans
3. **Monitoring and Logging**:
   - Application performance monitoring tools (e.g., Application Insights, New Relic)
   - Logging frameworks (e.g., Serilog, NLog)
   - Health checks and alerts for proactive maintenance
4. **Scalability**:
   - Load balancing for handling increased traffic
   - Auto-scaling configurations on cloud platforms

By meeting these requirements, you can ensure that your .NET Core web application is developed, deployed, and maintained effectively.

Implementation Tools and Technology:

1. Backend Technology:

.NET Core (C#):

Description: .NET Core is a cross-platform, high-performance framework for building modern, cloud-based, and internet-connected applications. C# is the primary programming language used in .NET Core.

Usage: The backend of the university website is built using .NET Core, providing a robust and scalable framework for handling business logic, data access, and server-side operations.

Benefits: High performance, cross-platform capabilities, extensive library support, strong community and enterprise backing

2. Frontend Technology:

HTML:

Description: HTML (Hypertext Markup Language) is the standard language for creating and structuring web pages.

Usage: Used for building the structure of the web pages, ensuring content is presented in a clear and accessible format.

Benefits: Universally supported by web browsers, easy to learn and use, and forms the backbone of web development.

Bootstrap CSS:

Description: Bootstrap is a popular front-end framework for developing responsive and mobile-first websites using CSS and JavaScript.

Usage: Bootstrap is used for styling the web pages, providing a consistent and modern look and feel across the site. It ensures responsiveness, making the website accessible on various devices and screen sizes.

Benefits: Pre-built components, grid system for layout design, responsive design capabilities, and extensive customization options.

jQuery:

Description: jQuery is a fast, small, and feature-rich JavaScript library that simplifies HTML document traversal, event handling, and animation.

Usage: jQuery is used to enhance interactivity and dynamic content on the website. It simplifies AJAX calls for asynchronous data fetching and provides a wide range of utility functions to manipulate the DOM.

Benefits: Simplifies JavaScript code, improves browser compatibility, and enhances user experience with animations and event handling.

3. Real-Time Communication:

SignalR:

Description: SignalR is a library for ASP.NET that simplifies the process of adding real-time web functionality to applications. It allows server-side code to push content to connected clients instantly.

Usage: SignalR is used for real-time features such as live notifications, chat functionalities, and dynamic updates without requiring page reloads.

Benefits: Real-time communication capabilities, easy to implement and integrate with .NET Core, supports various transport methods (WebSocket's, Server-Sent Events, Long Polling).

4. Development Tools:

Visual Studio:

Description: Visual Studio is an integrated development environment (IDE) from Microsoft used for developing computer programs, websites, web apps, web services, and mobile apps.

Usage: Visual Studio is the primary IDE used for writing, debugging, and testing the code for the university website.

Benefits: Comprehensive suite of development tools, integrated debugging and testing features, strong support for .NET Core, and a wide range of extensions and integrations.

CHAPTER 2

REQUIREMENTS SPECIFICATIONS

1. Functional Requirements:

Identifier	priority	Requirement
REQ1	1	System should enable the student to login to the website
REQ2	3	System should enable the student to register to the website
REQ3	5	System should enable the student to view all available programs
REQ4	1	System should enable the student to view all diplomas
REQ5	3	System should enable the student to search for any diploma
REQ6	3	System should enable the student to apply for specific diploma
REQ7	4	System should enable the student to have a profile page and be able to edit this page
REQ8	2	System should enable the student to cancel his admission anytime

REQ9	2	System should enable the Administrator to login as admin
REQ10	4	System should enable the Administrator to register as admin
REQ11	4	System should enable the Administrator to view all users in the website
REQ12	5	System should enable the Administrator to edit user roles
REQ13	4	System should enable the Administrator to view all admissions
REQ14	4	System should enable the Administrator to edit user admission status
REQ15	3	System should enable the Administrator to edit events
REQ16	3	System should enable the Administrator to add new events
REQ17	3	System should enable the Administrator to delete an event
REQ18	4	System should enable the Administrator to edit diplomas
REQ19	4	System should enable the Administrator to add new diplomas

REQ20	3	System should enable the Administrator to delete diplomas
REQ21	4	System should enable the Administrator to add new department
REQ22	4	System should enable the Administrator to edit department
REQ23	3	System should enable the Administrator to delete a department
REQ24	4	System should enable the Administrator to add new staff member
REQ25	3	System should enable the Administrator to edit staff member info
REQ26	3	System should enable the Administrator to delete staff member
REQ27	3	System should enable the Administrator to add new news
REQ28	3	System should enable the Administrator to edit news
REQ29	3	System should enable the Administrator to delete news

2. Non-Functional Requirements:

1- Scalable

System should be scalable for future integration and updates.

2- security

System should be secure and allow only each account to see his authorized dashboard

No one can access any account without valid and active username and password.

3- Availability: The website must be available 99.9% of the time, 24/7, excluding scheduled maintenance.

4- Backup and Recovery: Daily backups should be performed

5- Error Handling: The website should gracefully handle errors and provide meaningful error messages to users without exposing system details.

6- Code Quality: The codebase should adhere to coding standards and best practices, including proper documentation and comments.

7- Compatibility: The website should be compatible with the latest versions of major browsers (Chrome, Firefox, Safari, Edge) and degrade gracefully on older versions.

SYSTEM ARCHITECTURE

MVC is not a programming language, it is simply a design pattern which is used to develop applications. MVC design pattern was very popular for development of Graphical User Interface (GUIs) applications, but now it is also popular in the designing of web applications, mobile and desktop etc. Many programming languages use this pattern

The MVC pattern helps you create applications that separate the different aspects of the application (input logic, business logic, and UI logic), while providing a loose coupling between these elements. The pattern specifies where each kind of logic should be located in the application. The UI logic belongs in the view. Input logic belongs in the controller. Business logic belongs in the model. This separation helps you manage complexity when you build an application, because it enables you to focus on one aspect of the implementation at a time. For example, you can focus on the view without depending on the business logic.

The Model-View-Controller (MVC) architectural pattern separates an application into three main components: the model, the view, and the controller.

Models. Model objects are the parts of the application that implement the logic for the application s data domain. Often, model objects retrieve and store model state in a database. For example, a Product object might retrieve information from a database, operate on it, and then write updated information back to a Products table in SQL Server.

Views. Views are the components that display the application s user interface (UI). Typically, this UI is created from the model data. An example would be an edit view of a Products table that displays text boxes, drop-down lists, and check boxes based on the current state of a Products object.

Controllers. Controllers are the components that handle user interaction, work with the model, and ultimately select a view to render that displays UI. In an MVC application, the view only displays information; the controller handles and responds to user input and interaction. For example, the controller handles query-string values, and passes these values to the model, which in turn queries the database by using the values.

SYSTEM MODULES

User Authentication and Authorization: Managing user login, registration, roles (e.g., student, staff, admin)

Student Management: Managing student information, admission, and other related data.

Communication Module: Facilitating interactions such as messaging between students and staff, notifications, and email integration.

Diplomas Management: Creating, updating, and deleting diplomas, as well as enrolling students in them.

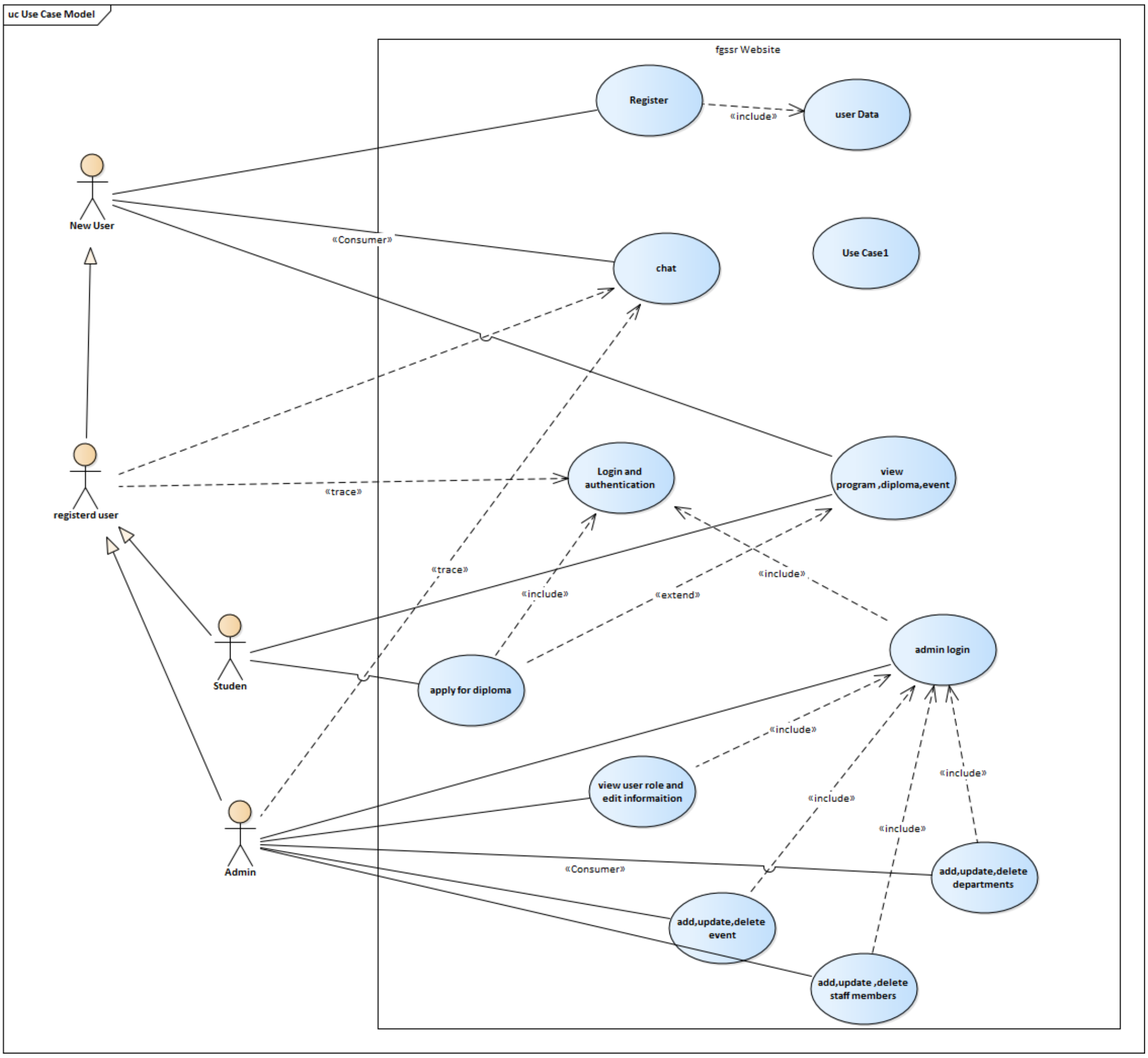
Events and News: Handles the creation, update, and display of events and news items relevant to the university community.

Admin Dashboard: Provides administrative capabilities to manage users, courses, events, and news.

Diplomas admission: students can apply for specific diplomas and view its details

UML DESIGN

USE CASE DIAGRAM



USE CASE

01.UC01 (USER REGISTRATION)

Entry Point:

- **Event:** User arrives at the registration page.
- **Description:** The entry point occurs when a user navigates to the registration page of the platform with the intention of creating a new account.

Exit Point:

- **Event:** User is redirected to the login page with a success message.
- **Description:** The exit point happens when the user has successfully completed the registration process. They are redirected to the login page of the platform with a success message indicating that their account has been created. From this point, the user can proceed to log in with their newly registered credentials.

event flow

This entry and exit points define the start and end of the "Register" use case, capturing the user's journey from arriving at the registration page to successfully creating an account and being directed to the login page.

User Registration

Objective: To allow a user to create an account on the Fgssr website.

Preconditions:

- The user is on the registration page of the platform.

Postconditions:

- The user has a registered account on the platform.

Main Flow:

1. User arrives at the registration page.
2. The system displays a registration form with the following fields:
    - Full Name
    - Email Address
    - Password
    - Confirm Password
    - Date of Birth
    - Gender
    - Address
    - Phone Number
    - Security Questions (if required)
3. User fills in the required information.
4. The system validates the entered information:
    - Checks if the email address is not already registered.
    - Verifies that the password meets security requirements.
    - Ensures that the email address is in a valid format.
    - Verifies that the Date of Birth is within an acceptable range.
5. If there are validation errors, the system displays error messages and asks the user to correct them.
6. If there are no errors, the system proceeds to the next step.
7. User submits the registration form.
8. The system creates a new user account with the provided information.
9. The system sends a verification email to the user's email address to confirm the registration.
10. User receives the verification email and clicks on the verification link.
11. The system verifies the link and activates the user's account.
12. User is redirected to the login page with a success message.
13. User can now log in with the registered email and password.

02. UC02 (USER LOGIN)

Entry Point:

- **Event:** User arrives at the login page.
- **Description:** The entry point occurs when a user navigates to the login page of the platform with the intention of logging into their existing account.

Exit Point:

- **Event:** User successfully logs in and is redirected to the platform's dashboard.
- **Description:** The exit point happens when the user has successfully entered their correct credentials (email and password) and is authenticated by the system. They are then redirected to the platform's dashboard or main interface.

Event Flow:

1. User arrives at the login page.
2. The system displays a login form with the following fields:
   - Email Address
   - Password
3. User enters their registered email address and password.
4. The system validates the entered credentials:
   - Checks if the email address exists in the system.
   - Verifies that the entered password matches the stored password for the corresponding email address.
5. If there are validation errors (e.g., incorrect email or password), the system displays an error message and asks the user to correct them.
6. If there are no errors, the system proceeds to the next step.
7. User submits the login form.
8. The system authenticates the user by verifying the provided email and password.
9. If authentication is successful, the system logs the user into their account.
10. The user is redirected to the platform's dashboard or main interface.
11. The user can now access their account and use the platform's features.

03. UC03 (CHATTING WITH ADMIN ON FGSSR WEBSITE)

Entry Point:

- User navigates to the FGSSR website and clicks on the chat icon to initiate a conversation with the admin.

Exit Point:

- Chat session ends with the admin providing assistance and resolving the user's concerns or issues.
- User clicks on the "End Chat" button or closes the chat window to conclude the conversation.

Event Flow:

1. **User Initiates Chat**
   - User navigates to the FGSSR website.
   - User clicks on the chat icon to initiate a conversation with the admin.
2. **Admin Receives Notification**
   - Admin receives a notification on the admin dashboard indicating that a user wants to chat.
3. **Admin Joins Chat**
   - Admin accepts the chat request from the user.
   - Admin joins the chat conversation to assist the user.
4. **Chatting**
   - User and Admin exchange messages to discuss the user's questions, concerns, or issues related to the FGSSR website or services.
   - Admin provides assistance, answers queries, or addresses issues raised by the user during the chat session.
5. **Problem Resolution**
   - Admin resolves the user's concerns, provides solutions, or offers guidance based on the chat conversation.
   - Admin updates any necessary records or takes actions as required.
6. **Closing Chat**
   - Admin informs the user that the chat is concluding.
   - User acknowledges and ends the chat session by clicking on the "End Chat" button or closing the chat window.

ALTERNATIVE FLOWS:

1. **Admin Unavailable**
   - If the admin is not available to chat immediately, the user may leave a message.
   - Admin can respond later when available to address the user's concerns.
2. **Technical Issues**
   - If there are technical issues with the chat feature, the user and admin may switch to another communication method like email or phone to continue the conversation.

EXCEPTION FLOWS:

1. **User Abandons Chat**
   - If the user abruptly leaves the chat without resolving the issue, the admin may follow up with the user through other means to address the concerns.

2. **Miscommunication**

- If there is a misunderstanding or miscommunication during the chat, the admin and user should clarify and restate their points to ensure clear communication and problem resolution.
- .

04. UC04 (CHATTING WITH ADMIN ON FGSSR WEBSITE)

USE CASE: APPLYING FOR DIPLOMA ON FGSSR WEBSITE

ACTORS:

- Applicant (User)
- FGSSR Website System

ENTRY POINT:

- Applicant navigates to the FGSSR website and accesses the "Apply for Diploma" section or clicks on the "Apply Now" button/link.

EXIT POINT:

- Applicant completes the application process by submitting all required information and documents.
- Applicant receives a confirmation message or email indicating successful submission of the diploma application.

EVENT FLOW:

1. **Access Application Form**
   - Applicant navigates to the "Apply for Diploma" section on the FGSSR website.
   - Applicant clicks on the "Apply Now" button/link to access the online application form.
2. **Provide Personal Information**
   - Applicant fills in personal details such as full name, date of birth, contact information, etc., in the online application form.
3. **Select Diploma Program**
   - Applicant selects the desired diploma program from the available options in the application form.
4. **Upload Required Documents**
   - Applicant uploads necessary documents such as educational transcripts, identification proof, passport-size photo, etc., as specified in the application guidelines.
5. **Review and Confirm Details**
   - Applicant reviews all entered information and uploaded documents to ensure accuracy and completeness.
   - Applicant confirms that all details are correct and ready to submit.
6. **Submit Application**
   - Applicant submits the completed application form by clicking on the "Submit" button.
   - The FGSSR website system processes the submitted application and stores the applicant's information and documents securely.
7. **Receive Confirmation**
   - Applicant receives a confirmation message or email from the FGSSR website indicating successful submission of the diploma application.
   - The confirmation message/email may include details such as application ID, submission date, and next steps in the admission process.

ALTERNATIVE FLOWS:

1. **Incomplete Application**

- If the applicant fails to fill in all required fields or upload necessary documents, the system prompts the applicant to complete the missing information or upload the missing documents before proceeding with the submission.

2. **Application Fee Payment**
   - If there is an application fee associated with the diploma application, the system redirects the applicant to the payment gateway to complete the payment process before final submission.

EXCEPTION FLOWS:

1. **Technical Issues**
   - If the applicant encounters technical issues or errors while filling out the online application form, the system provides troubleshooting tips or prompts the applicant to try again later.
2. **Application Rejection**
   - If the submitted application does not meet the eligibility criteria or contains incorrect information, the FGSSR admissions team reviews the application and may reject it, notifying the applicant via email or phone with reasons for rejection and possible next steps.

05. UC05 (VIEWING AND EDITING USER ROLES ON FGSSR WEBSITE)

ACTORS:

- Admin (User with administrative privileges)
- FGSSR Website System

ENTRY POINT:

- Admin logs in to the FGSSR website admin dashboard and navigates to the user management section to view and edit user roles.

EXIT POINT:

- Admin completes the viewing and editing of user roles.
- Admin logs out of the admin dashboard or navigates to a different section of the website.

EVENT FLOW:

1. **Admin Login**
    - Admin accesses the FGSSR website admin dashboard by entering valid login credentials (username/password).
2. **Navigate to User Management**
    - Admin navigates to the user management section of the admin dashboard where user roles can be viewed and edited.
3. **View User Roles**
    - Admin selects a specific user or lists all users to view their current roles and permissions assigned on the FGSSR website.
    - The system displays a list of users along with their respective roles and permissions.
4. **Edit User Role**
    - Admin selects a user whose role needs to be edited.
    - Admin modifies the user's role by adding or removing specific permissions or changing the user's role type (e.g., from 'User' to 'Admin').
5. **Save Changes**
    - Admin saves the changes made to the user's role by clicking on the "Save" or "Update" button.
    - The FGSSR website system updates the user's role and permissions as per the admin's modifications.
6. **Confirmation**
    - The system displays a confirmation message indicating that the user's role and permissions have been successfully updated.
    - Admin reviews the confirmation message and ensures that the changes are accurately reflected.

ALTERNATIVE FLOWS:

1. **Add New User Role**
    - If the admin wants to create a new role, the admin navigates to the role management section, adds a new role with specific permissions, and assigns it to the desired user(s).
2. **Search and Filter Users**

- Admin uses search and filter options to quickly find specific users based on criteria such as username, role type, or permissions.

EXCEPTION FLOWS:

1. **Invalid Role Changes**
    - If the admin attempts to assign invalid or conflicting permissions to a user, the system displays an error message prompting the admin to correct the role settings.
2. **Unauthorized Access**
    - If a user without admin privileges tries to access the user management section to view or edit user roles, the system denies access and displays a message indicating unauthorized access.
3. **Technical Issues**
    - If the admin encounters technical issues or errors while viewing or editing user roles, the system provides troubleshooting tips or prompts the admin to try again later.

06.C06 (ADDING AND EDITING EVENTS ON FGSSR WEBSITE BY ADMIN)

ACTORS:

- Admin (User with administrative privileges)
- FGSSR Website System

ENTRY POINT:

- Admin logs in to the FGSSR website admin dashboard and navigates to the events management section to add or edit events.

EXIT POINT:

- Admin completes the addition or editing of events.
- Admin logs out of the admin dashboard or navigates to a different section of the website.

EVENT FLOW:

1. **Admin Login**
   - Admin accesses the FGSSR website admin dashboard by entering valid login credentials (username/password).
2. **Navigate to Events Management**
   - Admin navigates to the events management section of the admin dashboard where events can be added or edited.
3. **Add New Event**
   - Admin selects the option to add a new event.
   - Admin fills in the event details such as event name, date, time, location, description, and any other relevant information.
   - Admin uploads event images, flyers, or other multimedia files if required.
4. **Save New Event**
   - Admin saves the newly added event by clicking on the "Save" or "Publish" button.
   - The FGSSR website system processes the event details and adds the new event to the events calendar or listing on the website.
5. **Edit Existing Event**
   - Admin selects an existing event from the events list to edit.
   - Admin modifies the event details such as date, time, location, description, or updates event images and multimedia files as necessary.
6. **Save Edited Event**
   - Admin saves the changes made to the event by clicking on the "Update" or "Save Changes" button.
   - The FGSSR website system updates the event details and reflects the changes in the events calendar or listing on the website.
7. **Confirmation**
   - The system displays a confirmation message indicating that the event has been successfully added or edited.
   - Admin reviews the confirmation message and ensures that the event details are accurately updated and displayed on the website.

ALTERNATIVE FLOWS:

1. **Delete Event**
   - If the admin wants to remove an event, the admin selects the delete option next to the event in the events list, confirms the deletion, and the event is removed from the website.
2. **Search and Filter Events**
   - Admin uses search and filter options to quickly find specific events based on criteria such as event name, date, location, or category.

EXCEPTION FLOWS:

1. **Invalid Event Details**
   - If the admin enters invalid or incomplete event details, the system displays an error message prompting the admin to correct the event information before saving.
2. **Unauthorized Access**
   - If a user without admin privileges tries to access the events management section to add or edit events, the system denies access and displays a message indicating unauthorized access.
3. **Technical Issues**
   - If the admin encounters technical issues or errors while adding or editing events, the system provides troubleshooting tips or prompts the admin to try again later.

07. UC07 (USER CAN VIEW AND EDIT HIS INFO ON FGSSR WEBSITE)

ACTORS:

- User (Registered Member)
- FGSSR Website System

ENTRY POINT:

- User logs in to the FGSSR website using their username and password to access their account dashboard.

EXIT POINT:

- User completes viewing and editing of their personal information.
- User logs out of their account or navigates to a different section of the website.

EVENT FLOW:

1. **User Login**
    - User accesses the FGSSR website and logs in using their registered username and password.
2. **Navigate to Profile Settings**
    - Once logged in, user navigates to the profile or account settings section of the website to view and edit their personal information.
3. **View Personal Info**
    - User selects the option to view their profile or personal information.
    - The system displays the user's current information such as name, email address, contact number, and any other registered details.
4. **Edit Personal Info**
    - User clicks on the "Edit" or "Update" button next to the personal information section.
    - User modifies the desired fields such as name, email address, contact number, and updates any other necessary details.
5. **Save Changes**
    - User saves the updated information by clicking on the "Save" or "Update" button.
    - The FGSSR website system processes the changes and updates the user's personal information in the database.
6. **Confirmation**
    - The system displays a confirmation message indicating that the user's information has been successfully updated.
    - User reviews the confirmation message and verifies that the updated information is correctly reflected in their profile.

ALTERNATIVE FLOWS:

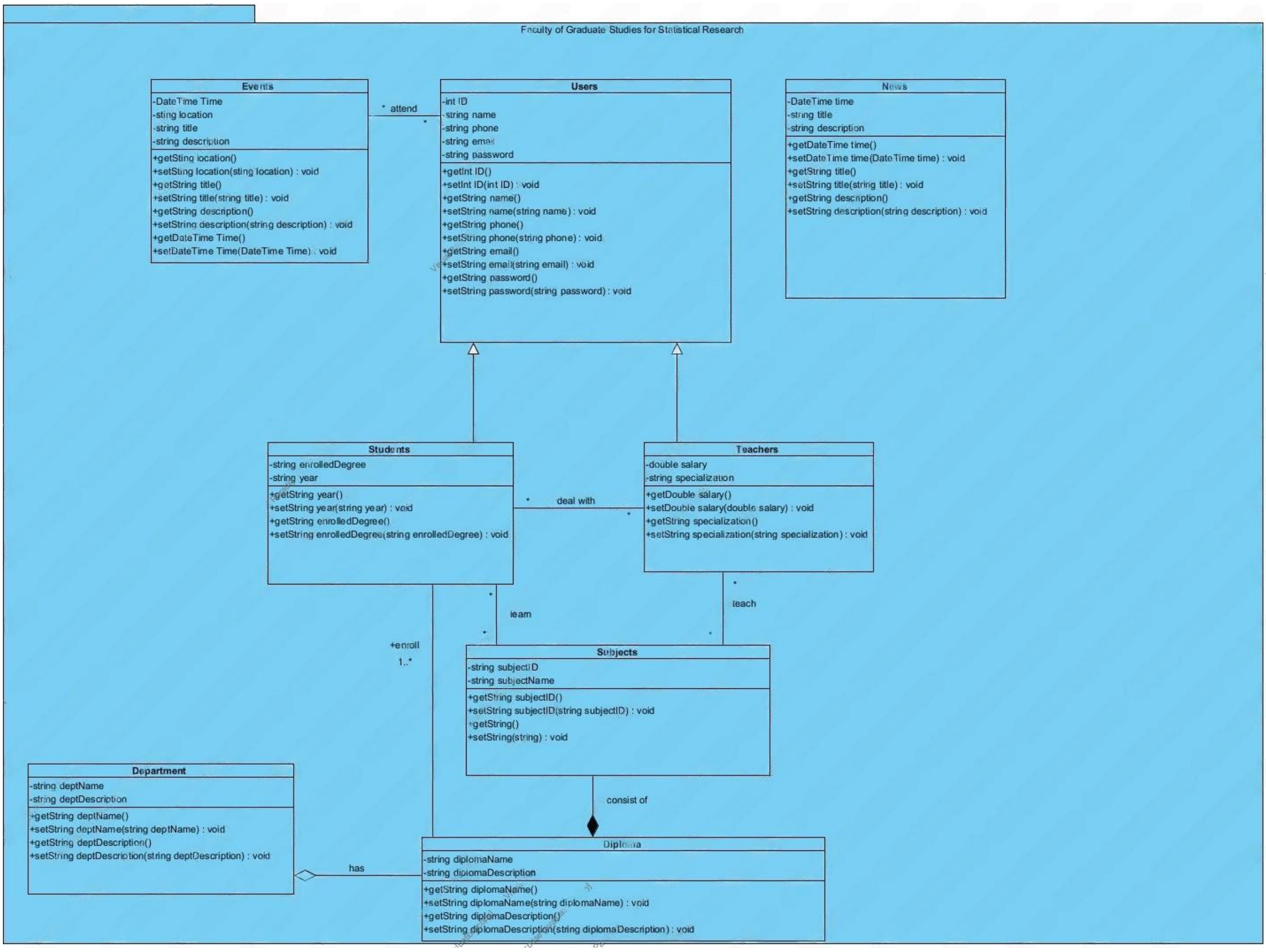
1. **Change Password**
    - User navigates to the password settings section to change their password.
    - User enters the current password, sets a new password, and confirms the new password to update their login credentials.
2. **Upload Profile Picture**

- User has the option to upload a profile picture or avatar to personalize their account.
- User selects an image file, uploads it, and the system updates the profile picture displayed on their account.

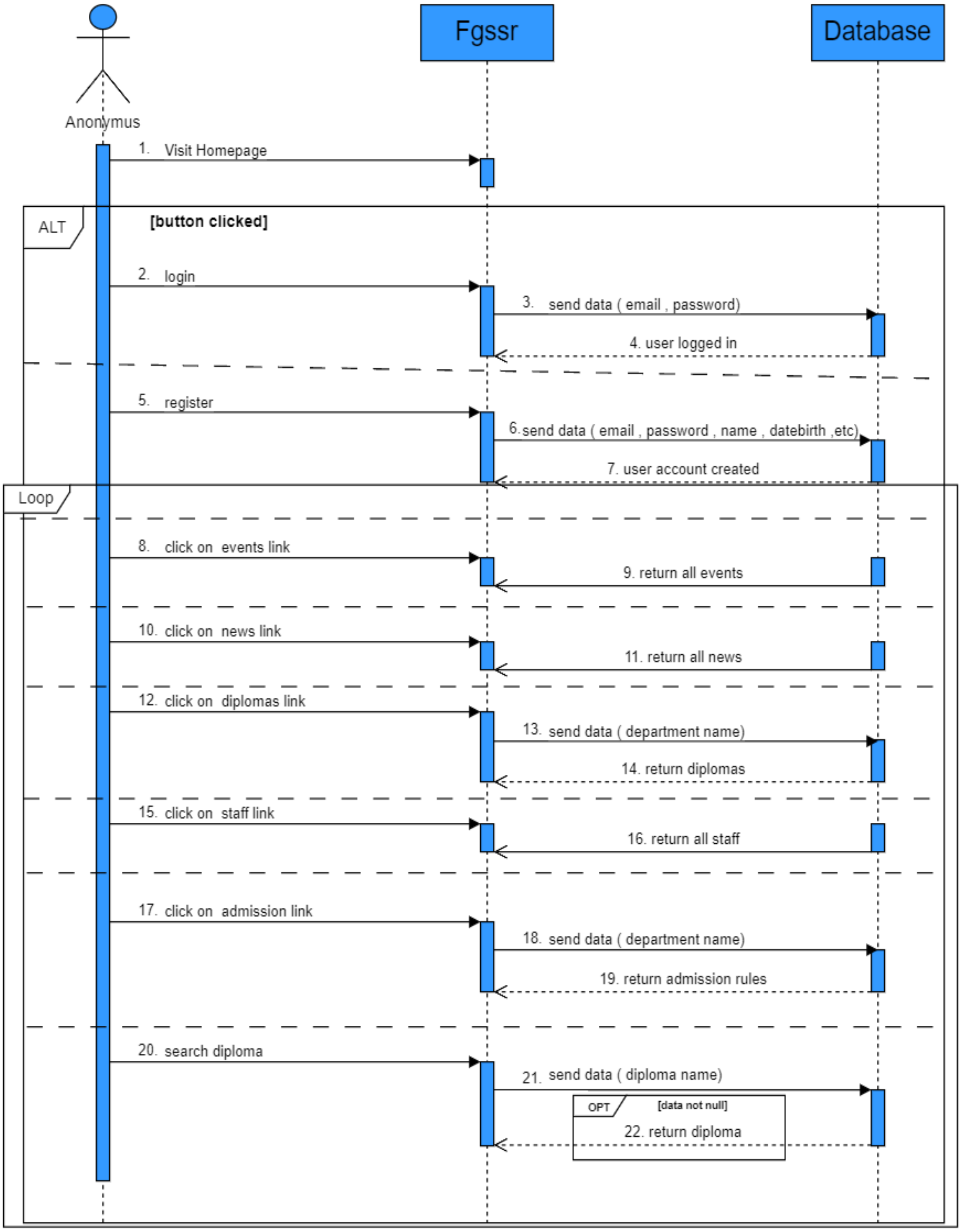
EXCEPTION FLOWS:

1. **Invalid Information**
    - If the user enters invalid or incorrect information while editing their profile, the system displays an error message prompting the user to correct the fields with invalid data.
2. **Unauthorized Access**
    - If a user tries to access or edit another user's personal information, the system denies access and displays a message indicating unauthorized access.
3. **Technical Issues**
    - If the user encounters technical issues or errors while editing their profile, the system provides troubleshooting tips or prompts the user to try again later.

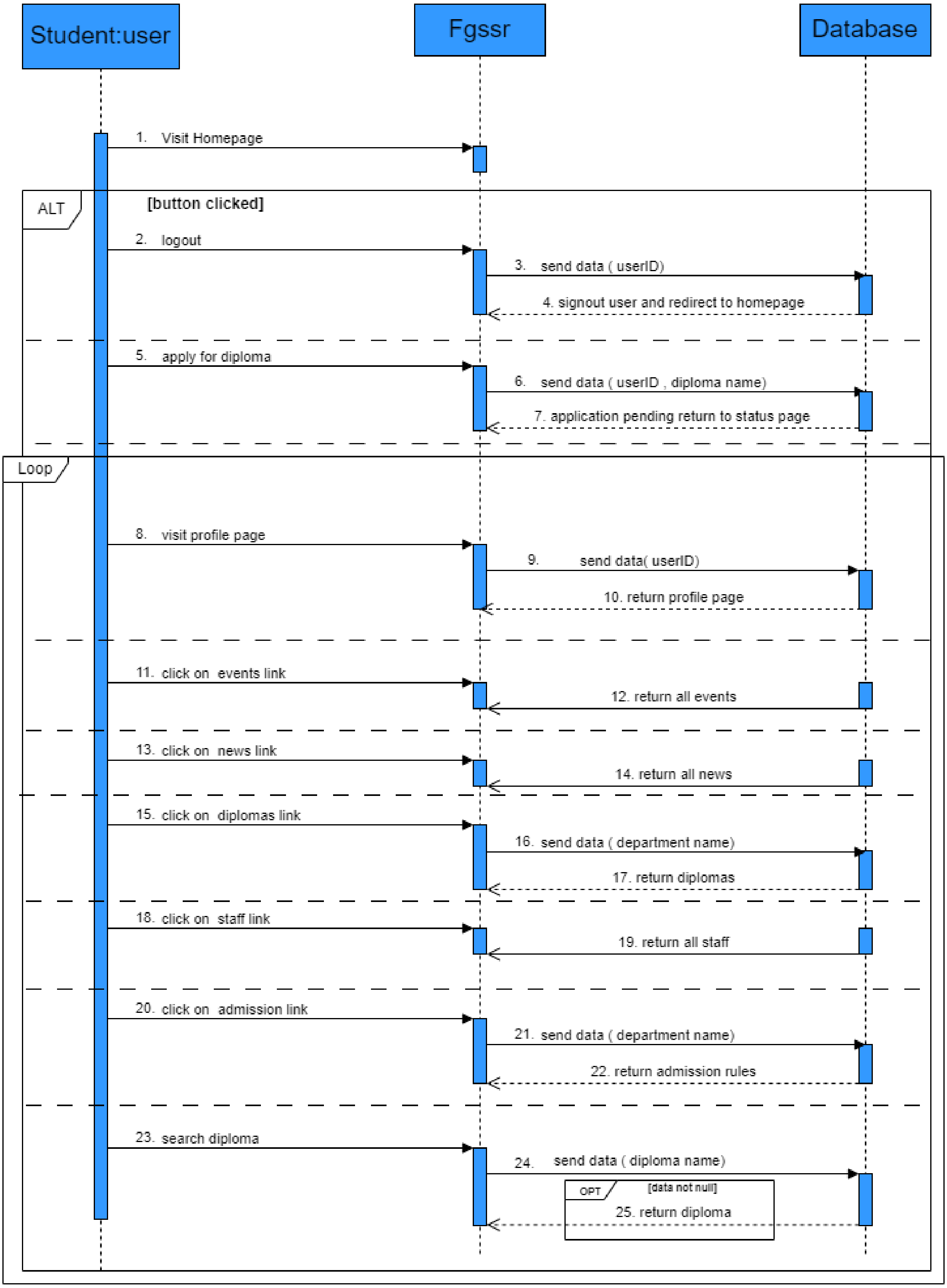
CLASS DIAGRAM


Faculty of Graduate Studies for Statistical Research
Events
-DateTime Time
-sting location
-string title
-string description
+getSting location()
+setSting location(sting location) : void
+getString title()
+setString title(string title) : void
+getString description()
+setString description(string description) : void
+getDateTime Time()
+setDateTime Time(DateTime Time) : void
* attend *
Users
-int ID
-string name
-string phone
-string email
-string password
+getInt ID()
+setInt ID(int ID) : void
+getString name()
+setString name(string name) : void
+getString phone()
+setString phone(string phone) : void
+getString email()
+setString email(string email) : void
+getString password()
+setString password(string password) : void
News
-DateTime time
-string title
-string description
+getDateTime time()
+setDateTime time(DateTime time) : void
+getString title()
+setString title(string title) : void
+getString description()
+setString description(string description) : void
Students
-string enrolledDegree
-string year
+getString year()
+setString year(string year) : void
+getString enrolledDegree()
+setString enrolledDegree(string enrolledDegree) : void
* deal with *
Teachers
-double salary
-string specialization
+getDouble salary()
+setDouble salary(double salary) : void
+getString specialization()
+setString specialization(string specialization) : void
* learn *
* teach *
+enroll
1..*
Subjects
-string subjectID
-string subjectName
+getString subjectID()
+setString subjectID(string subjectID) : void
+getString()
+setString(string) : void
consist of
Department
-string deptName
-string deptDescription
+getString deptName()
+setString deptName(string deptName) : void
+getString deptDescription()
+setString deptDescription(string deptDescription) : void
has
Diploma
-string diplomaName
-string diplomaDescription
+getString diplomaName()
+setString diplomaName(string diplomaName) : void
+getString diplomaDescription()
+setString diplomaDescription(string diplomaDescription) : void

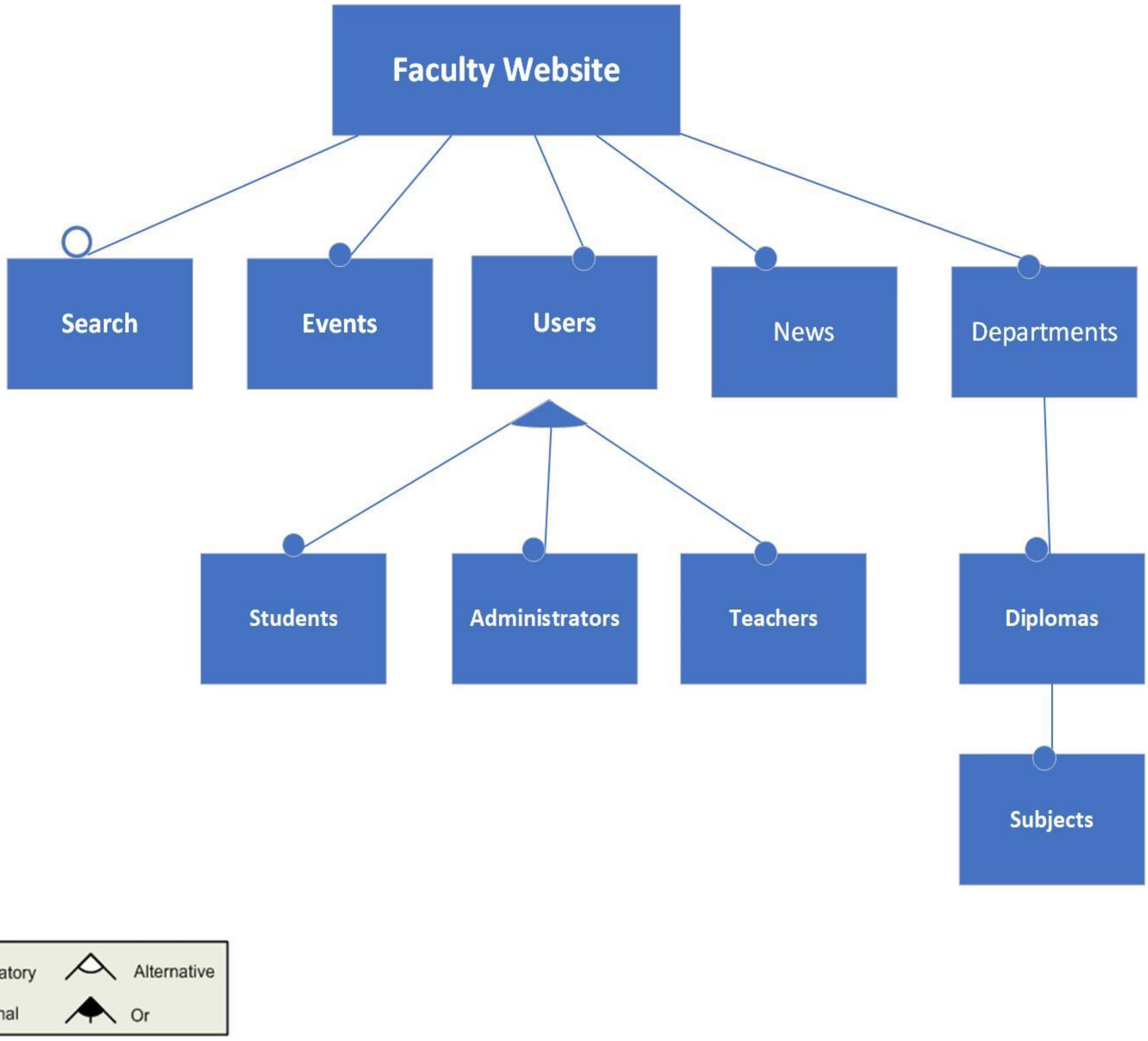
SEQUENCE DIAGRAM FOR ANONYMUS USER



SEQUENCE DIAGRAM FOR USER OBJECT



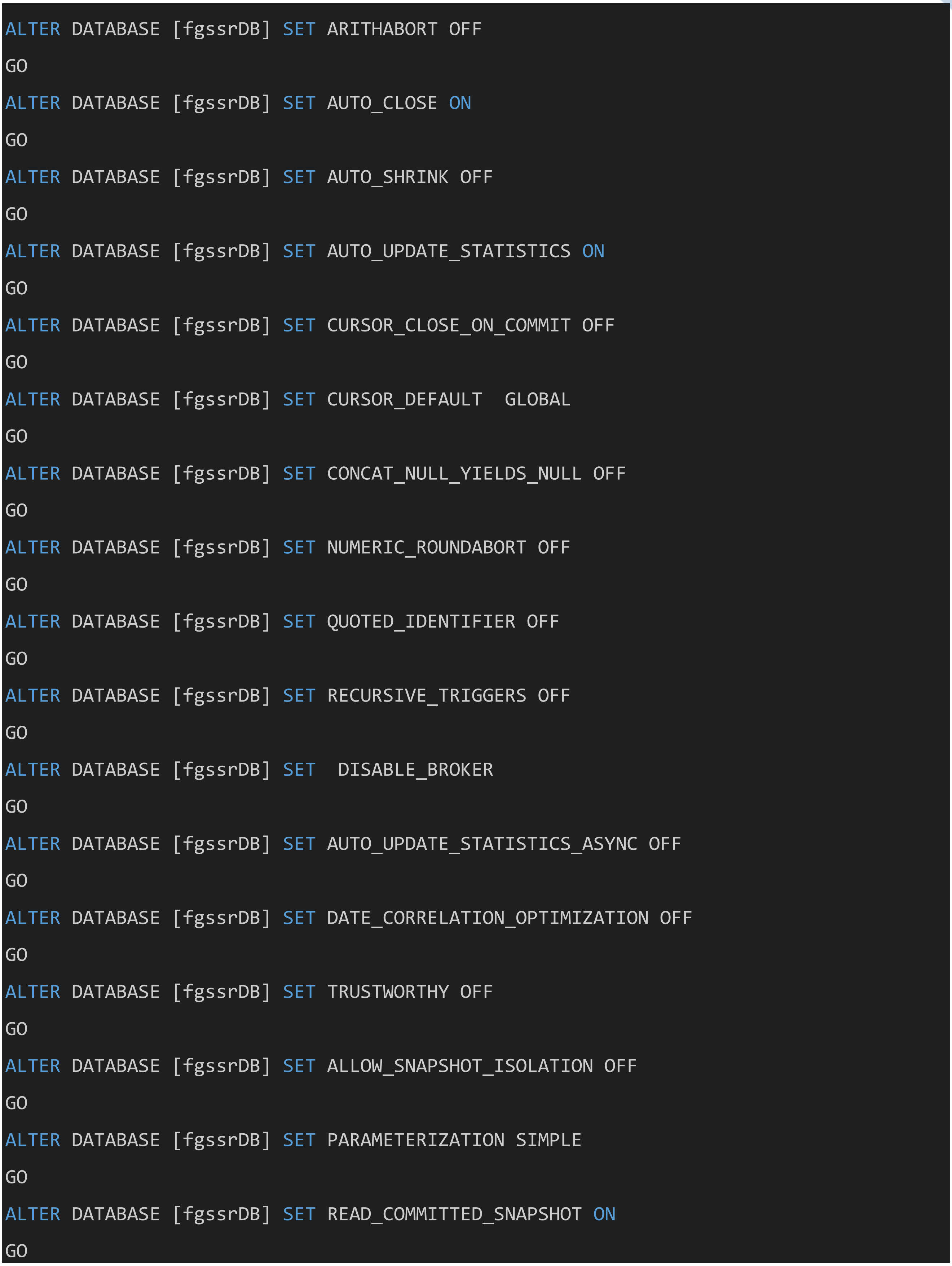
FEATURE DIAGRAM



DATABASE DESIGN

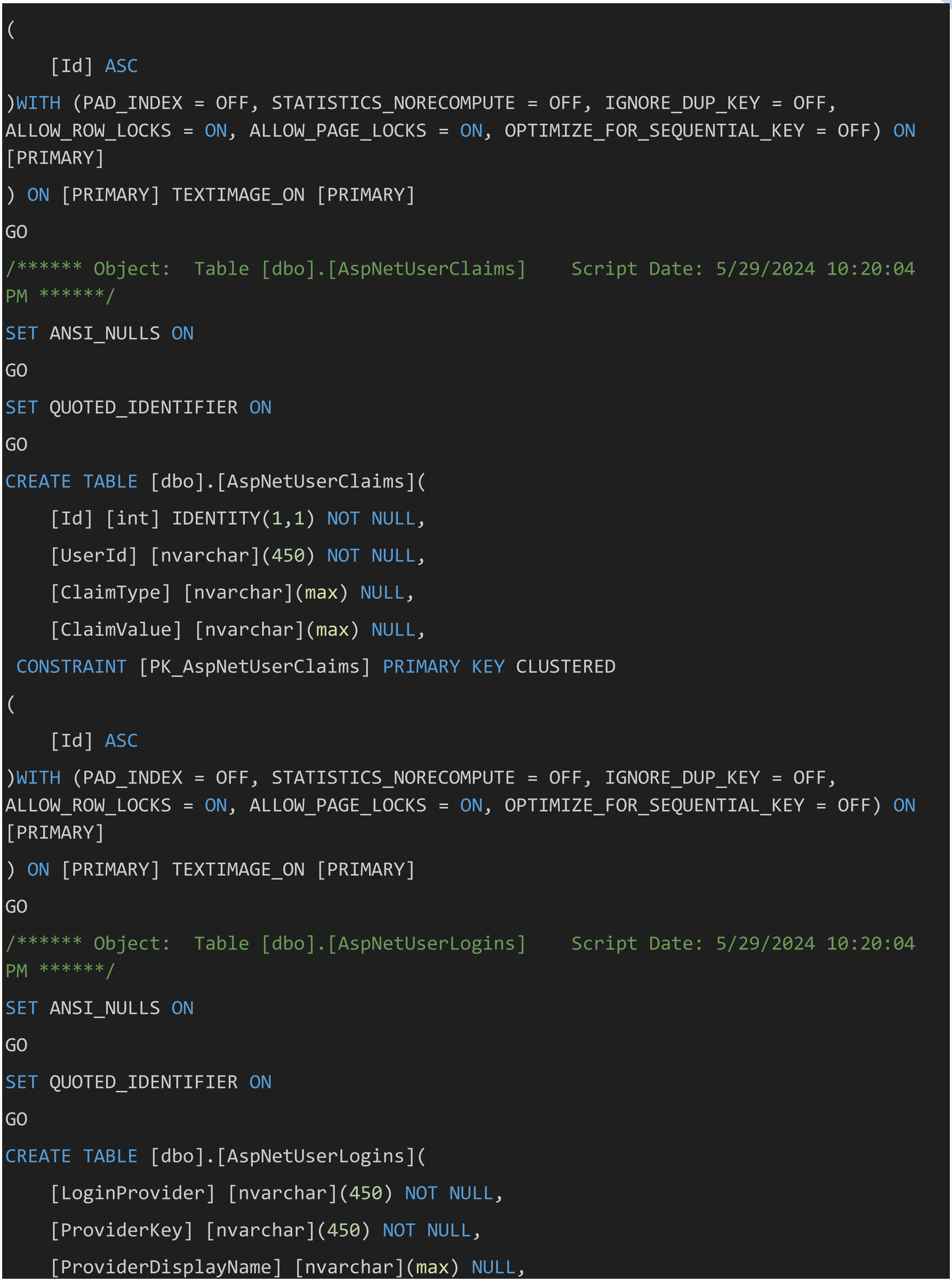
1. Data Dictionary:

```
USE [master]
GO
/****** Object:  Database [fgssrDB]    Script Date: 5/29/2024 10:20:04 PM ******/
CREATE DATABASE [fgssrDB]
 CONTAINMENT = NONE
 ON  PRIMARY
( NAME = N'fgssrDB', FILENAME = N'C:\Program Files\Microsoft SQL Server\MSSQL16.SQLEXPRESS\MSSQL\DATA\fgssrDB.mdf' , SIZE = 8192KB , MAXSIZE = UNLIMITED, FILEGROWTH = 65536KB )
 LOG ON
( NAME = N'fgssrDB_log', FILENAME = N'C:\Program Files\Microsoft SQL Server\MSSQL16.SQLEXPRESS\MSSQL\DATA\fgssrDB_log.ldf' , SIZE = 8192KB , MAXSIZE = 2048GB , FILEGROWTH = 65536KB )
 WITH CATALOG_COLLATION = DATABASE_DEFAULT, LEDGER = OFF
GO
ALTER DATABASE [fgssrDB] SET COMPATIBILITY_LEVEL = 130
GO
IF (1 = FULLTEXTSERVICEPROPERTY('IsFullTextInstalled'))
begin
EXEC [fgssrDB].[dbo].[sp_fulltext_database] @action = 'enable'
end
GO
ALTER DATABASE [fgssrDB] SET ANSI_NULL_DEFAULT OFF
GO
ALTER DATABASE [fgssrDB] SET ANSI_NULLS OFF
GO
ALTER DATABASE [fgssrDB] SET ANSI_PADDING OFF
GO
ALTER DATABASE [fgssrDB] SET ANSI_WARNINGS OFF
GO
```



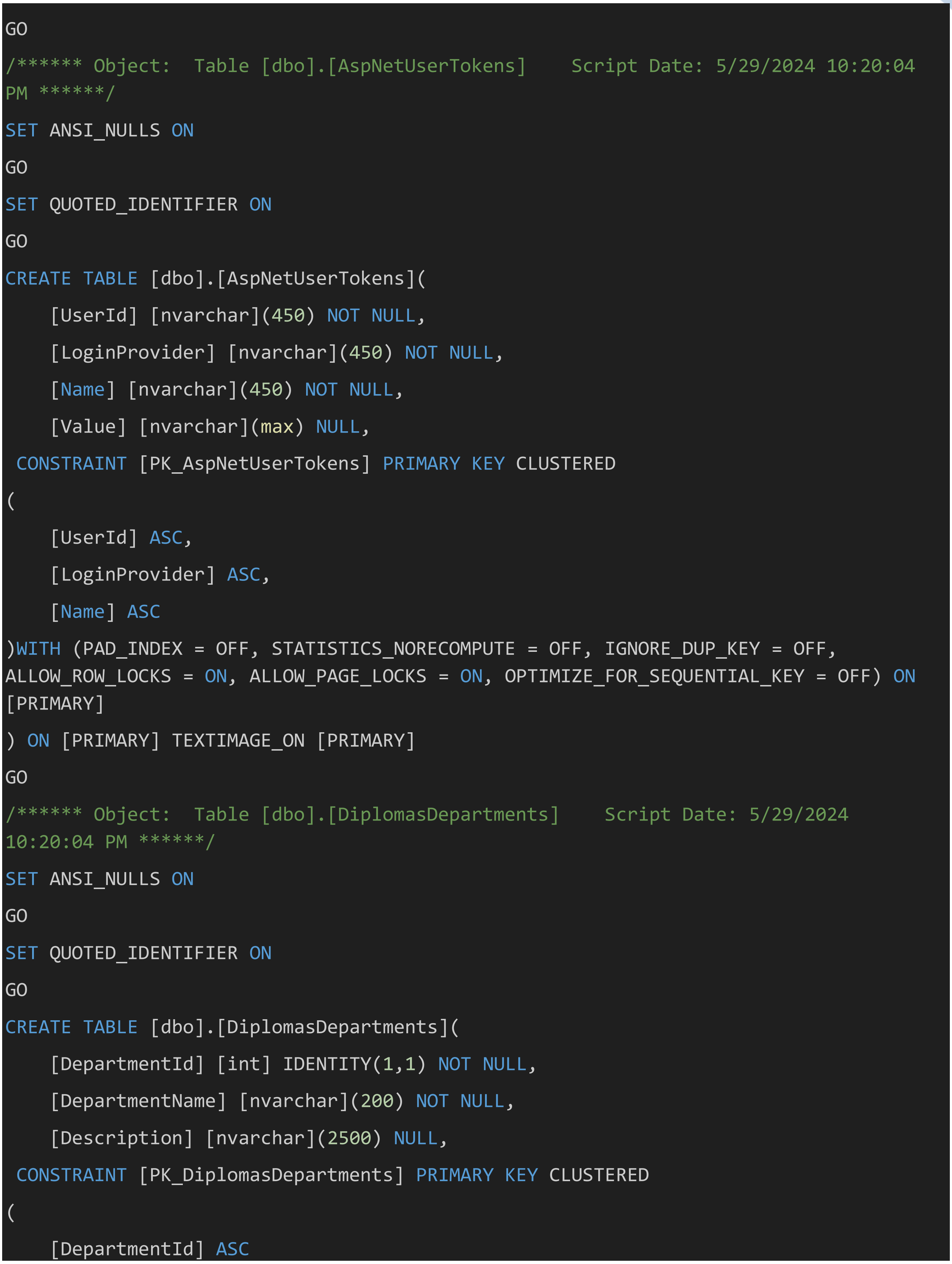
```
ALTER DATABASE [fgssrDB] SET ARITHABORT OFF
GO
ALTER DATABASE [fgssrDB] SET AUTO_CLOSE ON
GO
ALTER DATABASE [fgssrDB] SET AUTO_SHRINK OFF
GO
ALTER DATABASE [fgssrDB] SET AUTO_UPDATE_STATISTICS ON
GO
ALTER DATABASE [fgssrDB] SET CURSOR_CLOSE_ON_COMMIT OFF
GO
ALTER DATABASE [fgssrDB] SET CURSOR_DEFAULT  GLOBAL
GO
ALTER DATABASE [fgssrDB] SET CONCAT_NULL_YIELDS_NULL OFF
GO
ALTER DATABASE [fgssrDB] SET NUMERIC_ROUNDABORT OFF
GO
ALTER DATABASE [fgssrDB] SET QUOTED_IDENTIFIER OFF
GO
ALTER DATABASE [fgssrDB] SET RECURSIVE_TRIGGERS OFF
GO
ALTER DATABASE [fgssrDB] SET  DISABLE_BROKER
GO
ALTER DATABASE [fgssrDB] SET AUTO_UPDATE_STATISTICS_ASYNC OFF
GO
ALTER DATABASE [fgssrDB] SET DATE_CORRELATION_OPTIMIZATION OFF
GO
ALTER DATABASE [fgssrDB] SET TRUSTWORTHY OFF
GO
ALTER DATABASE [fgssrDB] SET ALLOW_SNAPSHOT_ISOLATION OFF
GO
ALTER DATABASE [fgssrDB] SET PARAMETERIZATION SIMPLE
GO
ALTER DATABASE [fgssrDB] SET READ_COMMITTED_SNAPSHOT ON
GO
```

```
ALTER DATABASE [fgssrDB] SET HONOR_BROKER_PRIORITY OFF
GO
ALTER DATABASE [fgssrDB] SET RECOVERY SIMPLE
GO
ALTER DATABASE [fgssrDB] SET  MULTI_USER
GO
ALTER DATABASE [fgssrDB] SET PAGE_VERIFY CHECKSUM
GO
ALTER DATABASE [fgssrDB] SET DB_CHAINING OFF
GO
ALTER DATABASE [fgssrDB] SET FILESTREAM( NON_TRANSACTED_ACCESS = OFF )
GO
ALTER DATABASE [fgssrDB] SET TARGET_RECOVERY_TIME = 60 SECONDS
GO
ALTER DATABASE [fgssrDB] SET DELAYED_DURABILITY = DISABLED
GO
ALTER DATABASE [fgssrDB] SET ACCELERATED_DATABASE_RECOVERY = OFF
GO
ALTER DATABASE [fgssrDB] SET QUERY_STORE = OFF
GO
USE [fgssrDB]
GO
/****** Object:  Table [dbo].[__EFMigrationsHistory]    Script Date: 5/29/2024
10:20:04 PM ******/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE TABLE [dbo].[__EFMigrationsHistory](
    [MigrationId] [nvarchar](150) NOT NULL,
    [ProductVersion] [nvarchar](32) NOT NULL,
 CONSTRAINT [PK___EFMigrationsHistory] PRIMARY KEY CLUSTERED
(
    [MigrationId] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON, OPTIMIZE_FOR_SEQUENTIAL_KEY = OFF) ON [PRIMARY]
) ON [PRIMARY]
GO
/****** Object:  Table [dbo].[AspNetRoleClaims]    Script Date: 5/29/2024 10:20:04 PM ******/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE TABLE [dbo].[AspNetRoleClaims](
    [Id] [int] IDENTITY(1,1) NOT NULL,
    [RoleId] [nvarchar](450) NOT NULL,
    [ClaimType] [nvarchar](max) NULL,
    [ClaimValue] [nvarchar](max) NULL,
 CONSTRAINT [PK_AspNetRoleClaims] PRIMARY KEY CLUSTERED
(
    [Id] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON, OPTIMIZE_FOR_SEQUENTIAL_KEY = OFF) ON [PRIMARY]
) ON [PRIMARY] TEXTIMAGE_ON [PRIMARY]
GO
/****** Object:  Table [dbo].[AspNetRoles]    Script Date: 5/29/2024 10:20:04 PM ******/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE TABLE [dbo].[AspNetRoles](
    [Id] [nvarchar](450) NOT NULL,
    [Name] [nvarchar](256) NULL,
    [NormalizedName] [nvarchar](256) NULL,
    [ConcurrencyStamp] [nvarchar](max) NULL,
 CONSTRAINT [PK_AspNetRoles] PRIMARY KEY CLUSTERED
```



```
(
    [Id] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,
ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON, OPTIMIZE_FOR_SEQUENTIAL_KEY = OFF) ON
[PRIMARY]
) ON [PRIMARY] TEXTIMAGE_ON [PRIMARY]
GO
/****** Object:  Table [dbo].[AspNetUserClaims]    Script Date: 5/29/2024 10:20:04
PM ******/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE TABLE [dbo].[AspNetUserClaims](
    [Id] [int] IDENTITY(1,1) NOT NULL,
    [UserId] [nvarchar](450) NOT NULL,
    [ClaimType] [nvarchar](max) NULL,
    [ClaimValue] [nvarchar](max) NULL,
 CONSTRAINT [PK_AspNetUserClaims] PRIMARY KEY CLUSTERED
(
    [Id] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,
ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON, OPTIMIZE_FOR_SEQUENTIAL_KEY = OFF) ON
[PRIMARY]
) ON [PRIMARY] TEXTIMAGE_ON [PRIMARY]
GO
/****** Object:  Table [dbo].[AspNetUserLogins]    Script Date: 5/29/2024 10:20:04
PM ******/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE TABLE [dbo].[AspNetUserLogins](
    [LoginProvider] [nvarchar](450) NOT NULL,
    [ProviderKey] [nvarchar](450) NOT NULL,
    [ProviderDisplayName] [nvarchar](max) NULL,
    [UserId] [nvarchar](450) NOT NULL,
 CONSTRAINT [PK_AspNetUserLogins] PRIMARY KEY CLUSTERED
(
    [LoginProvider] ASC,
    [ProviderKey] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,
ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON, OPTIMIZE_FOR_SEQUENTIAL_KEY = OFF) ON
[PRIMARY]
) ON [PRIMARY] TEXTIMAGE_ON [PRIMARY]
GO
/****** Object:  Table [dbo].[AspNetUserRoles]    Script Date: 5/29/2024 10:20:04 PM
******/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE TABLE [dbo].[AspNetUserRoles](
    [UserId] [nvarchar](450) NOT NULL,
    [RoleId] [nvarchar](450) NOT NULL,
 CONSTRAINT [PK_AspNetUserRoles] PRIMARY KEY CLUSTERED
(
    [UserId] ASC,
    [RoleId] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,
ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON, OPTIMIZE_FOR_SEQUENTIAL_KEY = OFF) ON
[PRIMARY]
) ON [PRIMARY]
GO
/****** Object:  Table [dbo].[AspNetUsers]    Script Date: 5/29/2024 10:20:04 PM
******/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE TABLE [dbo].[AspNetUsers](
    [Id] [nvarchar](450) NOT NULL,
```

```sql
    [UserName] [nvarchar](256) NULL,
    [NormalizedUserName] [nvarchar](256) NULL,
    [Email] [nvarchar](256) NULL,
    [NormalizedEmail] [nvarchar](256) NULL,
    [EmailConfirmed] [bit] NOT NULL,
    [PasswordHash] [nvarchar](max) NULL,
    [SecurityStamp] [nvarchar](max) NULL,
    [ConcurrencyStamp] [nvarchar](max) NULL,
    [PhoneNumber] [nvarchar](max) NULL,
    [PhoneNumberConfirmed] [bit] NOT NULL,
    [TwoFactorEnabled] [bit] NOT NULL,
    [LockoutEnd] [datetimeoffset](7) NULL,
    [LockoutEnabled] [bit] NOT NULL,
    [AccessFailedCount] [int] NOT NULL,
    [Address] [nvarchar](250) NOT NULL,
    [ApplicationStatus] [nvarchar](200) NULL,
    [BirthDate] [date] NOT NULL,
    [EmailCodeCreated] [datetime2](7) NOT NULL,
    [EmailVerifyCode] [nvarchar](20) NULL,
    [FullNameArabic] [nvarchar](250) NOT NULL,
    [FullNameEnglish] [nvarchar](250) NOT NULL,
    [Gender] [nvarchar](10) NOT NULL,
    [GraduationCertUpload] [nvarchar](100) NOT NULL,
    [PhoneCodeCreated] [datetime2](7) NOT NULL,
    [PhoneConfirmed] [bit] NOT NULL,
    [PhoneVerifyCode] [nvarchar](20) NULL,
    [ProfileImage] [nvarchar](200) NULL,
    [SectionId] [int] NULL,
 CONSTRAINT [PK_AspNetUsers] PRIMARY KEY CLUSTERED
(
    [Id] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON, OPTIMIZE_FOR_SEQUENTIAL_KEY = OFF) ON [PRIMARY]
) ON [PRIMARY] TEXTIMAGE_ON [PRIMARY]
```

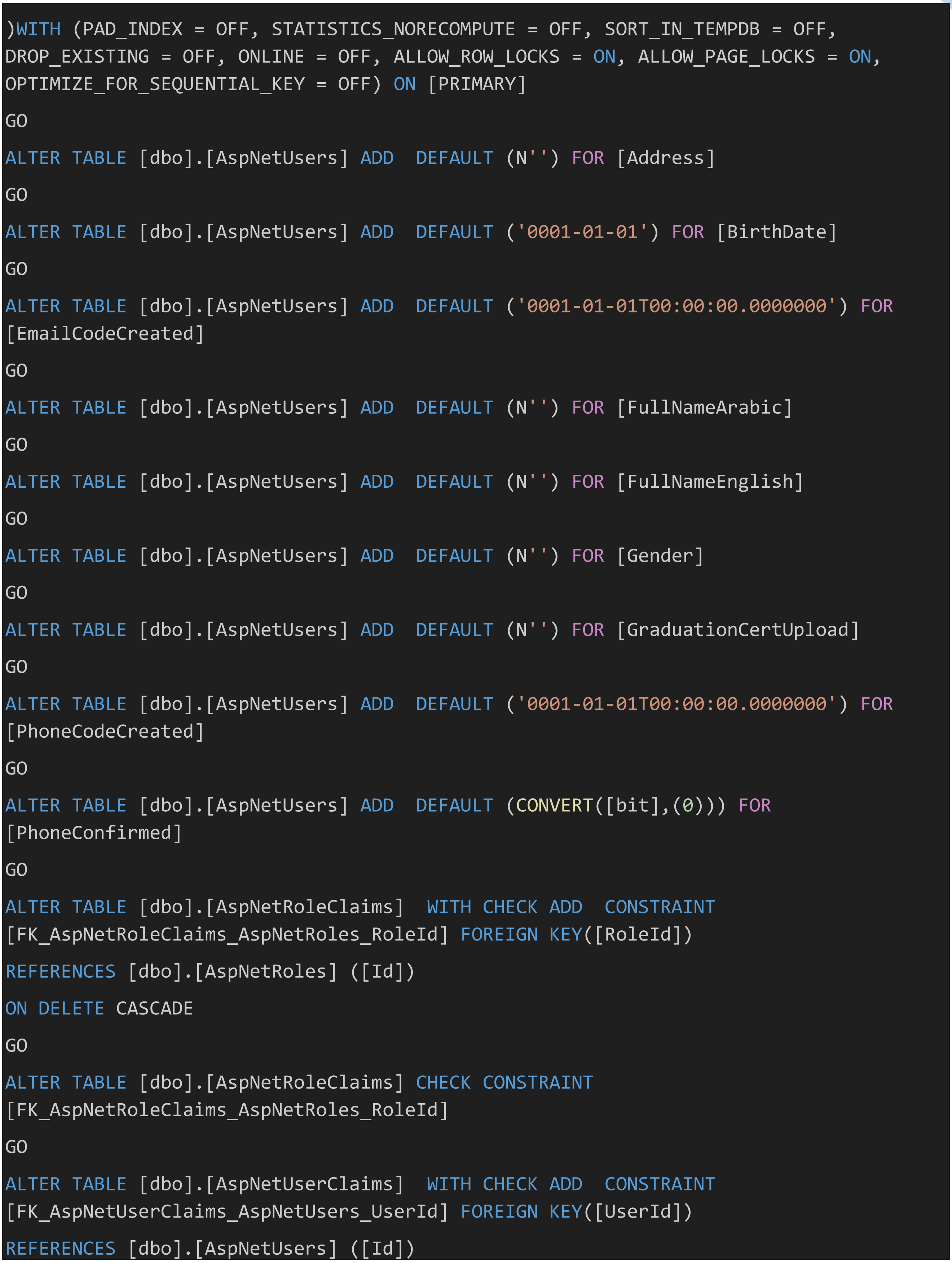


```
GO
/****** Object:  Table [dbo].[AspNetUserTokens]    Script Date: 5/29/2024 10:20:04 PM ******/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE TABLE [dbo].[AspNetUserTokens](
    [UserId] [nvarchar](450) NOT NULL,
    [LoginProvider] [nvarchar](450) NOT NULL,
    [Name] [nvarchar](450) NOT NULL,
    [Value] [nvarchar](max) NULL,
 CONSTRAINT [PK_AspNetUserTokens] PRIMARY KEY CLUSTERED
(
    [UserId] ASC,
    [LoginProvider] ASC,
    [Name] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON, OPTIMIZE_FOR_SEQUENTIAL_KEY = OFF) ON [PRIMARY]
) ON [PRIMARY] TEXTIMAGE_ON [PRIMARY]
GO
/****** Object:  Table [dbo].[DiplomasDepartments]    Script Date: 5/29/2024 10:20:04 PM ******/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE TABLE [dbo].[DiplomasDepartments](
    [DepartmentId] [int] IDENTITY(1,1) NOT NULL,
    [DepartmentName] [nvarchar](200) NOT NULL,
    [Description] [nvarchar](2500) NULL,
 CONSTRAINT [PK_DiplomasDepartments] PRIMARY KEY CLUSTERED
(
    [DepartmentId] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF, 
ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON, OPTIMIZE_FOR_SEQUENTIAL_KEY = OFF) ON 
[PRIMARY]
) ON [PRIMARY]
GO
/****** Object:  Table [dbo].[DiplomasSections]    Script Date: 5/29/2024 10:20:04 
PM ******/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE TABLE [dbo].[DiplomasSections](
    [SectionId] [int] IDENTITY(1,1) NOT NULL,
    [SectionName] [nvarchar](200) NOT NULL,
    [Description] [nvarchar](2500) NULL,
    [isActive] [bit] NOT NULL,
    [SectionImage] [nvarchar](1000) NOT NULL,
    [DepartmentId] [int] NOT NULL,
 CONSTRAINT [PK_DiplomasSections] PRIMARY KEY CLUSTERED 
(
    [SectionId] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF, 
ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON, OPTIMIZE_FOR_SEQUENTIAL_KEY = OFF) ON 
[PRIMARY]
) ON [PRIMARY]
GO
/****** Object:  Table [dbo].[Events]    Script Date: 5/29/2024 10:20:04 PM ******/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE TABLE [dbo].[Events](
    [EventId] [int] IDENTITY(1,1) NOT NULL,
    [EventTitle] [nvarchar](250) NOT NULL,
    [EventDescription] [nvarchar](2500) NOT NULL,
    [EventLocation] [nvarchar](1500) NOT NULL,
    [Time] [int] NOT NULL,
    [DateDay] [int] NOT NULL,
    [DateMonth] [nvarchar](150) NOT NULL,
    [EventImage] [nvarchar](1000) NOT NULL,
 CONSTRAINT [PK_Events] PRIMARY KEY CLUSTERED
(
    [EventId] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,
ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON, OPTIMIZE_FOR_SEQUENTIAL_KEY = OFF) ON
[PRIMARY]
) ON [PRIMARY]
GO
/****** Object:  Table [dbo].[News]    Script Date: 5/29/2024 10:20:04 PM ******/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE TABLE [dbo].[News](
    [NewsId] [int] IDENTITY(1,1) NOT NULL,
    [NewsTitle] [nvarchar](250) NOT NULL,
    [NewsDescription] [nvarchar](2500) NOT NULL,
    [NewsDate] [date] NULL,
 CONSTRAINT [PK_News] PRIMARY KEY CLUSTERED
(
    [NewsId] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,
ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON, OPTIMIZE_FOR_SEQUENTIAL_KEY = OFF) ON
[PRIMARY]
) ON [PRIMARY]
GO
/****** Object:  Table [dbo].[Staff]    Script Date: 5/29/2024 10:20:04 PM ******/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
```

```
CREATE TABLE [dbo].[Staff](
    [StaffId] [int] IDENTITY(1,1) NOT NULL,
    [StaffName] [nvarchar](max) NOT NULL,
    [StaffPosition] [nvarchar](max) NOT NULL,
    [StaffImage] [nvarchar](1000) NOT NULL,
    [Biograpghy] [nvarchar](2500) NOT NULL,
    [Email] [nvarchar](1000) NOT NULL,
 CONSTRAINT [PK_Staff] PRIMARY KEY CLUSTERED
(
    [StaffId] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,
ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON, OPTIMIZE_FOR_SEQUENTIAL_KEY = OFF) ON
[PRIMARY]
) ON [PRIMARY] TEXTIMAGE_ON [PRIMARY]
GO
SET ANSI_PADDING ON
GO
/****** Object:  Index [IX_AspNetRoleClaims_RoleId]    Script Date: 5/29/2024
10:20:04 PM ******/
CREATE NONCLUSTERED INDEX [IX_AspNetRoleClaims_RoleId] ON [dbo].[AspNetRoleClaims]
(
    [RoleId] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, SORT_IN_TEMPDB = OFF,
DROP_EXISTING = OFF, ONLINE = OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON,
OPTIMIZE_FOR_SEQUENTIAL_KEY = OFF) ON [PRIMARY]
GO
SET ANSI_PADDING ON
GO
/****** Object:  Index [RoleNameIndex]    Script Date: 5/29/2024 10:20:04 PM ******/
CREATE UNIQUE NONCLUSTERED INDEX [RoleNameIndex] ON [dbo].[AspNetRoles]
(
    [NormalizedName] ASC
)
WHERE ([NormalizedName] IS NOT NULL)
```

```
WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, SORT_IN_TEMPDB = OFF,
IGNORE_DUP_KEY = OFF, DROP_EXISTING = OFF, ONLINE = OFF, ALLOW_ROW_LOCKS = ON,
ALLOW_PAGE_LOCKS = ON, OPTIMIZE_FOR_SEQUENTIAL_KEY = OFF) ON [PRIMARY]
GO
SET ANSI_PADDING ON
GO
/****** Object:  Index [IX_AspNetUserClaims_UserId]    Script Date: 5/29/2024
10:20:04 PM ******/
CREATE NONCLUSTERED INDEX [IX_AspNetUserClaims_UserId] ON [dbo].[AspNetUserClaims]
(
    [UserId] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, SORT_IN_TEMPDB = OFF,
DROP_EXISTING = OFF, ONLINE = OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON,
OPTIMIZE_FOR_SEQUENTIAL_KEY = OFF) ON [PRIMARY]
GO
SET ANSI_PADDING ON
GO
/****** Object:  Index [IX_AspNetUserLogins_UserId]    Script Date: 5/29/2024
10:20:04 PM ******/
CREATE NONCLUSTERED INDEX [IX_AspNetUserLogins_UserId] ON [dbo].[AspNetUserLogins]
(
    [UserId] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, SORT_IN_TEMPDB = OFF,
DROP_EXISTING = OFF, ONLINE = OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON,
OPTIMIZE_FOR_SEQUENTIAL_KEY = OFF) ON [PRIMARY]
GO
SET ANSI_PADDING ON
GO
/****** Object:  Index [IX_AspNetUserRoles_RoleId]    Script Date: 5/29/2024
10:20:04 PM ******/
CREATE NONCLUSTERED INDEX [IX_AspNetUserRoles_RoleId] ON [dbo].[AspNetUserRoles]
(
    [RoleId] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, SORT_IN_TEMPDB = OFF,
DROP_EXISTING = OFF, ONLINE = OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON,
OPTIMIZE_FOR_SEQUENTIAL_KEY = OFF) ON [PRIMARY]
GO
```

```sql
SET ANSI_PADDING ON
GO
/****** Object:  Index [EmailIndex]    Script Date: 5/29/2024 10:20:04 PM ******/
CREATE NONCLUSTERED INDEX [EmailIndex] ON [dbo].[AspNetUsers]
(
    [NormalizedEmail] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, SORT_IN_TEMPDB = OFF,
DROP_EXISTING = OFF, ONLINE = OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON,
OPTIMIZE_FOR_SEQUENTIAL_KEY = OFF) ON [PRIMARY]
GO
/****** Object:  Index [IX_AspNetUsers_SectionId]    Script Date: 5/29/2024 10:20:04
PM ******/
CREATE NONCLUSTERED INDEX [IX_AspNetUsers_SectionId] ON [dbo].[AspNetUsers]
(
    [SectionId] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, SORT_IN_TEMPDB = OFF,
DROP_EXISTING = OFF, ONLINE = OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON,
OPTIMIZE_FOR_SEQUENTIAL_KEY = OFF) ON [PRIMARY]
GO
SET ANSI_PADDING ON
GO
/****** Object:  Index [UserNameIndex]    Script Date: 5/29/2024 10:20:04 PM ******/
CREATE UNIQUE NONCLUSTERED INDEX [UserNameIndex] ON [dbo].[AspNetUsers]
(
    [NormalizedUserName] ASC
)
WHERE ([NormalizedUserName] IS NOT NULL)
WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, SORT_IN_TEMPDB = OFF,
IGNORE_DUP_KEY = OFF, DROP_EXISTING = OFF, ONLINE = OFF, ALLOW_ROW_LOCKS = ON,
ALLOW_PAGE_LOCKS = ON, OPTIMIZE_FOR_SEQUENTIAL_KEY = OFF) ON [PRIMARY]
GO
/****** Object:  Index [IX_DiplomasSections_DepartmentId]    Script Date: 5/29/2024
10:20:04 PM ******/
CREATE NONCLUSTERED INDEX [IX_DiplomasSections_DepartmentId] ON
[dbo].[DiplomasSections]
(
    [DepartmentId] ASC
```

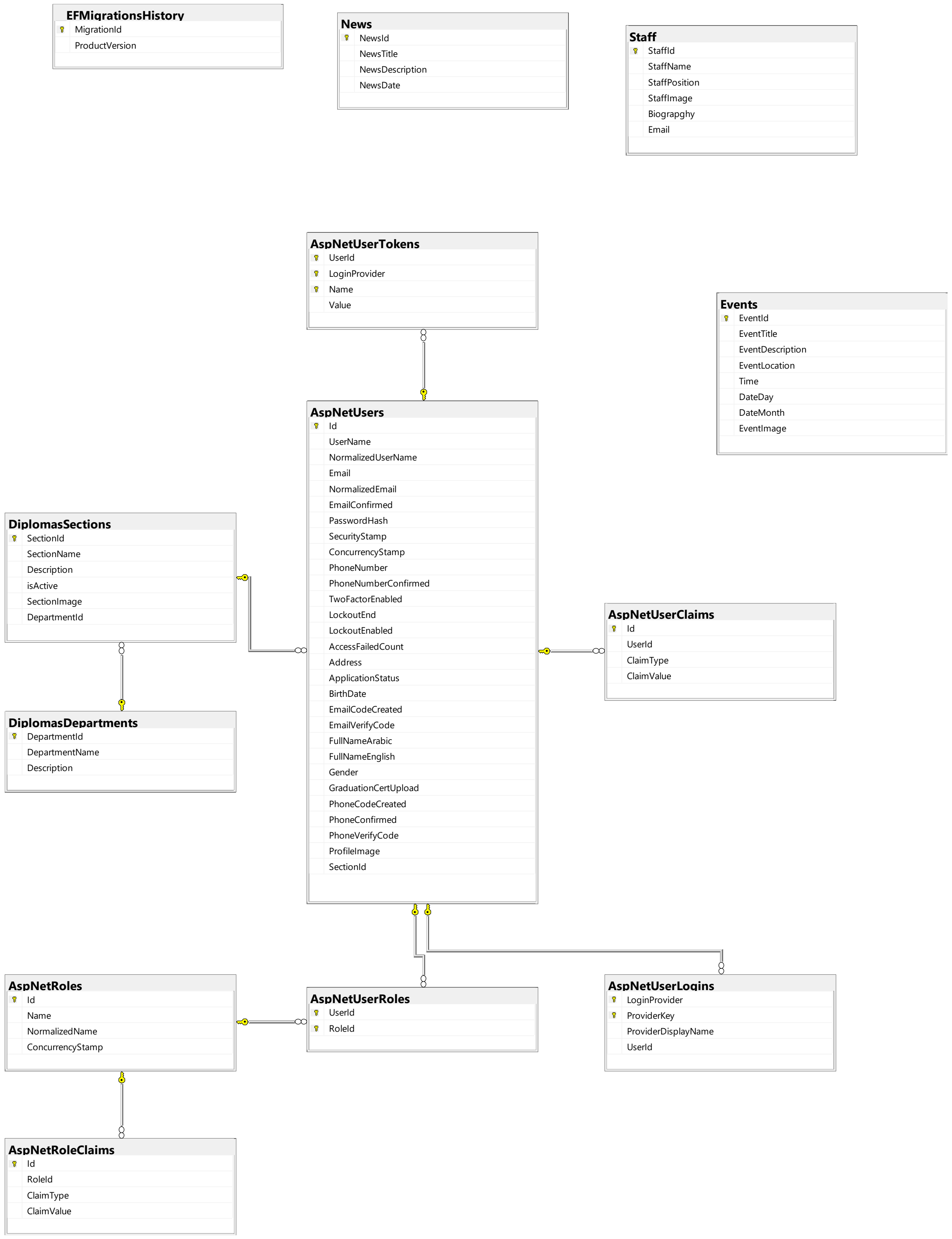


```
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, SORT_IN_TEMPDB = OFF,
DROP_EXISTING = OFF, ONLINE = OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON,
OPTIMIZE_FOR_SEQUENTIAL_KEY = OFF) ON [PRIMARY]
GO
ALTER TABLE [dbo].[AspNetUsers] ADD  DEFAULT (N'') FOR [Address]
GO
ALTER TABLE [dbo].[AspNetUsers] ADD  DEFAULT ('0001-01-01') FOR [BirthDate]
GO
ALTER TABLE [dbo].[AspNetUsers] ADD  DEFAULT ('0001-01-01T00:00:00.0000000') FOR
[EmailCodeCreated]
GO
ALTER TABLE [dbo].[AspNetUsers] ADD  DEFAULT (N'') FOR [FullNameArabic]
GO
ALTER TABLE [dbo].[AspNetUsers] ADD  DEFAULT (N'') FOR [FullNameEnglish]
GO
ALTER TABLE [dbo].[AspNetUsers] ADD  DEFAULT (N'') FOR [Gender]
GO
ALTER TABLE [dbo].[AspNetUsers] ADD  DEFAULT (N'') FOR [GraduationCertUpload]
GO
ALTER TABLE [dbo].[AspNetUsers] ADD  DEFAULT ('0001-01-01T00:00:00.0000000') FOR
[PhoneCodeCreated]
GO
ALTER TABLE [dbo].[AspNetUsers] ADD  DEFAULT (CONVERT([bit],(0))) FOR
[PhoneConfirmed]
GO
ALTER TABLE [dbo].[AspNetRoleClaims]  WITH CHECK ADD  CONSTRAINT
[FK_AspNetRoleClaims_AspNetRoles_RoleId] FOREIGN KEY([RoleId])
REFERENCES [dbo].[AspNetRoles] ([Id])
ON DELETE CASCADE
GO
ALTER TABLE [dbo].[AspNetRoleClaims] CHECK CONSTRAINT
[FK_AspNetRoleClaims_AspNetRoles_RoleId]
GO
ALTER TABLE [dbo].[AspNetUserClaims]  WITH CHECK ADD  CONSTRAINT
[FK_AspNetUserClaims_AspNetUsers_UserId] FOREIGN KEY([UserId])
REFERENCES [dbo].[AspNetUsers] ([Id])
```

```
ON DELETE CASCADE

GO

ALTER TABLE [dbo].[AspNetUserClaims] CHECK CONSTRAINT
[FK_AspNetUserClaims_AspNetUsers_UserId]

GO

ALTER TABLE [dbo].[AspNetUserLogins]  WITH CHECK ADD  CONSTRAINT
[FK_AspNetUserLogins_AspNetUsers_UserId] FOREIGN KEY([UserId])

REFERENCES [dbo].[AspNetUsers] ([Id])

ON DELETE CASCADE

GO

ALTER TABLE [dbo].[AspNetUserLogins] CHECK CONSTRAINT
[FK_AspNetUserLogins_AspNetUsers_UserId]

GO

ALTER TABLE [dbo].[AspNetUserRoles]  WITH CHECK ADD  CONSTRAINT
[FK_AspNetUserRoles_AspNetRoles_RoleId] FOREIGN KEY([RoleId])

REFERENCES [dbo].[AspNetRoles] ([Id])

ON DELETE CASCADE

GO

ALTER TABLE [dbo].[AspNetUserRoles] CHECK CONSTRAINT
[FK_AspNetUserRoles_AspNetRoles_RoleId]

GO

ALTER TABLE [dbo].[AspNetUserRoles]  WITH CHECK ADD  CONSTRAINT
[FK_AspNetUserRoles_AspNetUsers_UserId] FOREIGN KEY([UserId])

REFERENCES [dbo].[AspNetUsers] ([Id])

ON DELETE CASCADE

GO

ALTER TABLE [dbo].[AspNetUserRoles] CHECK CONSTRAINT
[FK_AspNetUserRoles_AspNetUsers_UserId]

GO

ALTER TABLE [dbo].[AspNetUsers]  WITH CHECK ADD  CONSTRAINT
[FK_AspNetUsers_DiplomasSections_SectionId] FOREIGN KEY([SectionId])

REFERENCES [dbo].[DiplomasSections] ([SectionId])

GO

ALTER TABLE [dbo].[AspNetUsers] CHECK CONSTRAINT
[FK_AspNetUsers_DiplomasSections_SectionId]

GO
```

```
ALTER TABLE [dbo].[AspNetUserTokens]  WITH CHECK ADD  CONSTRAINT
[FK_AspNetUserTokens_AspNetUsers_UserId] FOREIGN KEY([UserId])

REFERENCES [dbo].[AspNetUsers] ([Id])

ON DELETE CASCADE

GO

ALTER TABLE [dbo].[AspNetUserTokens] CHECK CONSTRAINT
[FK_AspNetUserTokens_AspNetUsers_UserId]

GO

ALTER TABLE [dbo].[DiplomasSections]  WITH CHECK ADD  CONSTRAINT
[FK_DiplomasSections_DiplomasDepartments_DepartmentId] FOREIGN KEY([DepartmentId])

REFERENCES [dbo].[DiplomasDepartments] ([DepartmentId])

GO

ALTER TABLE [dbo].[DiplomasSections] CHECK CONSTRAINT
[FK_DiplomasSections_DiplomasDepartments_DepartmentId]

GO

USE [master]

GO

ALTER DATABASE [fgssrDB] SET  READ_WRITE

GO
```

2. ERD Diagrams:



CHAPTER 3

AGILE METHODOLOGY

What is Agile

Agile is a timeboxed, iterative approach to software delivery that builds software

incrementally from the start of the project, instead of trying to deliver it all at once near the end.

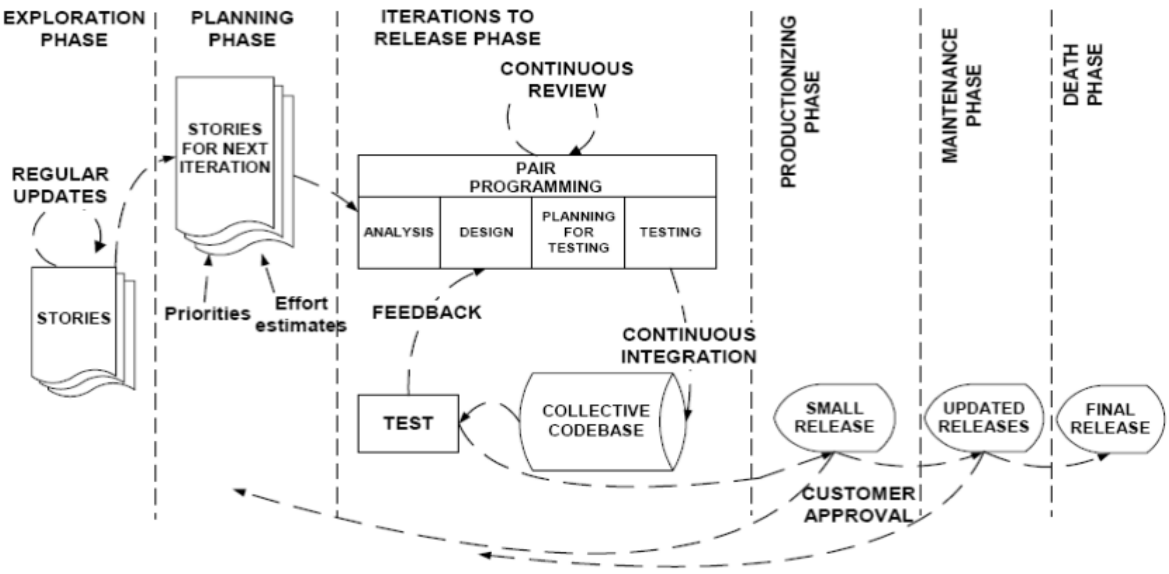
It works by breaking projects down into little bits of user functionality

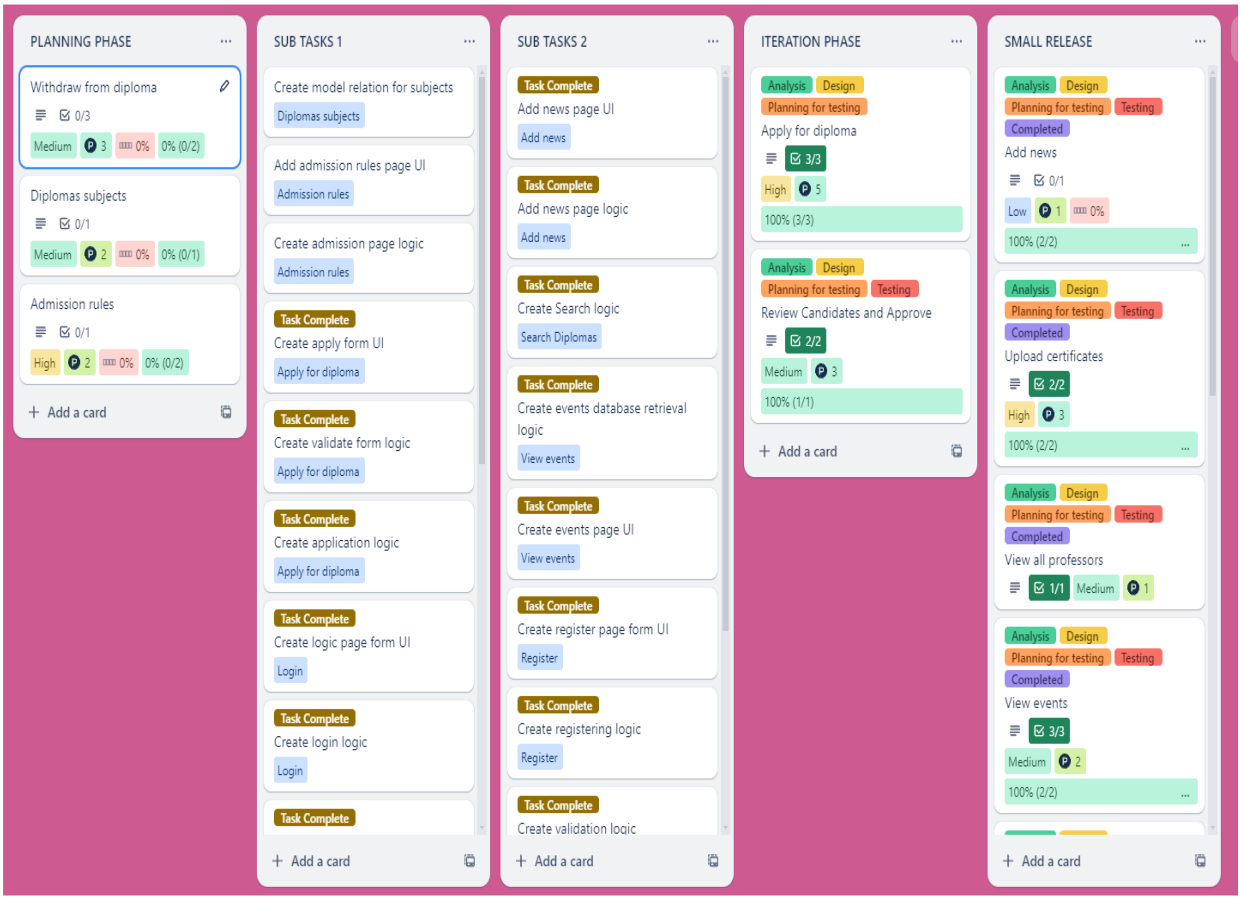
called user stories, prioritizing them, and then continuously delivering them in short cycles

Adopting Agile XP Methodology

In the development of our college website, our team members embraced the Agile XP (Extreme Programming) methodology to ensure efficient collaboration, rapid iteration, and high-quality results. By following the Agile principles of communication, simplicity, feedback, and courage, we aimed to deliver a website that meets the diverse needs of our college community while maintaining flexibility to adapt to evolving requirements. Through continuous communication, frequent releases, and a focus on teamwork, we strived to maximize value and deliver a website that reflects our commitment to excellence and innovation in higher education.

XP Life Cycle





PLANNING PHASE

Withdraw from diploma
☑ 0/3
Medium · 3 · 0% · 0% (0/2)

Diplomas subjects
☑ 0/1
Medium · 2 · 0% · 0% (0/1)

Admission rules
☑ 0/1
High · 2 · 0% · 0% (0/2)

+ Add a card

SUB TASKS 1

Create model relation for subjects
Diplomas subjects

Add admission rules page UI
Admission rules

Create admission page logic
Admission rules

Task Complete
Create apply form UI
Apply for diploma

Task Complete
Create validate form logic
Apply for diploma

Task Complete
Create application logic
Apply for diploma

Task Complete
Create logic page form UI
Login

Task Complete
Create login logic
Login

Task Complete

+ Add a card

SUB TASKS 2

Task Complete
Add news page UI
Add news

Task Complete
Add news page logic
Add news

Task Complete
Create Search logic
Search Diplomas

Task Complete
Create events database retrieval logic
View events

Task Complete
Create events page UI
View events

Task Complete
Create register page form UI
Register

Task Complete
Create registering logic
Register

Task Complete
Create validation logic

+ Add a card

ITERATION PHASE

Analysis · Design · Planning for testing
Apply for diploma
☑ 3/3
High · 5
100% (3/3)

Analysis · Design · Planning for testing · Testing
Review Candidates and Approve
☑ 2/2
Medium · 3
100% (1/1)

+ Add a card

SMALL RELEASE

Analysis · Design · Planning for testing · Testing · Completed
Add news
☑ 0/1
Low · 1 · 0%
100% (2/2)

Analysis · Design · Planning for testing · Testing · Completed
Upload certificates
☑ 2/2
High · 3
100% (2/2)

Analysis · Design · Planning for testing · Testing · Completed
View all professors
☑ 1/1 · Medium · 1

Analysis · Design · Planning for testing · Testing · Completed
View events
☑ 3/3
Medium · 2
100% (2/2)

+ Add a card

USER STORIES

ID	User story	Size
1	As a student I want to be able to login to the website	
2	As a student I want to be able to register to the website	5
3	As a student I want to be able to view all available programs	1
4	As a student I want to be able to view all diplomas	1
5	As a student I want to be able to search for any diploma	1
6	As a student I want to be able to apply for specific diploma	8
7	As a student I want to be have a profile page and be able to edit this page	3
8	As a student I want to be able to cancel my admission anytime	3
9	As an admin I want to be able to login as admin	5
10	As an admin I want to be able to register as admin	3
11	As an admin I want to be able to view all users in the website	1
12	As an admin I want to be able to edit user roles	8
13	As an admin I want to be able view all admissions	3
14	As an admin I want to be able to edit user admission status	5
15	As an admin I want to be able to edit events	5
16	As an admin I want to be able to add new events	3
17	As an admin I want to be able to delete an event	1
18	As an admin I want to be able to edit diplomas	3
19	As an admin I want to add new diplomas	1
20	As an admin I want to be able to delete diplomas	3
21	As an admin I want to be able to add new department	3
22	As an admin I want to be able to edit department	5
23	As an admin I want to be able to delete a department	3
24	As an admin I want to be able to add new staff member	1
25	As an admin I want to be able to edit staff member info	3
26	As an admin I want to be able to delete staff member	3
27	As an admin I want to be able to add new news	3
28	As an admin I want to be able to edit news	5
29	As an admin I want to be able to delete news	3

PLAN BY FEATURE

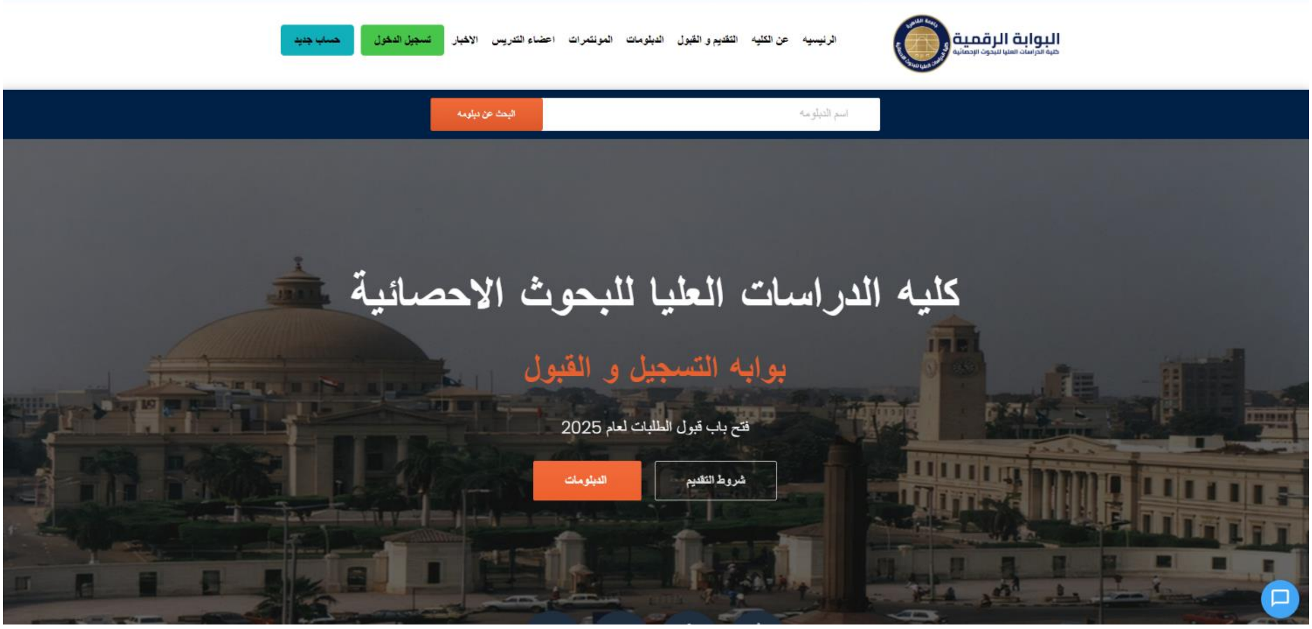
Task	Task Name	Owned By	Chief programmer	Priority
1	Database Design		Sameh Taha	
1.1	Er diagram	Sameh Taha		High
1.2	Relational schema			High
1.3	Installing SQL Server	Naglaa Ahmed		High
2	Front-end development		Naglaa Ahmed	
2.1	Design homepage	Naglaa Ahmed		Medium
2.2	Login page			High
2.3	Register page	Ahmed Moataz		High
2.4	Diplomas page			High
2.5	Departments page			Medium
2.6	Events page			Low
2.7	News page			Low
2.8	Admission page			High
2.9	Dashboard			Medium
2.10	Profile page			Medium
3	Back-end development		Ahmed Moataz	
3.1	Login	Ahmed Moataz		High
3.2	Register			High
3.3	Admission	Sameh Taha		High
3.4	Search diploma			Low
3.5	Configuring SQL server			high
3.6	Add diplomas			Medium
3.7	Edit diploma			Medium
3.8	Add event			Medium
3.9	Edit event			Medium
3.10	Edit profile			High
3.11	Edit admission status			High
3.12	Edit user roles			Medium
4	Testing		Sameh Taha	
4.1	Login	Sameh Taha		High
4.2	Register			High
4.3	Admission	Ahmed Moataz		High
4.4	Add diploma			Medium

4.5	Edit diploma	Naglaa Ahmed		Medium
4.6	Add event			Medium
4.7	Edit event			Medium
4.8	View users			Medium
4.9	Edit admission stats			High
4.10	Edit user roles			Low
4.11	Add department			Medium
4.12	Edit department			Medium
5	Deployment		Ahmed Moataz	
5.1	Hosting provider	Ahmed Moataz Naglaa Ahmed Sameh Taha		High
5.2	Publish website			High

CHAPTER 4

GRAPHICAL USER INTERFACE

1. Home Page:

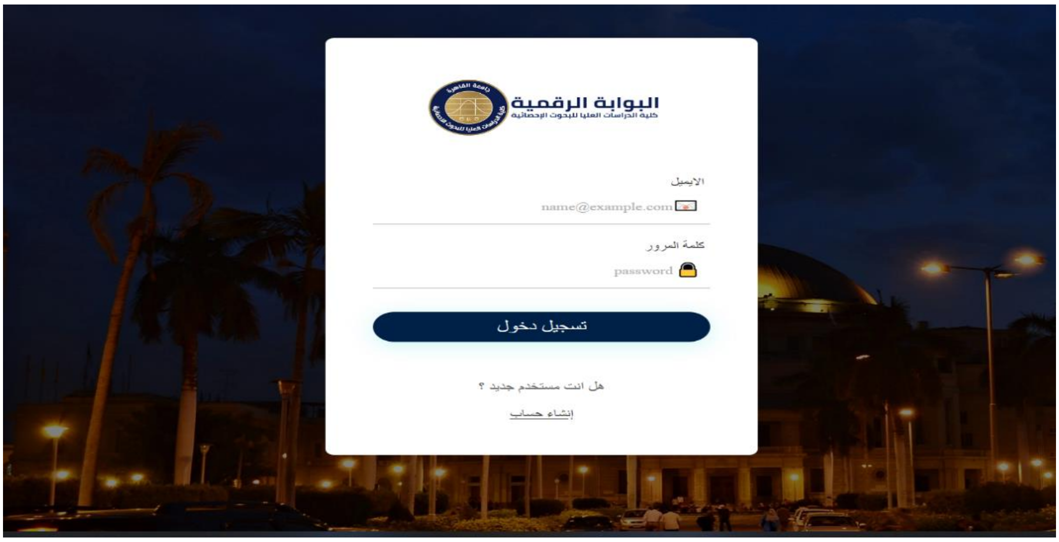


كليه الدراسات العليا للبحوث الاحصائية

بوابه التسجيل و القبول

فتح باب قبول الطلبات لعام 2025

2. Login Page:



البوابة الرقمية
كلية الدراسات العليا للبحوث الإحصائية

الايميل

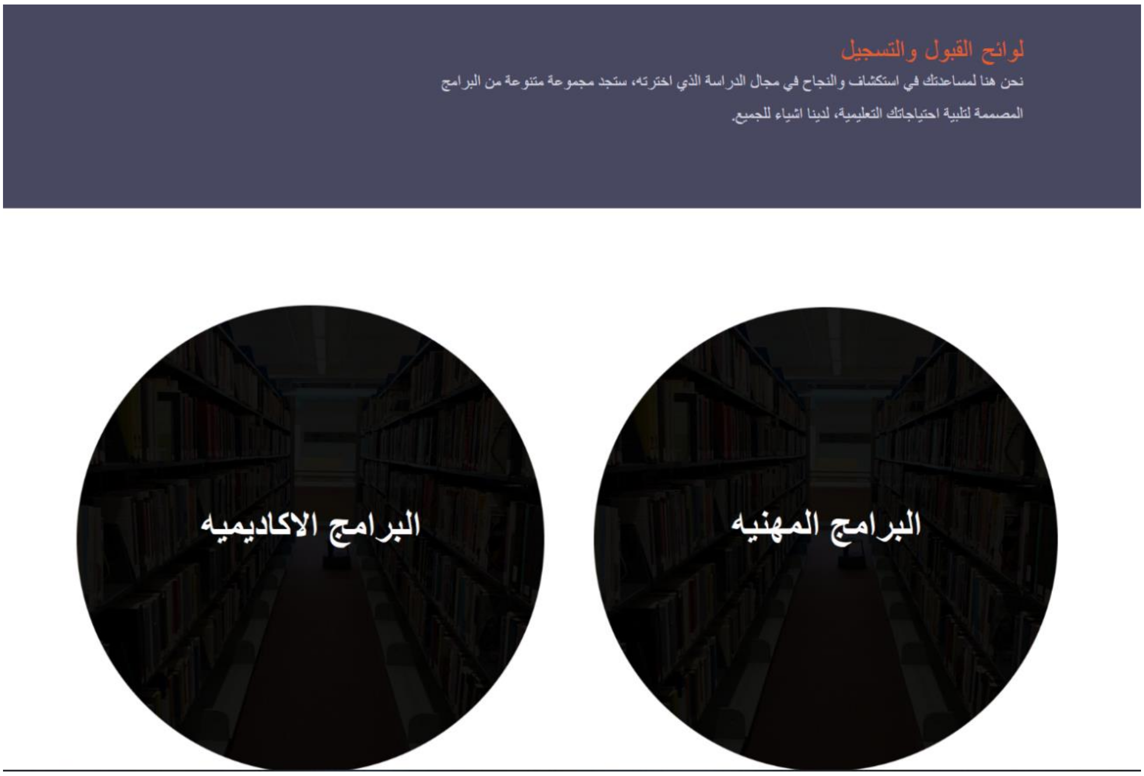
name@example.com

كلمة المرور

password

هل انت مستخدم جديد ؟

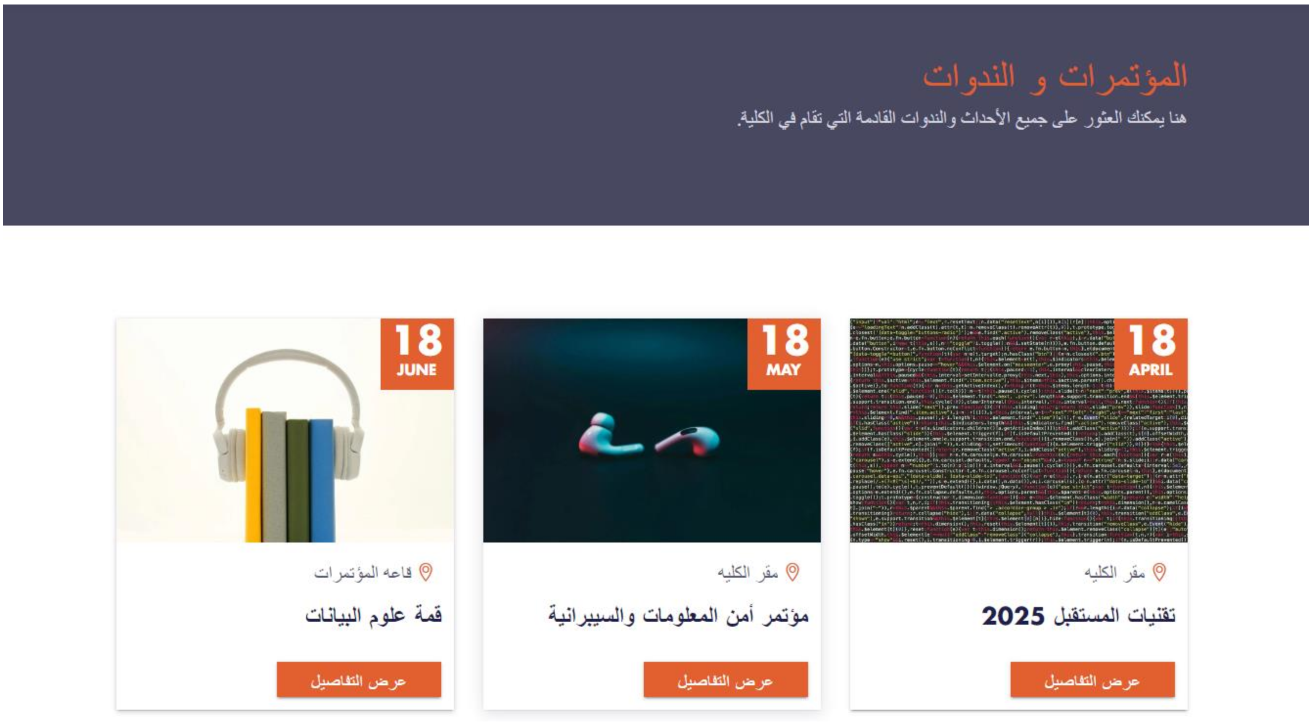
3. Admission Regulations:



لوائح القبول والتسجيل

نحن هنا لمساعدتك في استكشاف والنجاح في مجال الدراسة الذي اخترته، ستجد مجموعة متنوعة من البرامج
المصممة لتلبية احتياجاتك التعليمية، لدينا اشياء للجميع.

4. College Events:



المؤتمرات و الندوات

هنا يمكنك العثور على جميع الأحداث والندوات القادمة التي تقام في الكلية.

18 APRIL

مقر الكليه

تقنيات المستقبل 2025

عرض التفاصيل

18 MAY

مقر الكليه

مؤتمر أمن المعلومات والسيبرانية

عرض التفاصيل

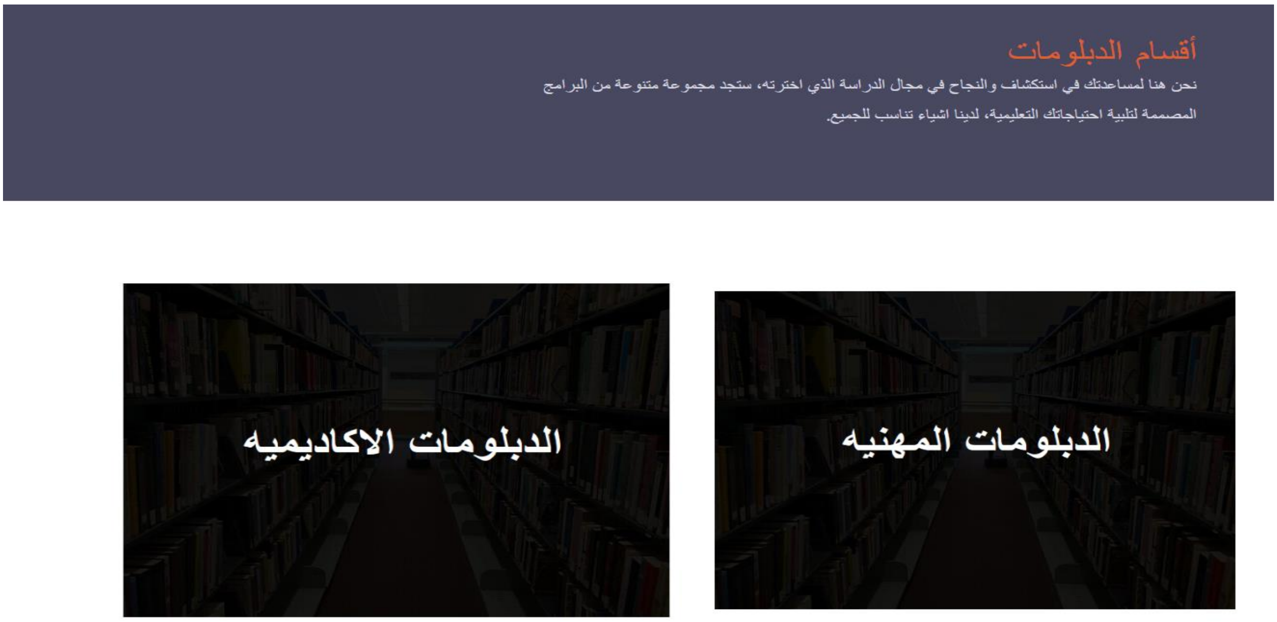
18 JUNE

قاعه المؤتمرات

قمة علوم البيانات

عرض التفاصيل

5. Diplomas:



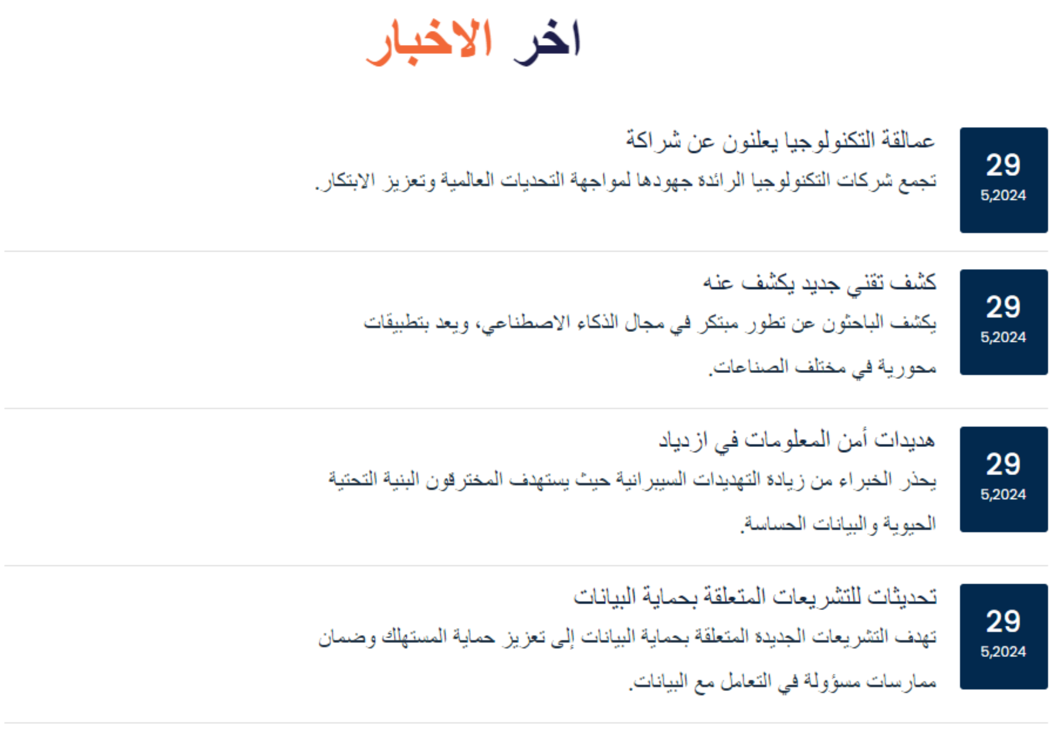
أقسام الدبلومات

نحن هنا لمساعدتك في استكشاف والنجاح في مجال الدراسة الذي اخترته، ستجد مجموعة متنوعة من البرامج المصممة لتلبية احتياجاتك التعليمية، لدينا اشياء تناسب للجميع.

الدبلومات المهنيه

الدبلومات الاكاديميه

6. News Page:



اخر الاخبار

29
5,2024

عمالقة التكنولوجيا يعلنون عن شراكة

تجمع شركات التكنولوجيا الرائدة جهودها لمواجهة التحديات العالمية وتعزيز الابتكار.

29
5,2024

كشف تقني جديد يكشف عنه

يكشف الباحثون عن تطور مبتكر في مجال الذكاء الاصطناعي، ويعد بتطبيقات
محورية في مختلف الصناعات.

29
5,2024

هديدات أمن المعلومات في ازدياد

يحذر الخبراء من زيادة التهديدات السيبرانية حيث يستهدف المخترقون البنية التحتية
الحيوية والبيانات الحساسة.

29
5,2024

تحديثات للتشريعات المتعلقة بحماية البيانات

تهدف التشريعات الجديدة المتعلقة بحماية البيانات إلى تعزيز حماية المستهلك وضمان
ممارسات مسؤولة في التعامل مع البيانات.

CHAPTER 5

CONCLUSION AND FUTURE WORK

The FGSSR Faculty website is designed to be a comprehensive, user-friendly platform for managing faculty resources, enhancing communication, and streamlining administrative processes. By leveraging the latest web technologies, this .NET Core application will provide a secure, high-performance environment tailored to meet the needs of faculty members, administrators, and students. The features outlined in this document ensure that the website will support essential functions such as user management, course management, resource allocation, and communication, thereby fostering an efficient and collaborative academic environment. This SRS document details the necessary requirements and specifications to develop and deploy the FGSSR Faculty website successfully, aligning with the institution's goals and user expectations.

The FGSSR Faculty website will include the following key features:

USER MANAGEMENT

- **User Registration and Login**: Secure user authentication and authorization using the ASP.NET Identity framework.
- **Role-Based Access Control**: Implement varying access levels for administrators, faculty members, and students.
- **Profile Management**: Allow users to update their personal information and manage their profiles.

COURSE MANAGEMENT

- **Course Creation and Editing**: Enable faculty members to create and modify course details, including syllabi, schedules, and materials.
- **Student Enrollment**: Manage student enrollment in courses, track attendance, and maintain academic records.
- **Gradebook**: Allow faculty to enter and manage grades, and provide students with access to their academic performance.

RESOURCE ALLOCATION

- **Resource Scheduling**: Allocate resources such as classrooms, labs, and equipment to courses based on availability.
- **Resource Dashboard**: Provide an overview of resource utilization and availability.
- **Notifications**: Send alerts for upcoming classes, assignment deadlines, and resource bookings.

COMMUNICATION TOOLS

- **Messaging System**: Internal messaging system for streamlined communication between faculty, students, and administrators.
- **Email Notifications**: Automated email updates for important events, assignments, and announcements.
- **Discussion Forums**: Provide a platform for course-related discussions and knowledge sharing.

REPORTING AND ANALYTICS

- **Custom Reports**: Generate detailed reports on course enrollment, student performance, and resource utilization.
- **Data Visualization**: Present key metrics and KPIs through graphical representations.
- **Export Functionality**: Allow users to export reports in various formats including PDF, Excel, and CSV.

PERFORMANCE AND SCALABILITY

- **Load Balancing**: Implement load balancing to distribute incoming traffic efficiently and ensure optimal performance.
- **Caching Mechanisms**: Utilize caching to reduce load times and enhance user experience.
- **Scalable Architecture**: Design the system to support future growth and an increased user base.

SUPPORT AND MAINTENANCE

- **Help Center**: Provide a comprehensive help center with FAQs, user guides, and tutorials.
- **Customer Support**: Offer dedicated support for resolving user issues and queries.
- **Regular Updates**: Ensure continuous improvement and regular updates to the system.

These features collectively aim to create a robust, efficient, and user-friendly faculty website that will significantly enhance FGSSR’s ability to manage academic and administrative tasks. This application will not only meet the current needs of the faculty and students but also scale to accommodate future growth and technological advancements.

REFERENCES

https://www.w3schools.com/cs/index.php

https://www.w3schools.com/html/default.asp

https://getbootstrap.com/docs/4.6/getting-started/introduction/

https://materializecss.com/getting-started.html

https://www.w3schools.com/jquery/default.asp

https://learn.microsoft.com/en-us/aspnet/core/introduction-to-aspnet-core?view=aspnetcore-8.0