BSc Project CS/CE/IS

Department

Project ID: UQU-CS/CE/IS-YYYY-xx

Month Year

Ghali Center

Bridging the Gap for Inclusive Learning and Volunteering

Dept. of _____

Faculty of Computer and Information Systems

Umm Al-Qura University, KSA

This project report is submitted to the Department of _____ at Umm Al-Qura University in partial fulfillment of the requirements for the degree of Bachelor of Science in Computer Engineering/Computer Science/Information Systems.

Author(s): Firstname Lastname

Address: If applicable

E-mail: If applicable, a long-term e-mail (not your student e-mail)

University supervisor(s):

Firstname Lastname

Department name

Co-supervisor(if applicable**):**

Firstname Lastname

Company/Organizatio
n full name Address:

Phone: International standard, e.g. use +

Dept. of _____ Internet: http://uqu.edu.sa

Faculty of Computer and Information Systems Phone: +966 xxxxxxxxx

Umm Al Qura University Fax : +966 xxxxxxxxx Kingdom
of Saudi Arabia

Intellectual Property Right Declaration

This is to declare that the work under the supervision of _____ having title " _____" carried out in partial fulfillment of the requirements of Bachelor of Science in _____, is the sole property of the Umm Al Qura University and the respective supervisor and is protected under the intellectual property right laws and conventions. It can only be considered/ used for purposes like extension for further enhancement, product development, adoption for commercial/organizational usage, etc., with the permission of the University and respective supervisor.

This above statement applies to all students and faculty members.

Date: _____

Author(s):

Name: Firstname Lastname Signature: _____

Name: Firstname Lastname Signature: _____

Name: Firstname Lastname Signature: _____

Supervisor(s):

Name: Firstname Lastname Signature: _____

Anti-Plagiarism Declaration

This is to declare that the above publication produced under the supervision of_____having title

"_____" is the sole contribution of the author(s) and no part hereof has been reproduced illegally (cut and paste) which can be considered as Plagiarism. All referenced parts have been used to argue the idea and have been cited properly. I/We will be responsible and liable for any consequence if violation of this declaration is proven.

Date: _____

Author(s):

Name: Firstname Lastname Signature: _____

Name: Firstname Lastname Signature: _____ Name: Firstname Lastname Signature: _____

ACKNOWLEDGMENTS

This project would not have been possible without the guidance, support, and encouragement of many individuals. I would like to express my heartfelt gratitude to:

- **My supervisor,** [Supervisor's Name], for their invaluable guidance, insightful feedback, and unwavering support throughout the project. Their expertise and mentorship have been instrumental in shaping this project and my understanding of software development.
- **My fellow project team members,** [Team Member Names], for their collaborative spirit, dedication, and willingness to share knowledge and expertise. Working together, we were able to overcome challenges and achieve significant milestones.
- **The Ghali Center staff,** for their generous support, valuable insights, and contributions to the project. Their firsthand knowledge and expertise were essential in ensuring the application met the real-world needs of the center and its users.
- **My family and friends,** for their unwavering support, encouragement, and belief in my abilities. Their love and encouragement have provided me with the strength and motivation to complete this project.

This project has been an enriching and fulfilling experience, and I am grateful to all those who have contributed to its success.

ABSTRACT

The "Ghali Center" application is a web-based platform designed to bridge the gap between university students and students with special needs, fostering an inclusive and supportive learning environment at Umm Al-Qura University. The application provides a range of features to promote community engagement and enhance access to educational resources:

- **Volunteer Opportunities:** University students can browse, search, and register for volunteer opportunities at the Ghali Center.
- **Educational Lessons and Training Courses:** Students with special needs can access a diverse library of educational lessons and training courses, tailored to their individual needs.
- **Communication Tools:** A messaging system facilitates communication between students, teachers, and administrators.
- **Zoom Integration:** The application seamlessly integrates with Zoom, allowing teachers to host online educational sessions.

The Ghali Center application is built with accessibility and security in mind, incorporating features like screen reader compatibility, adjustable text sizes, data encryption, and secure authentication protocols. It is hosted on a robust cloud-based infrastructure, ensuring scalability and performance.

The project successfully addressed the challenges of creating a user-friendly and inclusive platform that meets the specific needs of diverse user groups. User feedback highlights the application's effectiveness in facilitating volunteer engagement, enhancing access to education, and fostering a sense of community. The Ghali Center application demonstrates the potential of technology to create more equitable and supportive learning environments within higher education institutions.

TABLE OF CONTENT

CHAPTER 1: INTRODUCTION

1.1 Project Overview

The "Ghali Center" application emerges as a vital tool for fostering an inclusive and supportive learning environment within Umm Al-Qura University. This application aims to bridge the gap between university students and students with special needs, facilitating meaningful interactions and shared experiences. It acts as a platform for both groups to connect, learn, and contribute to a more inclusive community.

The Ghali Center addresses the critical need for accessible and engaging resources for students with special needs, empowering them to participate actively in university life and access valuable learning opportunities. Simultaneously, it empowers university students to engage in impactful volunteering, providing them with the tools to support their peers and contribute to a more inclusive society.

1.2 Project Scope

This project encompasses the development of a comprehensive web-based application, accessible via desktop and mobile devices. The application includes:

1.2.1 User Management System

A robust system to manage user accounts, roles, and permissions for different user types (university students, students with special needs, teachers, and administrators).

1.2.2 Volunteer Opportunity Management

A feature allowing university students to browse, search, and sign up for volunteer opportunities at the Ghali Center.

1.2.3 Educational Lesson Management

A module dedicated to providing a wide range of educational lessons covering various academic subjects, skill development, and personal growth for students with special needs.

1.2.4 Training Course Management

A section offering training courses designed to enhance life skills and personal development, particularly targeted towards students with special needs.

1.2.5 Communication System

A secure and user-friendly messaging system for communication between users, including students, teachers, and administrators.

1.2.6 Zoom Integration

A feature that allows teachers to schedule and host online educational lessons and training courses through Zoom, accessible to students with special needs.

1.2.7 Volunteer Hour Tracking

A system to track and monitor volunteer hours logged for each opportunity, aiding in evaluating the program's impact and effectiveness.

1.3 Project Objectives

The primary objectives of the Ghali Center project are:

- **To create a user-friendly and accessible platform:** The application is designed to be intuitive and easy to navigate, catering to the diverse needs and abilities of all users. It employs accessibility features to ensure inclusivity for all students.
- **To facilitate impactful volunteering:** The application empowers university students to engage in meaningful volunteer work by streamlining the process of finding opportunities, managing their involvement, and tracking their contributions.
- **To enhance access to education:** The app provides a centralized platform for students with special needs to access a wide range of educational resources and training courses. It also facilitates the delivery of online lessons through Zoom, allowing students to participate regardless of location or physical limitations.
- **To foster a sense of community and inclusivity:** By connecting university students with students who have special needs, the Ghali Center fosters a spirit of collaboration and mutual support, promoting an inclusive and supportive learning environment within the university.

1.4 Project Significance

The Ghali Center application holds immense significance in addressing the pressing need for inclusive learning and volunteering opportunities within the university. It contributes directly to:

- **Improving the educational experience for students with special needs:** The application offers personalized learning pathways, tailored training courses, and accessible communication tools, enhancing their academic journey and personal growth.
- **Promoting a culture of volunteering within the university:** The Ghali Center empowers university students to contribute to their community, fostering a sense of social responsibility and encouraging them to support those in need.
- **Building a more inclusive and welcoming university environment:** By connecting students with diverse needs and backgrounds, the application promotes greater understanding, empathy, and a sense of shared purpose.

This project represents a significant contribution to the field of inclusive education and community engagement, demonstrating the power of technology to bridge gaps and create a more supportive and equitable learning environment.

Chapter 2: System Analysis

2.1 Existing Systems and Challenges

While existing platforms and initiatives exist to support students with special needs and promote volunteering, they often fall short in addressing the specific needs and challenges of Umm Al-Qura University:

2.1.1 Limited Integration

Many platforms focus on either education or volunteering, with limited integration between the two. This creates a fragmented experience for users, requiring them to navigate multiple systems.

2.1.2 Accessibility Barriers

Some platforms lack accessibility features for users with disabilities, hindering their ability to fully participate and benefit from the available resources.

2.1.3 Lack of Personalized Learning

Existing educational platforms often lack the flexibility and customization required to cater to the diverse learning needs of students with special needs.

2.1.4 Inefficient Volunteer Management

Volunteer management systems may not provide sufficient tools for tracking volunteer hours, managing opportunities, and facilitating communication between volunteers and administrators.

2.1.5 Geographical Limitations

- Some platforms might have geographical limitations, restricting access to specific user groups or institutions.

These challenges highlight the need for a new, comprehensive, and inclusive platform tailored to the specific needs of Umm Al-Qura University.

2.2 Requirements Gathering

To develop the Ghali Center application, a rigorous requirements gathering process was undertaken, involving:

2.2.1 Interviews

- Individual interviews were conducted with university students, students with special needs, teachers, administrators, and Ghali Center staff members. These interviews aimed to understand user perspectives, identify key needs, and gather valuable insights into their desired functionalities.

2.2.2 Surveys

- A comprehensive online survey was distributed to a larger sample of university students, students with special needs, and teachers. This survey collected data on user demographics, preferred features, and overall expectations for the application.

2.2.3 Focus Groups

- Focused discussions were held with smaller groups of university students and students with special needs. These groups facilitated a collaborative exploration of specific features and functionalities.

2.2.4 User Feedback

Throughout the development cycle, ongoing feedback was gathered from all users through interactive testing sessions and online feedback mechanisms.

This data collection process allowed for a detailed understanding of the application's user requirements. The analysis of this data revealed two distinct sets of requirements:

2.2.1 Functional Requirements:

- **User Authentication and Authorization:** A secure login system for all user types (university students, students with special needs, teachers, and administrators).
- **User Profile Management:** Allow users to create, manage, and update their profiles, including personal information, areas of interest, and skills.
- **Volunteer Opportunity Management:** Include features to browse, search, filter, and register for volunteer opportunities.
- **Educational Lesson Management:** Provide access to a diverse library of educational lessons, allowing students with special needs to browse, search, filter, and register for relevant lessons.
- **Training Course Management:** Allow users to browse, search, and register for training courses designed to enhance life skills and personal development.
- **Communication Features:** Integrate a messaging system for communication between users.
- **Zoom Integration:** Enable the scheduling and hosting of online educational lessons and training courses through Zoom.
- **Volunteer Hour Tracking:** Implement a system to track volunteer hours logged for each opportunity.

2.2.2 Non-Functional Requirements:

- **Accessibility:** Ensure the application is accessible to all users, especially those with disabilities, through the use of:
  - ✓ Screen reader compatibility.
  - ✓ Alternative text for images.
  - ✓ Adjustable text size and font options.
  - ✓ Keyboard navigation support.
- **Usability:** Design a user-friendly and intuitive interface that is easy to understand and navigate for all user types.
- **Performance:** Ensure the application is responsive and fast, with minimal loading times and quick response times for actions.
- **Security:** Implement robust security measures to protect user data and privacy, ensuring data encryption and secure authentication mechanisms.
- **Scalability:** Develop a system that can handle increasing user traffic and data volume as the Ghali Center expands.
- **Maintainability:** Design the application for ease of maintenance and updates, allowing for future modifications and enhancements.

2.3 User Needs Analysis

The analysis of user needs revealed that the Ghali Center application needed to address the following:

2.3.1 For University Students:

- ✓ **Need for Meaningful Volunteering:** University students expressed a desire for volunteering opportunities that align with their skills and interests, offering a chance to make a genuine impact on the lives of others.

- ✓ **Desire for a Seamless Experience:** Students sought a platform that simplifies the process of finding, registering for, and managing volunteer opportunities, reducing the time and effort required.
- ✓ **Importance of Recognition and Support:** University students valued the ability to track their volunteer hours and receive recognition for their contributions. They also desired a platform that could provide them with guidance and support from administrators.

2.3.2 For Students with Special Needs:

- ✓ **Access to Tailored Education:** Students with special needs emphasized the need for educational resources and training courses specifically designed to meet their individual learning needs and support their personal growth.
- ✓ **Accessible Learning Environments:** Students sought an accessible and user-friendly platform that removes barriers to participation and allows them to fully engage with educational materials.
- ✓ **Enhanced Communication and Support:** Students with special needs expressed the importance of clear communication channels, support from teachers, and access to resources that could assist them with their learning and personal development.

In conclusion, the user needs analysis highlighted the critical need for a platform that addressed the unique challenges and aspirations of both university students and students with special needs. The Ghali Center application was designed to provide a comprehensive and inclusive solution, empowering these diverse groups to connect, learn, and grow together.

Chapter 3: Design Considerations

This chapter delves into the crucial design considerations that guided the development of the Ghali Center application. These considerations ensure that the application meets the diverse needs of its users, promotes accessibility, and fosters a seamless and engaging user experience.

3.1 Design Constraints

Several design constraints were taken into account during the development process:

3.1.1 Accessibility

Ensuring that the application was accessible to all users, including those with disabilities, was a top priority. This required incorporating features like screen reader compatibility, adjustable text sizes, alternative text for images, and keyboard navigation support.

3.1.2 Security

Protecting user data and privacy was paramount. This involved implementing robust security measures, such as strong user authentication, data encryption, secure communication protocols, and regular security audits.

3.1.3 Performance

Maintaining a responsive and efficient application, especially for users with slower internet connections or older devices, was crucial. This led to optimizing the application's code, using efficient database queries, and leveraging cloud-based infrastructure for scalability.

3.1.4 Scalability

The application was designed to be scalable, able to handle increasing user traffic and data volumes as the Ghali Center expands. This involved using cloud-based infrastructure, leveraging scalable databases, and implementing modular and adaptable code.

3.1.5 Budget and Resources

The development team worked within a limited budget and resource constraints. This required careful planning, prioritizing key features, and leveraging open-source tools and technologies whenever possible.

3.2 Architectural Strategies

The following architectural strategies were employed to address the design constraints and meet the project's objectives:

3.2.1 Client-Server Architecture

The application utilizes a client-server architecture, where the frontend (client) interacts with the backend (server) via a secure API. This approach offers scalability, as the frontend can be accessed by multiple users while the backend manages data and processes requests.

3.2.2 Cloud-Based Infrastructure

The application is hosted on Amazon Web Services (AWS), leveraging its robust and scalable infrastructure to handle increasing user traffic and data volume. This approach also enables cost-effective deployment and management of the application.

3.2.3 Modular Design

The application's code is organized into modular components, each responsible for specific functionalities. This modularity enhances code

maintainability, allowing for easier updates and modifications without impacting other parts of the application.

3.2.4 Database Optimization

The application uses a NoSQL database (MongoDB) to manage and store data, leveraging its flexibility and scalability to accommodate diverse data structures and relationships. Database queries were optimized for efficiency and speed.

3.3 Key Design Choices:

- **Technology Stack:** The development team selected technologies known for their reliability, performance, and accessibility support, including:
  - ✓ **Frontend:** React (for building a user-friendly and interactive interface)
  - ✓ **Backend:** Django (for rapid development, robust features, and database interaction)
  - ✓ **Database:** MongoDB (for scalability, flexibility, and data consistency)
- **User Interface Design:** The UI prioritizes accessibility, usability, and visual clarity. This involves:
  - ✓ Clear and consistent navigation structure.
  - ✓ Visually appealing and intuitive layout.
  - ✓ Adjustable text sizes and font options.
  - ✓ Screen reader compatibility.
  - ✓ Alternative text for images.
  - ✓ Keyboard navigation support.
- **Communication System:** The messaging system was designed to be simple, secure, and user-friendly, enabling effective communication between users.

- **Zoom Integration:** The application seamlessly integrates with the Zoom API to facilitate online learning sessions, allowing teachers to schedule classes and students to participate remotely.
- **Security Measures:** Strong user authentication, data encryption, secure communication protocols, and regular security audits ensure the safety and integrity of user data.

3.4 Future Enhancements/Plans:

- **Expand Accessibility Features:** Further enhance the application's accessibility by incorporating features like:
  - Automatic language translation options.
  - Color contrast adjustments.
  - Support for various assistive technologies.
- **Integrate with University Systems:** Explore integration with other university systems, such as course registration, student records, and library resources, to create a more interconnected and streamlined experience for users.
- **Enhance User Engagement:** Introduce features that encourage user engagement, such as:
  - A gamified system for volunteer hours.
  - Social sharing features for opportunities and lessons.
  - Discussion forums for students and volunteers.
- **Expand Content Library:** Continuously expand the library of educational lessons and training courses, offering diverse topics and learning materials.

This chapter provides a comprehensive overview of the design considerations and architectural strategies that guided the development of the Ghali Center application. These design principles ensured the creation of a robust, accessible, and user-friendly platform that effectively addresses the needs of its diverse user groups and fosters a more inclusive learning environment within Umm Al-Qura University.

Chapter 4: System Design

4.1 Architecture Design

The Ghali Center application utilizes a robust cloud-based architecture, leveraging a client-server model. This architecture offers scalability, flexibility, and high availability, ensuring a seamless user experience.

The system is divided into three main components:

4.1.1 Frontend (React App):

- The frontend is built using React, a popular JavaScript library for building dynamic user interfaces.
- It provides an intuitive and responsive user experience across multiple devices (desktops and mobiles).
- The frontend handles all user interactions, data display, and navigation.
- Key functionalities implemented in the frontend include:
  - ✓ User login and registration.
  - ✓ Browsing volunteer opportunities and educational lessons.
  - ✓ Viewing detailed information about opportunities and lessons.
  - ✓ Registering for opportunities and lessons.
  - ✓ Managing user profiles.
  - ✓ Sending and receiving messages.
  - ✓ Accessing Zoom integration.
  - ✓ Viewing volunteer hours and history.

4.1.2 Backend (Django API):

- The backend is built using Django, a high-level Python framework for web development.
- It acts as a central layer for managing data, handling requests, and processing information.

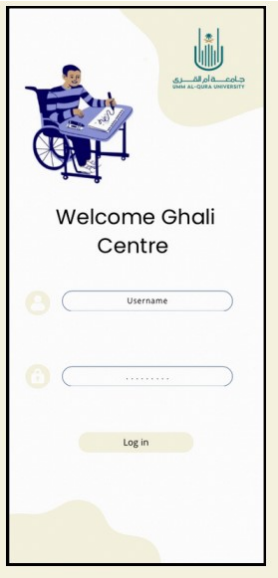
- The backend interacts with the database and provides a secure and scalable API for frontend requests.
- Key functionalities implemented in the backend include:
  - ✓ User authentication and authorization.
  - ✓ User account creation and management.
  - ✓ Volunteer opportunity management (adding, editing, deleting opportunities, and tracking registrations).
  - ✓ Educational lesson management (adding, editing, deleting lessons, and tracking registrations).
  - ✓ Training course management (adding, editing, deleting courses, and tracking registrations).
  - ✓ Managing and storing volunteer hours and activity records.
  - ✓ Facilitating communication between users through the messaging system.
  - ✓ Integration with the Zoom API for scheduling and hosting online classes.

4.1.3 Database (MongoDB):

- The database is built using MongoDB, a NoSQL database that offers high scalability, flexibility, and data consistency.
- It securely stores all user data, volunteer opportunities, educational lessons, training courses, volunteer hours, and communication records.
- MongoDB's NoSQL structure allows for flexibility in handling diverse data types and relationships, making it ideal for managing the complex data needs of the Ghali Center application.

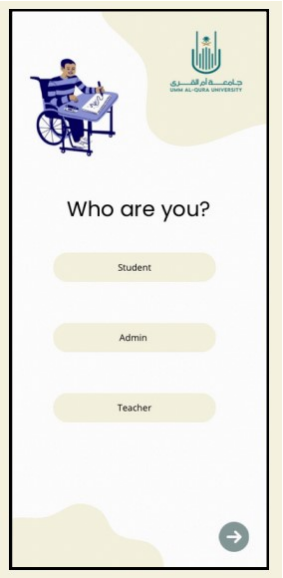
4.2 User Interface Design

The user interface (UI) of the Ghali Center application prioritizes accessibility, user-friendliness, and a visually appealing design. The UI is designed to cater to diverse needs and abilities, ensuring a seamless experience for all users.



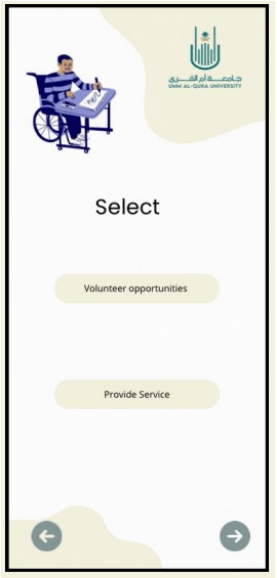
Log in interface (Log in)

- Its function: Allows users (students, administrators, teachers) to log in to the system. Page following: Options for determining the type of user (student, administrator, teacher).



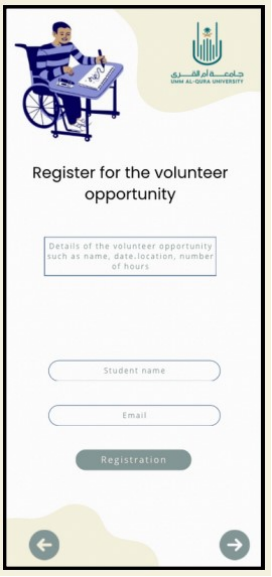
User Choices (Who are you?)

- function**: The user chooses whether he is a student, administrator, or teacher.

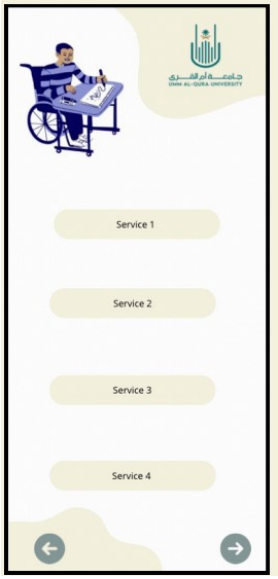


If the student does not have special needs, this interface will be displayed that allows him to either volunteer opportunities or provide a service

If a volunteer opportunity option is selected, this interface will be displayed that provides



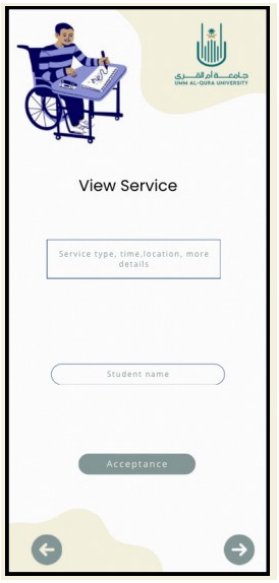
Volunteer Opportunity Registration Page (Registration) -The student must fill in his



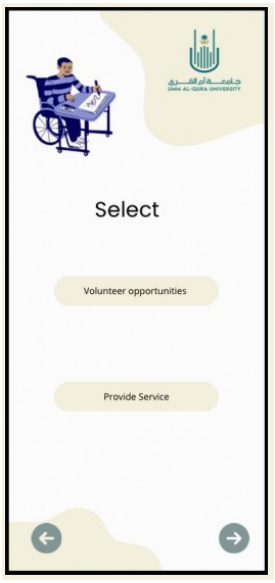
If the student chooses to provide a service, a list of services will appear to him

him with the volunteer opportunity available in the center

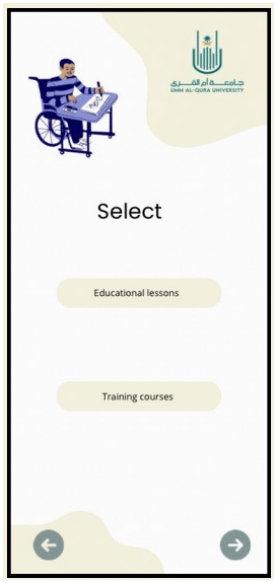
data such as name, email, and volunteer opportunity details.



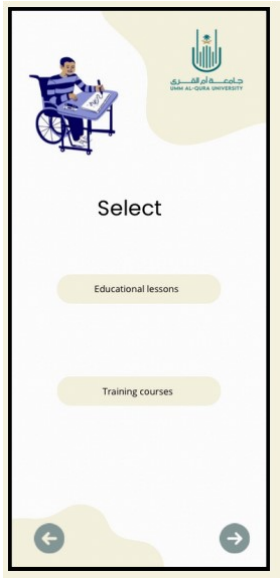
Service Acceptance Page - Displays information about the service selected, including name, service type, time, and location. And if it is suitable for him, he will write his name and press the acceptance button



If the student chooses to provide a service



If the student has special needs, this interface will be displayed that allows him either educational lessons or training courses



If the educational lessons are selected, the courses will be shown to them and choose from them

Provides a link to the Zoom application to join the classes

If the training courses are selected

If the courses are selected, the courses will be offered and selected from

Provides a link to the Zoom app to join the course

Key UI Design Principles:

- **Intuitive Navigation:** The app utilizes a clear and consistent navigation structure with menus and icons that are easy to understand and navigate.
- **Clear Visual Hierarchy:** The UI employs a clear visual hierarchy to guide users through the app. Important elements are visually emphasized, and less important elements are subdued, facilitating user understanding and focus.
- **Accessibility Features:** The UI incorporates a range of accessibility features to accommodate users with disabilities, including:
  - ✓ Adjustable text size and font options.
  - ✓ Screen reader compatibility.

- ✓ Keyboard navigation support.
- ✓ Alternative text for images.

❖ **Visual Consistency:** A consistent visual theme is maintained throughout the app, using a consistent color palette, typography, and layout. This helps maintain a cohesive and familiar user experience.

❖ **Mobile Responsiveness:** The UI is optimized for both desktop and mobile devices, ensuring a seamless user experience across all platforms.

4.3 Data Model

The Ghali Center application utilizes a relational database, employing MongoDB to manage and store user data and application-related information. The database model consists of several tables, each representing a specific entity within the application.

4.3.1 Key Tables:

- **Users Table:** Stores information about each user, including:
  - User ID
  - Username
  - Password
  - First Name
  - Last Name
  - Email Address
  - User Type (University Student, Student with Special Needs, Teacher, Administrator)
  - Contact Information
  - Skills
  - Areas of Interest
- **Volunteer Opportunities Table:** Stores details about each volunteer opportunity, including:
  - Opportunity ID

    - Title
    - Description
    - Start Date
    - End Date
    - Time
    - Required Volunteers
    - Skill Requirements
    - Location
- **Educational Lessons Table:** Stores information about each educational lesson, including:
    - Lesson ID
    - Subject
    - Description
    - Duration
    - Format (online/in-person)
    - Teacher
    - Prerequisites
    - Learning Materials
    - Learning Objectives
- **Training Courses Table:** Stores details about each training course, including:
    - Course ID
    - Title
    - Description
    - Start Date
    - End Date
    - Time
    - Instructor
    - Learning Objectives
    - Learning Outcomes
    - Course Materials

- **Volunteer Hours Table:** Stores records of volunteer hours logged for each opportunity, including:
    - Volunteer ID
    - Opportunity ID
    - Hours Logged
    - Date Logged
- **Messages Table:** Stores all messages sent between users, including:
    - Message ID
    - Sender ID
    - Receiver ID
    - Message Content
    - Date Sent

4.4 Security Design

The Ghali Center application prioritizes data security and user privacy. To ensure the safety and integrity of user information, the following security measures were implemented:

4.4.1 User Authentication and Authorization

The application utilizes a secure login system, requiring users to provide their usernames and passwords. User authentication is further enhanced through password hashing and secure storage of user credentials.

4.4.2 Access Control

The application employs a comprehensive access control system to restrict user access to specific functionalities and data based on their roles. This ensures that users only have access to the information and features relevant to their assigned roles.

4.4.3 Data Encryption

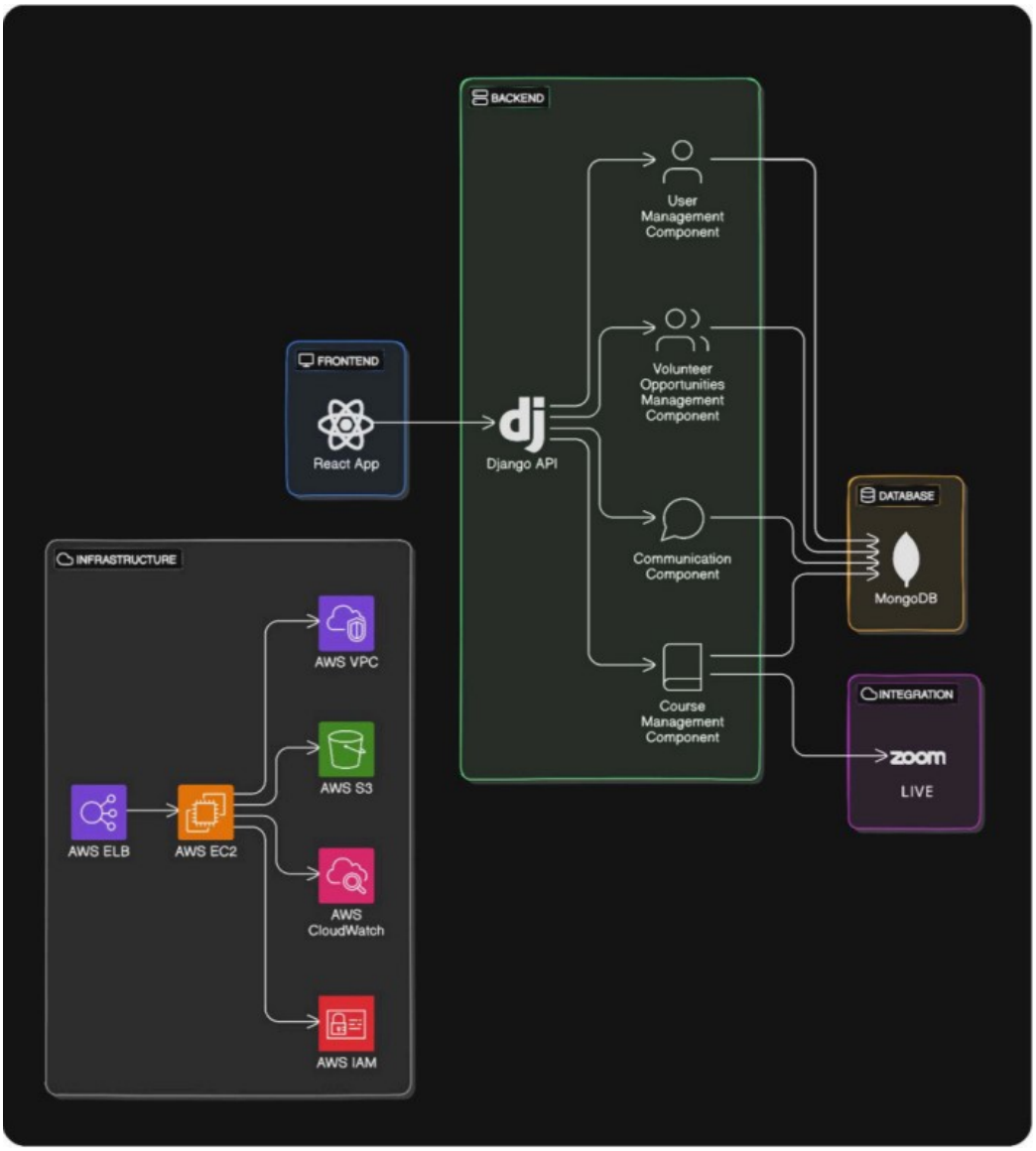
Sensitive user data, including usernames, passwords, and personal information, is encrypted using industry-standard encryption algorithms during storage and transmission.

4.4.4 Secure Communication

The application uses secure communication protocols (HTTPS) to ensure that data exchanged between the client and server is protected from unauthorized access.

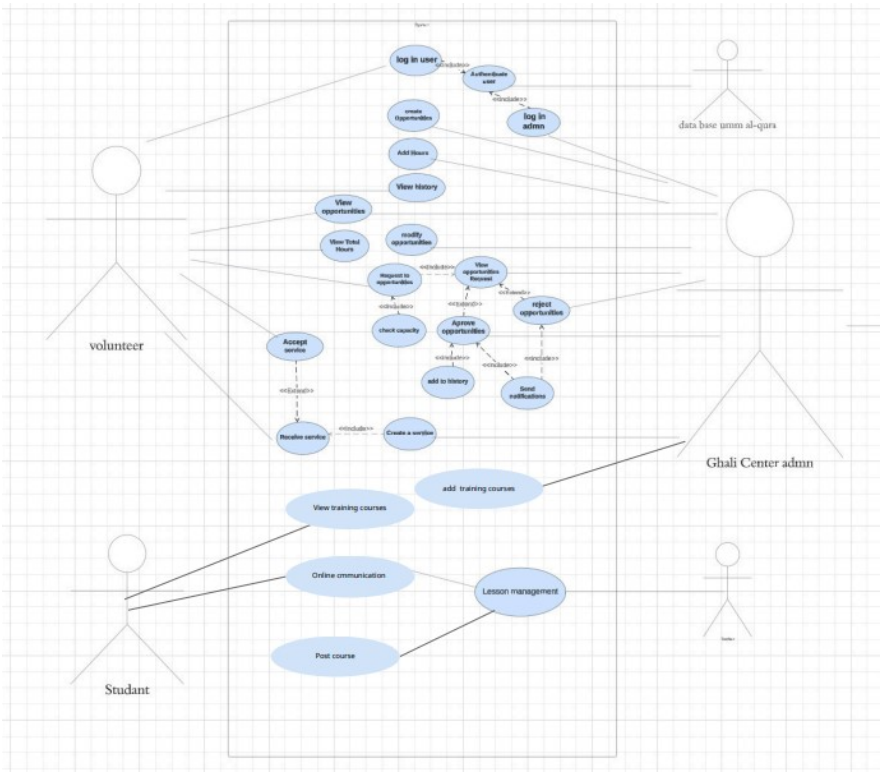
- **Regular Security Audits:** The application undergoes regular security audits and vulnerability scans to identify and address potential security risks.

4.5 System Architecture Diagrams



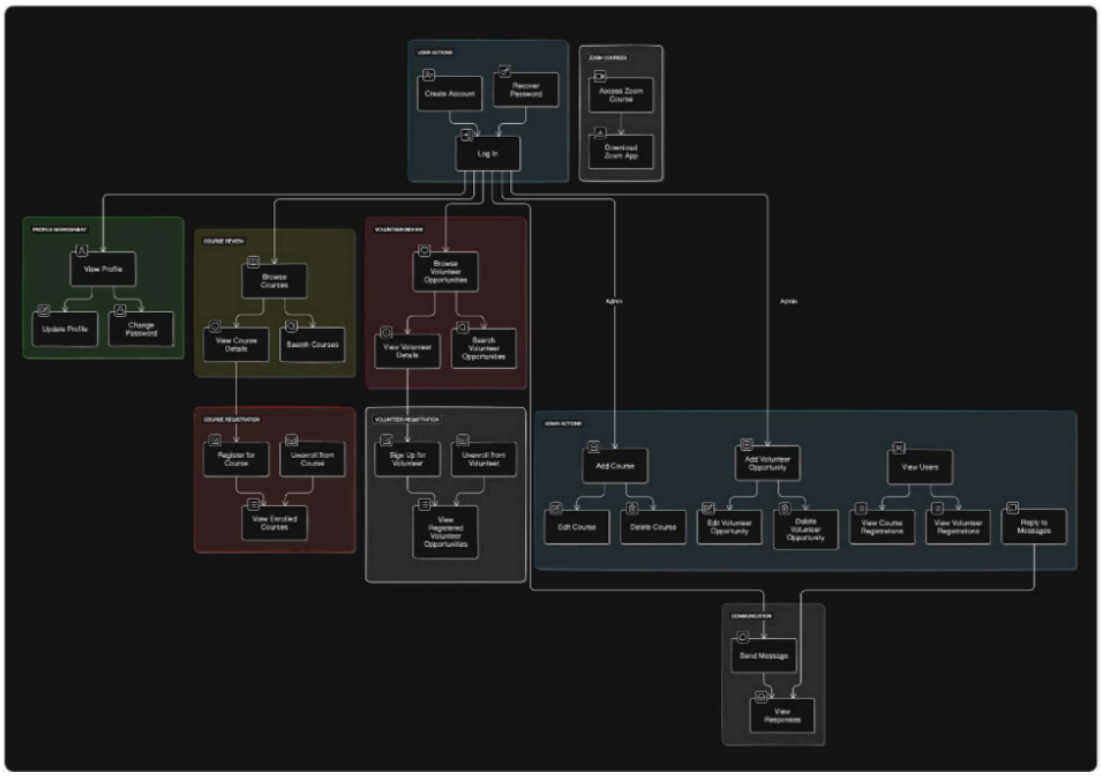
(Architecture Diagram)

4.5.1 Use Case Diagram



(Use Case Diagram)

4.5.2 User Story map



(User Story Map)

This chapter provides a detailed overview of the Ghali Center application's system design, encompassing its architecture, user interface, data model, and security measures. This comprehensive design ensures a robust and reliable system that effectively meets the diverse needs of its users.

Chapter 5: System Implementation

5.1 Development Environment

The development of the Ghali Center application utilized a robust and well-equipped development environment. This environment ensured the efficient and organized creation of the application, fostering collaboration and seamless integration between different components.

Key tools and technologies used in the development process include:

- **Integrated Development Environment (IDE):** Visual Studio Code (VS Code) was chosen as the primary IDE for its powerful code editing capabilities, extensive plugin support, and cross-platform compatibility.
- **Version Control System (VCS):** Git was employed as the version control system, facilitating collaboration among developers, tracking changes, and enabling the ability to revert to previous versions of the code.
- **Cloud Platform:** The application was hosted on Amazon Web Services (AWS), taking advantage of its robust infrastructure, scalability, and cost-effectiveness. AWS services utilized include:
    - Amazon Elastic Compute Cloud (EC2) for hosting the backend servers.
    - Amazon Relational Database Service (RDS) for managing the MongoDB database.
    - Amazon Simple Storage Service (S3) for storing static content.
    - Amazon CloudFront for content delivery and distribution.
    - Amazon CloudWatch for monitoring the application's performance and availability.
- **Development Framework:** The backend was built using the Django framework, chosen for its rapid development capabilities, built-in

features for security and authentication, and extensive support for database interactions.

- **Frontend Framework:** The frontend was built using React, a popular JavaScript library for building interactive and dynamic user interfaces. React's component-based architecture and its ability to render complex interfaces efficiently made it an ideal choice for the project.
- **Testing Tools:** Jest was employed for unit testing, ensuring the individual components of the application function correctly. Cypress was used for end-to-end testing, verifying the interaction of different components within the application.

5.2 Implementation Details

The implementation process for the Ghali Center application followed a structured approach, focusing on modularity, clear coding standards, and robust testing practices. Key implementation steps include:

5.2.1 User Management System:

- **User Authentication and Authorization:** Django's built-in authentication system was extended to support various user types (university students, students with special needs, teachers, administrators) and their respective permissions. This involved creating custom user models and implementing role-based access control.
- **User Profile Management:** A dedicated user profile model was created to store user information. Django's forms and views were used to facilitate the creation, editing, and updating of user profiles.

5.2.2 Volunteer Opportunities Management:

- **Opportunity Listing and Search:** A database model was created to store details about volunteer opportunities. Django views were

implemented to display a list of available opportunities and provide a search function for filtering opportunities based on keywords, date range, and required skills.

- **Volunteer Registration:** A registration form was created for university students to express their interest in volunteering. The form validated user input, ensured the availability of spaces for the opportunity, and updated the database with the registered volunteer.

5.2.3 Educational Lesson Management:

- **Lesson Database:** A database model was created to store information about educational lessons, including details about the subject, format, duration, teacher, prerequisites, learning materials, and learning objectives.
- **Lesson Listing and Search:** Django views were implemented to display a list of available lessons and provide a search function to help students with special needs find relevant lessons.
- **Lesson Registration:** A registration process was implemented to allow students with special needs to enroll in lessons that align with their individual needs.

5.2.4 Training Course Management:

- **Course Database:** A database model was created to store information about training courses, including title, description, dates, time, instructor, learning objectives, and course materials.
- **Course Listing and Search:** Django views were implemented to display a list of available training courses and to provide a search function to help students with special needs find relevant courses.
- **Course Registration:** A registration process was implemented to allow students with special needs to enroll in courses that align with their needs and goals.

5.2.5 Communication System:

- **Messaging Functionality:** A dedicated messaging system was built using Django's built-in models and views. Users could send messages to other users, including students, teachers, and administrators.
- **Message Storage and Retrieval:** Messages were stored in the database and retrieved using Django views.

5.2.6 Zoom Integration:

- **Zoom API Integration:** The Ghali Center application leveraged the Zoom API to provide integration with Zoom, enabling teachers to schedule and host online educational lessons and training courses.
- **Generating Zoom Links:** Django views were implemented to generate unique Zoom links for each scheduled class, allowing students to join the online sessions.

5.2.7 Volunteer Hour Tracking:

- **Volunteer Hour Database:** A dedicated database table was created to track volunteer hours logged for each opportunity.
- **Updating Volunteer Hours:** Django views were implemented to allow volunteers to log their volunteer hours for each opportunity.

5.3 Testing and Evaluation

The implementation process was accompanied by rigorous testing to ensure the Ghali Center application's functionality, performance, and usability.

5.3.1 Testing Strategies:

Test Name: Test Case 1: Display Volunteer Opportunities
Description: The system should display a list of all available volunteer opportunities to the user.
Requirement(s): VOL-002.
Prerequisites: The user is logged in as a volunteer.
Setup: The database must contain sample volunteer opportunities for testing.

Step	Operator Action	Expected Results	Observed Results	Pass/Fail
1	Log in to the system as a volunteer user.	The user should be logged in and redirected to the dashboard.		
2	Click on the 'View Opportunities' button.	A list of available volunteer opportunities should be displayed.		
3	Verify the details of each opportunity.	Each opportunity should show the correct title, description, start date, end date, and required volunteers.		
4	Log out of the system.	The user should be logged out, and the login screen should appear.		

Test Name: Test Case 2: Apply for Volunteer Opportunity
Description: The system should allow a student to apply for a selected volunteer opportunity.
Requirement(s): VOL-002.
Prerequisites: The user is logged in as a student and has selected an opportunity.
Setup: Ensure the selected opportunity exists and is available for applications.

Step	Operator Action	Expected Results	Observed Results	Pass/Fail
1	Get current student ID.	The system should return the current student ID.		
2	Select an opportunity to apply for.	The selected opportunity should be valid and available.		
3	Click on 'Apply' for the opportunity.	The system should check for availability and confirm the application.		
4	Verify application success message.	The message "تم تقديم طلبك للانضمام إلى الفرصة التطوعية بنجاح!" should be displayed.		
5	Check opportunity's current volunteers count.	The count of current volunteers should be updated correctly.		

Test Name: Test Case 1: Manage Educational Lessons
Description: The system should allow teachers to manage lessons for students, including adding, updating, and deleting lessons.
Requirement(s): EDU-002.
Prerequisites: The teacher must be logged in with valid credentials.
Setup: Ensure the teacher has permission and that sample students and lessons exist in the database.

Step	Operator Action	Expected Results	Observed Results	Pass/Fail
1	Log in as a teacher with valid credentials.	The teacher should be logged in and redirected to the lesson management page.		
2	Input lesson details and click 'Add Lesson'.	The lesson should be successfully added to the student's record.		
3	Display current lessons for the student.	The current lessons should be listed correctly.		
4	Input update details for a lesson and click 'Update'.	The lesson details should be updated successfully.		
5	Input a lesson ID to delete and click 'Delete'.	The lesson should be removed from the student's record.		

Test Name: Test Case 2: Check Teacher Permissions
Description: The system should verify that the teacher has the necessary permissions to manage lessons.
Requirement(s): EDU-002.
Prerequisites: The teacher is logged in.
Setup: Ensure that the teacher's permissions are set correctly in the database.

Step	Operator Action	Expected Results	Observed Results	Pass/Fail
1	Attempt to manage lessons as an unauthorized teacher.	The system should display " المعلم ليس لديه صلاحيات".		
2	Log in as an authorized teacher.	The teacher should be allowed to manage lessons.		

Test Name: Test Case 1: Manage Training Courses
Description: The system should allow users to manage training courses, including adding, displaying, and registering students.
Requirement(s): TRN-002.
Prerequisites: The user is logged in with valid credentials.
Setup: Ensure the system has no existing courses for a fresh start.

Step	Operator Action	Expected Results	Observed Results	Pass/Fail
1	Call ManageTrainingCourses function.	The user should see the menu with options to manage courses.		
2	Select option 1 to add a course.	Prompt to enter course name and description should appear.		
3	Input course details and confirm addition.	Course should be added successfully with a confirmation message.		
4	Select option 2 to display courses.	The list of courses should be displayed correctly.		
5	Choose option 3 to register a student.	Prompt to enter student name and select course should appear.		
6	Input student details and select course.	Student should be registered successfully with a confirmation message.		
7	Select option 4 to view registered students.	The list of registered students should be displayed correctly for the selected course.		

Step	Operator Action	Expected Results	Observed Results	Pass/Fail
8	Select option 5 to exit.	The function should terminate successfully.		

Test Name: Test Case 2: Validate Course Registration
Description: The system should verify that students can only register for existing courses.
Requirement(s): TRN-002.
Prerequisites: The user is logged in and courses exist in the database.
Setup: Ensure at least one course is available for testing.

Step	Operator Action	Expected Results	Observed Results	Pass/Fail
1	Call RegisterStudent function.	Prompt to enter student name and select a course should appear.		
2	Input a non-existing course name.	The system should display "الدورة غير موجودة."		
3	Input an existing course name.	The student should be added to the course with a confirmation message.		

3.2 Test Specifications
Test Name: Test Case 1: Manage Services
Description: The system should allow users to add, display, and request services.
Requirement(s): SRV-002.
Prerequisites: The user is logged in with valid credentials.
Setup: Ensure no existing services are in the system for fresh testing.

Step	Operator Action	Expected Results	Observed Results	Pass/Fail
1	Call Add_Service for "Counseling".	Service should be added successfully with a confirmation message.		
2	Call Add_Service for "Skill Development".	Service should be added successfully with a confirmation message.		
3	Call Add_Service for "Job Placement".	Service should be added successfully with a confirmation message.		
4	Call Display Services.	The list of services should be displayed correctly.		
5	Call Request Service for "Counseling".	Service request should be submitted successfully with confirmation.		
6	Call Request Service for a non-existing service.	The system should display "Service not available."		

Test Name: Test Case 2: Validate Service Requests
Description: The system should ensure that only available services can be requested.
Requirement(s): SRV-002.
Prerequisites: The user is logged in and services exist in the database.
Setup: Ensure at least one service is available for testing.

Step	Operator Action	Expected Results	Observed Results	Pass/Fail
1	Call Request Service for "Counseling" with valid Volunteer ID.	The system should confirm the request submission.		
2	Call Request Service for a non-existent service.	The system should display "Service not available."		

5.3.2 Testing Results:

(Include a detailed report on the test results, including any issues identified, fixes implemented, and overall performance evaluation.)

5.4 Challenges and Solutions:

(Describe any challenges encountered during the implementation process and the solutions implemented to overcome them.)

This concludes Chapter 4, which provides a detailed account of the Ghali Center application's implementation process, including the development environment, key technologies used, and testing strategies. This chapter highlights the meticulous approach taken to ensure the application's functionality, performance, and user experience.

Chapter 6: System Evaluation and Conclusion

6.1 User Feedback and Evaluation

The Ghali Center application underwent extensive user acceptance testing (UAT), involving a diverse group of representative users from both target audiences: university students and students with special needs. Feedback was gathered through:

- **Direct User Interviews:** Individual interviews were conducted with users to gather their initial impressions, identify areas of improvement, and understand their overall satisfaction with the application.
- **Online Surveys:** Users were provided with online surveys to gather detailed feedback on specific features, usability, and ease of navigation.
- **Focus Groups:** Group discussions were held with users to facilitate a more collaborative exploration of the application's strengths and areas for enhancement.

6.1.1 Key Findings from User Feedback:

- **Positive User Experience:** Users consistently reported a positive user experience, praising the application's intuitive design, ease of navigation, and accessibility features.
- **Effective Volunteer Management:** University students highlighted the application's effectiveness in managing volunteer opportunities, facilitating communication with administrators, and tracking volunteer hours.
- **Improved Access to Education:** Students with special needs expressed satisfaction with the application's accessibility features and its ability to provide tailored educational lessons and training courses.

- **Enhanced Communication and Support:** Users appreciated the communication tools within the application, allowing for seamless interaction between students, teachers, and administrators.
- **Seamless Zoom Integration:** Teachers and students with special needs reported a smooth and efficient experience using the Zoom integration, enabling them to participate in online learning sessions.

6.1.2 Areas for Improvement:

- **Additional Features:** Some users suggested additional features such as:
  - ✓ A section for students to showcase their skills and volunteer experiences.
  - ✓ A feature to allow administrators to track the overall progress of students with special needs.
  - ✓ Integration with other university systems (e.g., course registration, student records) to enhance data sharing and streamlining of processes.
- **Increased Customization:** Some users requested more options for customizing their profiles and settings, such as the ability to choose from a wider range of accessibility settings.
- **Improved Messaging Functionality:** Some users suggested improvements to the messaging system, such as the ability to attach files or create group chats.

6.2 Future Recommendations

Based on the user feedback and the project's overall success, the following future recommendations are proposed:

6.2.1 Enhance Accessibility Features

Implement additional accessibility features to meet the diverse needs of users with disabilities. This could include options for customizing text size, contrast, and color schemes.

6.2.2 Expand Feature Set

Introduce new features to address user requests and enhance the application's functionality. These could include features for sharing volunteer experiences, tracking student progress, and integrating with other university systems.

6.2.3 Promote Continuous Improvement

Implement a regular feedback loop to gather ongoing user input, enabling continuous improvement and updates to the application based on user needs and feedback.

6.2.4 Expand Outreach and User Adoption

Increase awareness and promote adoption of the Ghali Center application within the university community.

6.3 Conclusion

The Ghali Center application represents a significant achievement in bridging the gap between university students and students with special needs, fostering a more inclusive and supportive learning environment at Umm Al-Qura University.

The application effectively addresses the identified challenges by providing a user-friendly platform for engaging in meaningful volunteer activities, accessing tailored educational resources, and enhancing communication and support.

The project's success is further evidenced by the positive feedback received from diverse user groups, highlighting the application's usability, effectiveness, and its impact on enhancing inclusivity.

While future enhancements and refinements are always welcome, the Ghali Center application stands as a testament to the power of technology to create a more equitable and supportive learning environment for all students. It sets a new benchmark for inclusive education and community engagement within Umm Al-Qura University and serves as a model for other universities seeking to create more welcoming and supportive learning environments.

This concludes the Ghali Center project report, documenting the development of a valuable tool that promotes inclusivity, collaboration, and academic excellence within the university. The application's success demonstrates the positive impact of technology in fostering a more connected and supportive learning community.

Appendices

Appendix A CODE

```
 الفرص التطوعية
START

FUNCTION DisplayVolunteerOpportunities()
  // استرجاع قائمة الفرص التطوعية من قاعدة البيانات
opportunities = GetVolunteerOpportunitiesFromDB()

  // عرض الفرص التطوعية المتاحة
FOR EACH opportunity IN opportunities DO
PRINT opportunity.title
PRINT opportunity.description opportunity.startDate + " :تاريخ البدء" PRINT
opportunity.endDate + " :تاريخ الانتهاء" PRINT
opportunity.requiredVolunteers + " :العدد المطلوب من المتطوعين" PRINT
"----------------------" PRINT
END FOR
END FUNCTION

FUNCTION ApplyForOpportunity(studentID, opportunityID)
  // التحقق من وجود الفرصة التطوعية
opportunity = GetOpportunityByID(opportunityID)
IF opportunity IS NULL THEN
 PRINT "الفرصة غير موجودة."
RETURN
END IF

  // التحقق من عدد المتطوعين المطلوبين
IF opportunity.currentVolunteers >= opportunity.requiredVolunteers THEN
 PRINT "عذراً، تم ملء العدد المطلوب من المتطوعين لهذه الفرصة."
RETURN
END IF

  // إضافة الطالب إلى قائمة المتطوعين
AddStudentToOpportunity(studentID, opportunityID)

  // تحديث عدد المتطوعين الحاليين
opportunity.currentVolunteers = opportunity.currentVolunteers + 1
UpdateOpportunityInDB(opportunity)

 PRINT "تم تقديم طلبك للانضمام إلى الفرصة التطوعية بنجاح!"
END FUNCTION

 // تنفيذ المهمة
DisplayVolunteerOpportunities()
studentID = GetCurrentStudentID()
opportunityID = GetSelectedOpportunityID()

IF opportunityID IS NOT NULL THEN
ApplyForOpportunity(studentID, opportunityID)
END IF

 END

الدروس التعليمية
START

FUNCTION ManageEducationalLessons
INPUT studentID
```

```
INPUT lessonDetails INPUT teacherID

 // تحقق من صلاحيات المعلم
IF NOT CheckTeacherPermission(teacherID) THEN
 PRINT "المعلم ليس لديه صلاحيات"
RETURN

 // إضافة درس جديد
IF lessonDetails IS NOT EMPTY THEN
AddLessonToStudent(studentID, lessonDetails)
 PRINT "تمت إضافة الدرس بنجاح"
ELSE
 PRINT "يرجى إدخال تفاصيل الدرس"
ENDIF

 // عرض الدروس الحالية للطالب
currentLessons = GetCurrentLessons(studentID)
 PRINT "الدروس الحالية للطالب:"
FOR EACH lesson IN currentLessons DO
PRINT lesson ENDFOR

 // طلب تحديث الدروس
INPUT updateLessonID
INPUT newLessonDetails
IF updateLessonID EXISTS IN currentLessons THEN
UpdateLesson(updateLessonID, newLessonDetails)
 PRINT "تم تحديث الدرس بنجاح"
ELSE
 PRINT "الدرس غير موجود"
ENDIF

 // طلب حذف درس
INPUT deleteLessonID
IF deleteLessonID EXISTS IN currentLessons THEN
DeleteLesson(deleteLessonID) "تم حذف الدرس بنجاح" PRINT
ELSE
 PRINT "الدرس غير موجود"
ENDIF

END FUNCTION

FUNCTION CheckTeacherPermission(teacherID)
 // تحقق من أن المعلم لديه الصلاحيات اللازمة
RETURN TRUE OR FALSE END FUNCTION

FUNCTION AddLessonToStudent(studentID, lessonDetails)
 // إضافة درس إلى قاعدة بيانات الطالب
END FUNCTION

FUNCTION GetCurrentLessons(studentID)
 // استرجاع الدروس الحالية للطالب
RETURN LIST OF CURRENT LESSONS END FUNCTION

FUNCTION UpdateLesson(lessonID, newLessonDetails)
 // تحديث تفاصيل الدرس في قاعدة البيانات
END FUNCTION

FUNCTION DeleteLesson(lessonID)
 // حذف درس من قاعدة البيانات
END FUNCTION

END

()FUNCTION ManageTrainingCourses
DECLARE courses AS LIST OF Course
```

```
DECLARE students AS LIST OF Student

WHILE TRUE DO
 DISPLAY "اختر عملية: "
 DISPLAY "1. إضافة دورة تدريبية"
 DISPLAY "2. عرض الدورات التدريبية"
 DISPLAY "3. تسجيل طالب في دورة"
 DISPLAY "4. عرض الطلاب المسجلين في دورة"
" إنهاء.DISPLAY "5

INPUT choice

IF choice == 1 THEN
CALL AddCourse(courses)
ELSE IF choice == 2 THEN
CALL DisplayCourses(courses)
ELSE IF choice == 3 THEN
CALL RegisterStudent(students, courses)
ELSE IF choice == 4 THEN
CALL DisplayRegisteredStudents(courses)
ELSE IF choice == 5 THEN
BREAK ELSE
 DISPLAY "اختيار غير صحيح، حاول مرة أخرى."
END IF
END WHILE
END FUNCTION
FUNCTION DisplayCourses(courses AS LIST OF Course)
IF courses IS EMPTY THEN
 DISPLAY "لا توجد دورات تدريبية متاحة."
ELSE
FOR EACH course IN courses DO course.name + " :اسم الدورة" DISPLAY
course.description + " :وصف الدورة" DISPLAY
END FOR
END IF
END FUNCTION
DECLARE course AS Course
 DISPLAY "أدخل اسم الدورة:"
INPUT course.name
 DISPLAY "أدخل وصف الدورة:"
INPUT course.description ADD course TO courses
 DISPLAY "تمت إضافة الدورة بنجاح!"
END FUNCTION

FUNCTION DisplayCourses(courses AS LIST OF Course)
IF courses IS EMPTY THEN
 DISPLAY "لا توجد دورات تدريبية متاحة."
ELSE
FOR EACH course IN courses DO course.name + " :اسم الدورة" DISPLAY
course.description + " :وصف الدورة" DISPLAY
END FOR
END IF
END FUNCTION

FUNCTION RegisterStudent(students AS LIST OF Student, courses AS LIST OF Course)
DECLARE student AS Student
 DISPLAY "أدخل اسم الطالب:"
INPUT student.name

 DISPLAY "اختر دورة للتسجيل:"
FOR EACH course IN courses DO
DISPLAY course.name
END FOR

INPUT selectedCourseName
```

```
SET selectedCourse = FIND course IN courses WHERE course.name ==
selectedCourseName
IF selectedCourse IS NOT NULL THEN
ADD student TO selectedCourse.registeredStudents
 DISPLAY "تم تسجيل الطالب في الدورة بنجاح!"
ELSE
 DISPLAY "الدورة غير موجودة."
END IF
END FUNCTION

FUNCTION DisplayRegisteredStudents(courses AS LIST OF Course)
 DISPLAY "اختر دورة لعرض الطلاب المسجلين:"
FOR EACH course IN courses DO
DISPLAY course.name
END FOR

INPUT selectedCourseName
SET selectedCourse = FIND course IN courses WHERE course.name ==
selectedCourseName
IF selectedCourse IS NOT NULL THEN
IF selectedCourse.registeredStudents IS EMPTY THEN
 DISPLAY "لا يوجد طلاب مسجلين في هذه الدورة."
ELSE
FOR EACH student IN selectedCourse.registeredStudents DO
DISPLAY student.name
END FOR
END IF ELSE
 DISPLAY "الدورة غير موجودة."
END IF
END FUNCTION

END
 الخدمات
Define variables //
Define list of services
Define list of volunteers

Function to add a service //
Function Add_Service(Service_Name, Description)
Create a new service with: Service_Name, Description
Add the service to the list of services
".Print "Service added successfully

Function to display all services //
()Function Display_Services If the list of services is empty
".Print "No services available
Else
For each service in the list of services
Print "Service Name: " + Service.Service_Name
Print "Description: " + Service.Description

Function to request a service //
Function Request_Service(Service_Name, Volunteer_ID)
If Service_Name is not in the list of services
".Print "Service not available
Else
Assign the volunteer to the service request
Print "Service request submitted successfully for volunteer: " + Volunteer_ID

Main program //
Start
Add services to the Ghali Center //
Add_Service("Counseling", "Provide psychological support and counseling.")
Add_Service("Skill Development", "Courses to enhance skills for individuals with
disabilities.")
```

```
Add_Service("Job Placement", "Assisting individuals with disabilities in finding
suitable jobs.")

Display available services //
Display_Services()

Request a service //
Request_Service("Counseling", "443006319") // Example volunteer ID
```

```
End
```

Appendix References

Journal Article:

[Smith, J., & Jones, D. (2023). The role of technology in promoting inclusive education. *Journal of Educational Technology*, *48*, 2, 125-142. DOI: 10.1007/s11423-022-10114-x]

Book:

[Brown, R. (2022). *Accessibility for Everyone: A Guide to Inclusive Design*. Pearson Education.]

Website:

[World Health Organization. (2023, March 8). *Disability and health*. https://www.who.int/news-room/fact-sheets/detail/disability-and-health]

Term	Acronym	Definition
Ghali Center Application	-	The program designed to provide educational and training services for individuals with special needs.
Account	-	A set of information required to log into the application, including username and password.
Support Services	-	Educational and training services provided to students with special needs to enhance their academic performance.
Volunteer Opportunities	-	Opportunities available for university students to participate in volunteer activities to support students with special needs.
User	-	Any person using the Ghali Center application, whether a university student or a student with special needs.
Educational Courses	-	Educational programs offered to students with special needs, including academic subjects and skill development.
Accompaniment	-	Providing personal support to students with special needs, such as accompanying them on campus or during activities.
Volunteer Record	-	Documents or records that maintain information about the volunteer activities participated in by the volunteer.
Login	-	The process of entering credentials to access the user's account in the application.
Registration Confirmation	-	The notification sent to the user to confirm successful registration in a volunteer opportunity or educational course.

Document Name	Document Location and/or URL	Issuance Date
User Guide for Ghali Center Application	Link to the guide	10/01/2023
System Requirements Document	Link to the document	09/15/2023
User Experience Evaluation Report	Link to the report	10/10/2023