Faculty of Computers & Artificial Intelligence

Benha University

Sign Language Translator

A senior project submitted in partial fulfillment of the requirements for the degree of Bachelor of Computers and Artificial Intelligence.

Computer Science Departement,

Project Team

1- Ahmed Mohamed Ahmed Esmail

2- Ahmed Mohamed Shaban Abas

3- Eslam Hamdy Ragab Abd Allah

4- Abdul-Rhman Rashwan Elsaid

5- Omar Ashraf Abd El-Qader Hegab

6- Ghaidaa Hisham Abd El-Monem Abd El-Aziz

7- Sherif Alaa Abd El-Monem Mohamed

8- Mustafa Khaled Abdo Mohamed

Under Supervision of

Dr. Rasha Orban

Benha, June 2024

DECLARATION

We hereby certify that this material, which we now submit for assessment on the program of study leading to the award of Bachelor of Computers and Artificial Intelligence in *(Bachelor's)* is entirely our own work, that we have exercised reasonable care to ensure that the work is original, and does not to the best of our knowledge breach any law of copyright, and has not been taken from the work of others save and to the extent that such work has been cited and acknowledged within the text of our work.

Signed: _____

Date: Monday, 24 06 2024.

Table of Contents

TABLE OF FIGURES

1. Introduction and Background

1.1 Introduction

The Sign Language Translator project represents a significant step forward in addressing the communication challenges faced by the deaf community. By leveraging cutting-edge artificial intelligence technology, this initiative aims to provide a practical and user-friendly solution for seamless communication between individuals with hearing impairments and the broader society.

One key aspect of this project is its use of an AI-driven model, which excels in analyzing textual inputs with precision and translating them into sign language accurately and efficiently. This feature ensures that the communication is not only accessible but also reliable for the users.

Moreover, the integration of the camera function into the application opens up new possibilities by allowing the conversion of sign language into text. This innovative approach facilitates interaction with the surrounding environment, enabling more dynamic and versatile communication for individuals with hearing impairments.

The user interface is another crucial element of the Sign Language Translator project. Designed to be simple and intuitive, it ensures that users can navigate the application with ease. The customizable settings further enhance the user experience, catering to individual needs and preferences.

By focusing on enhancing communication for the deaf community, this project contributes to fostering a sense of inclusivity and improving the overall quality of daily life for individuals with hearing impairments. The goal is to make communication a universal right, transcending barriers and creating a more connected and inclusive world.

This transformative journey invites individuals to join the effort, recognizing the importance of making communication accessible to everyone and supporting the creation of a more inclusive society. Through collaborative efforts, the Sign Language Translator project aims to create positive change and make strides towards a world where effective communication is indeed a universal right.

1.2 Problem Definition:

1.2.1 Description of the Problem:

The project aims to develop a Sign Language Translation that facilitates communication between individuals who use sign language and those who do not. The goal is to bridge the communication gap and provide a seamless means for sign language users to interact with the broader community.

1.2.2 Scope of the Problem:

The scope includes the development of a real-time sign language translation system that can interpret and convert sign language gestures into spoken or written language.

1.2.3 Objectives and Goals:

- Enable effective communication for deaf or hard of hearing individuals in various settings, including educational institutions, workplaces, and social environments.
- Develop a user-friendly interface that accommodates both sign language users and those unfamiliar with sign language.
- Provide accurate and timely translation of sign language gestures to spoken or written language.

1.2.4 Constraints and Limitations:

- The system must operate in real-time to ensure effective and natural communication.
- Consideration of different dialects and variations within the chosen sign language.
- Accessibility requirements, ensuring the system is usable by individuals with varying levels of technological proficiency.

1.2.5 Stakeholders and Users:

- Primary stakeholders include those deaf or hard of hearing.
- Secondary stakeholders include educators, employers, and community members interested in fostering inclusive communication.
- Developers and Project Management Team: stakeholders responsible for designing, developing, and implementing the communication accessibility project. Their involvement is crucial for the success of the initiative.

- Potential Investors or Funding Sources: Stakeholders who provide financial support for the project. Their involvement is essential to ensure the availability of resources needed for development, implementation, and sustainability.
- User Experience and Accessibility Experts: Stakeholders who bring expertise in user experience and accessibility. Their role is important in advising and guiding the development team to create solutions that meet the specific needs of the deaf and hard-of-hearing community.

1.2.6 Functional Requirements:

- Capture and interpret sign language gestures through computer vision or similar technology.
- Translate interpreted gestures into spoken language or text.
- Provide a user interface that is intuitive for both sign language users and non-sign language users.

1.2.7 Non-functional Requirements:

- Ensure high accuracy in gesture recognition to minimize misinterpretations.
- Minimize latency to achieve real-time communication.
- Implement security measures to protect user privacy and data.

1.2.8 Assumptions and Risks:

- Assumption: Adequate training data for the chosen sign language will be available.
- Risk: Variability in individual signing styles may pose challenges for accurate interpretation.

1.2.9 Success Criteria:

Positive feedback from end-users regarding the system's usability and effectiveness.

1.3 Problem Solution:

1.3.1 Proposed Solution:

- Develop a Sign Language Translation that employs a combination of computer vision, machine learning, and natural language processing techniques to accurately interpret sign language gestures in real-time. The system will then translate these gestures into spoken language or text, providing a seamless means of communication for individuals who use sign language.

1.3.2 Technical Approach:

- Computer Vision: Utilize computer vision algorithms to capture and analyze sign language gestures through video input.
- Machine Learning: Implement machine learning models trained on diverse datasets to enhance the system's accuracy and adaptability to individual signing styles.
- Natural Language Processing: Employ natural language processing algorithms to convert interpreted gestures into spoken language or written text.

1.3.3 System Architecture:

- The system will consist of a camera to capture sign language gestures.
- Computer vision algorithms will preprocess and interpret the gestures.
- Machine learning models will refine the interpretation based on individual user patterns.
- The translated output will be presented through a user-friendly interface, including spoken language output and/or on-screen text.

1.3.4 User Interface Design:

- Develop an intuitive interface that accommodates both sign language users and non-sign language users.
- Provide options for customization, allowing users to adapt the system to their individual preferences.

1.3.5 Testing and Validation:

- Conduct rigorous testing using diverse datasets to train and evaluate the machine learning models.
- Perform user testing with individuals who use sign language to validate the system's accuracy and usability.
- Iterate on the system based on user feedback to continuously improve performance.

1.3.6 Accessibility and Inclusivity:

- Ensure the system is accessible to individuals with varying levels of technological proficiency.
- Implement features that consider different dialects and variations within the chosen sign language.
- Prioritize user privacy and data security in the design and implementation.

1.3.7 Deployment and Scalability:

- Develop the system with scalability in mind to accommodate potential expansion to other sign languages.
- Plan for easy deployment in diverse settings, including educational institutions, workplaces, and community spaces.

1.3.8 Monitoring and Maintenance:

- Implement monitoring tools to track system performance and user feedback.
- Establish a maintenance plan to address issues, update models, and incorporate improvements over time.

1.3.9 Expected Outcomes:

- Positive feedback from end-users regarding the system's usability and effectiveness.
- Increased accessibility and inclusivity in communication for individuals who use sign language

1.4 Scope:

1.4.1 Objective:

- Develop an advanced Sign Language Translator application that enables seamless communication between individuals with hearing impairments and the broader community.
- Provide a user-friendly interface for text-to-sign language and sign language-to-text translations, supporting inclusivity and accessibility.

1.4.2 Location:

- The project will be designed as a mobile application and a web page, ensuring accessibility across various devices and platforms.

1.4.3 Budget:

- Allocate resources for the development, testing, and implementation phases.
- Account for potential expenses related to AI model development, server hosting, and platform compatibility.

1.4.4 Milestones:

- Define key milestones, including the completion of AI model development, implementation of translation features, user interface design, testing phases, and the launch of the application.

1.4.5 Projects Parts/Components:

- Text-to-Sign Language Translation Module.
- Sign Language-to-Text Translation Module.
- User Interface Design.
- Camera Integration Module.
- Cross-Platform Compatibility Module.
- Security and Privacy Features.
- Testing and Quality Assurance.

1.4.6 Project Description:

- The project aims to create a state-of-the-art Sign Language Translator application, leveraging AI for accurate translation between text and sign language. The user-friendly interface and camera integration facilitate dynamic and inclusive communication.

1.4.7 Type of Works (BOQ - Bill of Quantities):

- Specify the quantity and types of resources needed for AI model development, server hosting, software development, and testing.

1.4.8 Roles and Responsibilities for Different Parties:

- Developers: Responsible for AI model development, software coding, and application testing.
- Project Management Team: Oversee project timelines, milestones, and resource allocation.
- User Experience Experts: Ensure the design meets accessibility standards and provides an optimal user experience.
- Investors/Funding Sources: Provide financial support for the project.
- Testing Team: Conduct thorough testing to identify and resolve any issues before the application launch.

By addressing these components within the project scope, you establish a comprehensive framework that guides the development and implementation of the Sign Language Translator application.

2. System Design

2.1 Use Case Diagram

The Sign Language Translator Software accommodates four distinct user roles Admin, Guest (Gest), Normal User, and Deaf User. This platform encompasses 15 essential functions to enhance communication between sign language speakers and non-sign language speakers. Below is an elaboration of the primary components and functions captured in the use case diagram:

2.1.1 User Roles:

- **Admin:** System administrator responsible for managing courses and user accounts.
- **Guest (Gest):** Unregistered users exploring the system without authentication.
- **Normal User:** Authenticated users interested in learning sign language and utilizing translation features.
- **Deaf User:** Users who are deaf and rely on sign language for communication.

2.1.2 Use Cases:

- **Login:** Allows users to securely log into their accounts.
- **Registration:** Enables new users to create accounts within the system.
- **Reset Password:** Provides a mechanism for users to reset their passwords securely.
- **Translate Sign to Text:** Converts sign language gestures into textual information for non- sign language speakers.
- **Translate Text to Sign:** Transforms textual input into sign language gestures for sign language speakers.
- **Education:** Offers educational resources and tutorials for users to learn sign language.
- **Add Course:** Allows administrators to add new sign language courses to the system.
- **Edit Course:** Permits administrators to modify existing course content.
- **Remove Course:** Enables administrators to delete courses from the system.
- **Chat bot:** provides a chat bot that has predefined questions and answers and if the user's question isn't answered the bot will forward user to an admin's email.
- **View Progress in Course:** Allows users to track their progress within enrolled courses.

- **View Courses:** Displays a list of available sign language courses.
- **View User Information:** Provides users with access to their account information.
- **Delete Account:** Allows users to permanently delete their accounts.

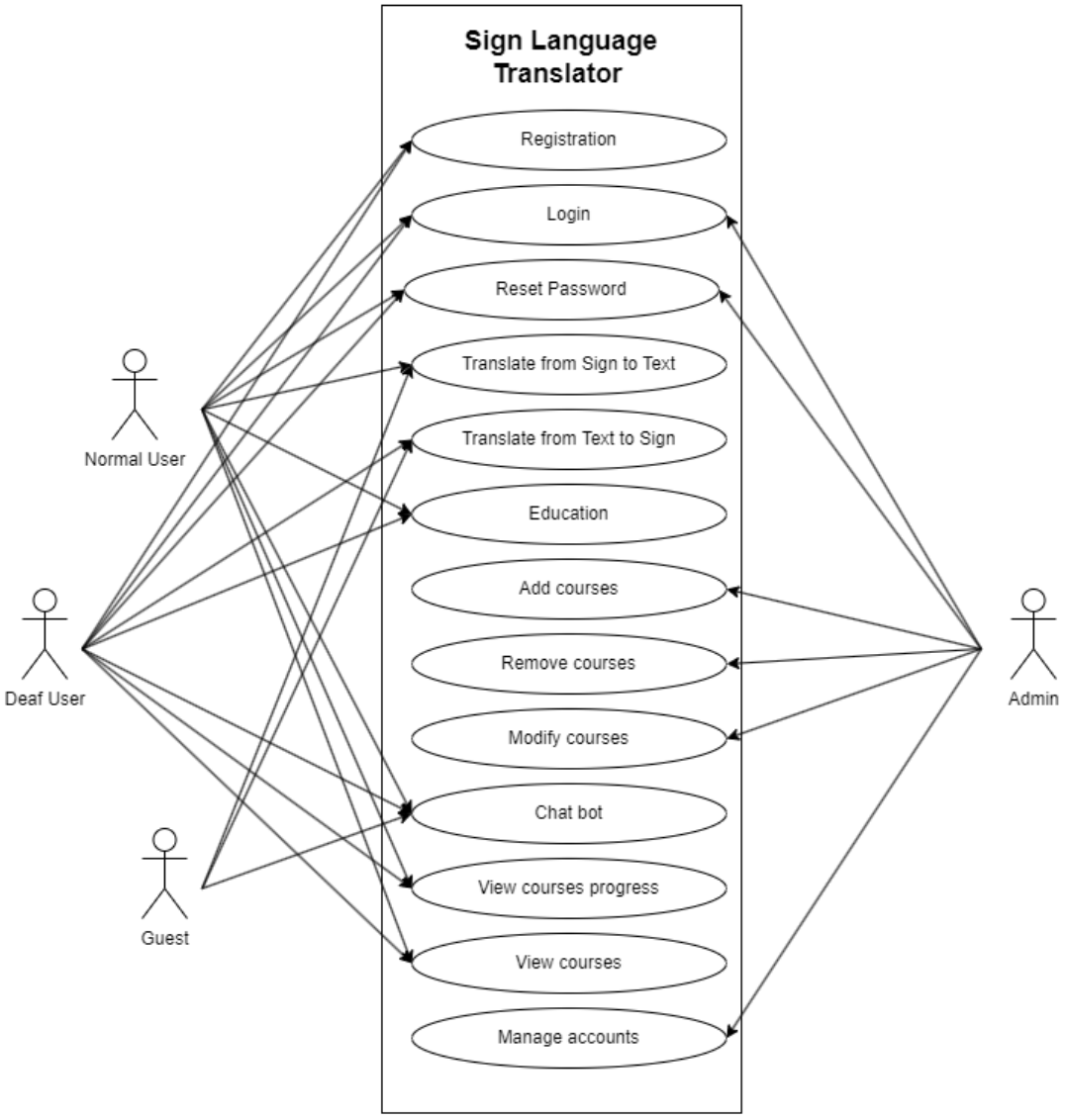


Figure 1 Use Case

Use Case ID	1
.0Use Case name	Registration
Actors	Normal user, Deaf user, Guest.
Preconditions	None.
Normal Flow	1. User clicks on Sing Up button. 2. User fills in his/her information. 3. Users receive confirmation E-mail. 4. The system displays the Sign Translation page.
Post conditions	1. Open Home page 2. The user's details are stored in the database.
Alternative Flow	The information already exists.

Use Case ID	2
Use Case name	Login
Actors	Admin, Normal User, Deaf User.
Preconditions	Admin, Normal User and Deaf User need to complete the registration process.
Normal Flow	1. Insert Username. 2. Insert Password. 3. Login.
Post conditions	Login Successfully.
Alternative Flow	- Insert the Incorrect Username. - Insert the Incorrect Password.

Use Case ID	3
Use Case name	Reset Password
Actors	Admin, Normal User, Deaf User.
Preconditions	Admin, Normal User and Deaf User Must be Registered.
Normal Flow	1. Insert username or email. 2. A reset password link will be sent to user email if it is correct and exists in database. 3. Enter a new password. 4. Confirm password.
Post conditions	Password has been Reset.
Alternative Flow	- Insert Incorrect Email Address or username. - Insert the Same Old Password.

Use Case ID	4
Use Case name	Translate From Sign to Text
Actors	Normal User, Guest
Preconditions	Normal User or Guest must enter the sign which he wants to translate to text
Normal Flow	1. Enter the sign you want to translate to text 2. Return text
Post conditions	Translate from sign to text successful

Use Case ID	5
Use Case name	Translate From Text to Sign
Actors	Deaf User, Guest
Preconditions	The Deaf User or the Guest must enter the text which he wants to translate to sign
Normal Flow	1. Enter the text you want to translate to sign. 2. Return sign
Post conditions	Translate from text to sign successful
Alternative Flow	• did not enter any text or enter a not understood text

Use Case ID	6
Use Case name	Education
Actors	Normal user, Deaf user
Preconditions	Users must log in first to access the educational platform. Course content is available for the selected course.
Normal Flow	1. User selects a course. 2. User views course content and interacts with it.
Post conditions	The user successfully accesses and interacts with the course content.
Alternative Flow	In case of technical issues, the user is redirected to a support page.

Use Case ID	7
Use Case name	Add Course
Actors	Admin
Preconditions	Admin must log in first to access the course management system.
Normal Flow	1. Admin navigates to the "Add Course" section. 2. Admin provides course details. 3. Admin click on “Add Course” button. 4. The system verifies and stores the course information.
Post conditions	The new course has been successfully added to the system.
Alternative Flow	If there are any validation errors (missing information or duplicate course name), the system prompts the admin to correct the errors and resubmit.

Use Case ID	8
Use Case name	Remove courses
Actors	Admin
Preconditions	The admin must enter the course code to remove it
Normal Flow	1. Admin login. 2. Admin click on delete course tap 3. Confirm the admin identity 4. Admin enter course code 5. Admin click on delete button
Post conditions	Course deletes successful
Alternative Flow	have not entered the course code or entered the course code incorrectly

Use Case ID	9
Use Case name	Modify courses
Actors	Admin
Preconditions	The admin must enter the course code to modify it
Normal Flow	1. Admin login. 2. Admin click on modify course tap. 3. Confirm the admin identity. 4. Admin enter course code. 5. Admin click on modify button.
Post conditions	Courses modify successful
Alternative Flow	have not entered the course code or entered the course code incorrectly

Use Case ID	10
Use Case name	Chat Bot
Actors	Normal User, Deaf User, Guest.
Preconditions	none
Normal Flow	1. Go to the chat bot section. 2. Choose one of the available question. 3. The bot should return the answer. 4. If the user want's more help the bot will provide the user with admin email to ask him directly
Post conditions	- User gets answer for his question. - User gets admin email to contact him directly
Alternative Flow	- Faild to access chat bot

Use Case ID	11
Use Case name	progress in courses
Actors	Normal user, deaf user
Preconditions	View courses and learn through them in the system
Normal Flow	1. view courses 2. learning 3. show progress in courses
Post conditions	show progress in courses
Alternative Flow	The user hasn't Enrolled in any courses yet

Use Case ID	12
Use Case name	View courses
Actors	Normal user, deaf user
Preconditions	Normal User and Deaf User Must do Registration and log in successfully. Admin must add course. Normal User, Deaf User must progress in courses
Normal Flow	1. User login 2. view courses
Post conditions	Access the courses through which we Learn
Alternative Flow	no courses exist

Use Case ID	13
Use Case name	Manage Accounts
Actors	Admin
Preconditions	User Must be Logged in As Admin
Normal Flow	1. Admin login. 2. Click on the Manage Accounts button. 3. Search for the profile he wants to view. 4. Display User profile 5. - If the admin wants to edit account, should click on edit account - or he can delete account by pressing delete account - to delete or edit account the admin must confirm his password
Post conditions	View, edit or delete user Account
Alternative Flow	- The user does not exist. - Wrong password

2.2 Activity Diagrams

The Activity Diagram explains the operations that are carried out on each process, and I see the validation, The powers, and how will the system work, In the case of success, one thing is done, and in case of failure, another thing is done.

2.2.1 Registration:

Users are required to provide relevant information such as personal details and contact information.

Once the necessary information is entered,

users will typically need to create a username and password, followed by verification steps to ensure the accuracy of the provided details.

After successfully completing these steps,

users gain access to the platform or system with their unique credentials.

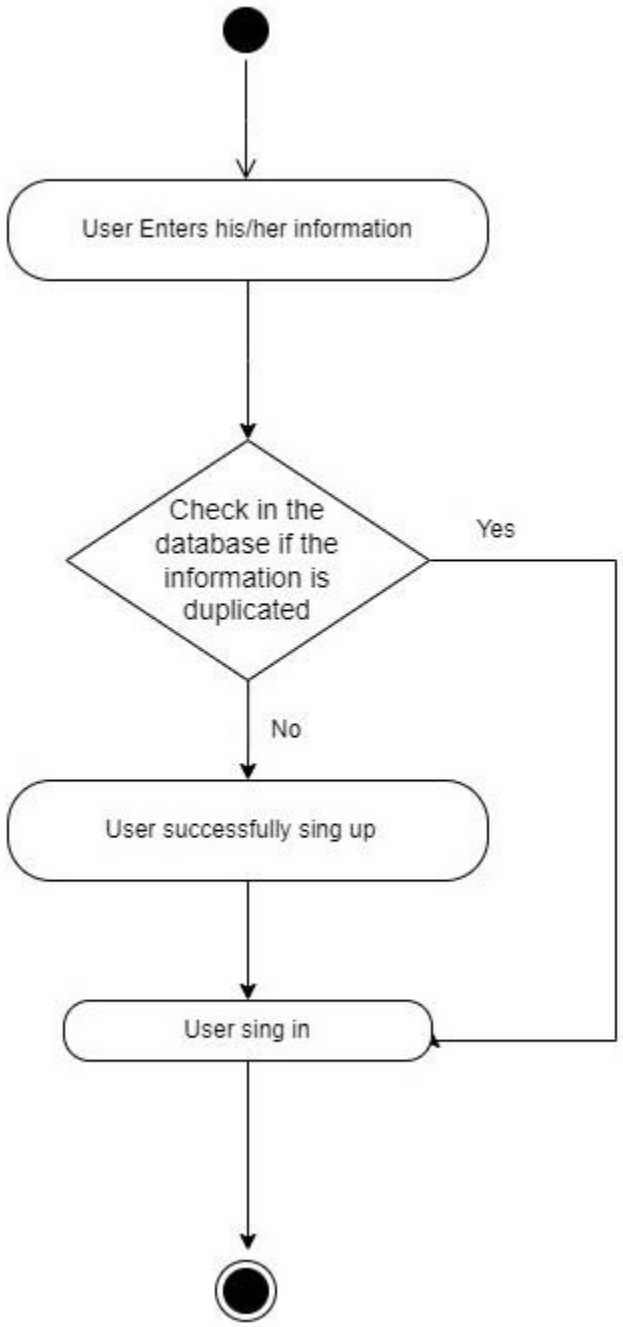


Figure 2 Registration Activity

2.2.2 Login:

The login process typically follows the registration process and involves users accessing the Application or Website using their previously created credentials.

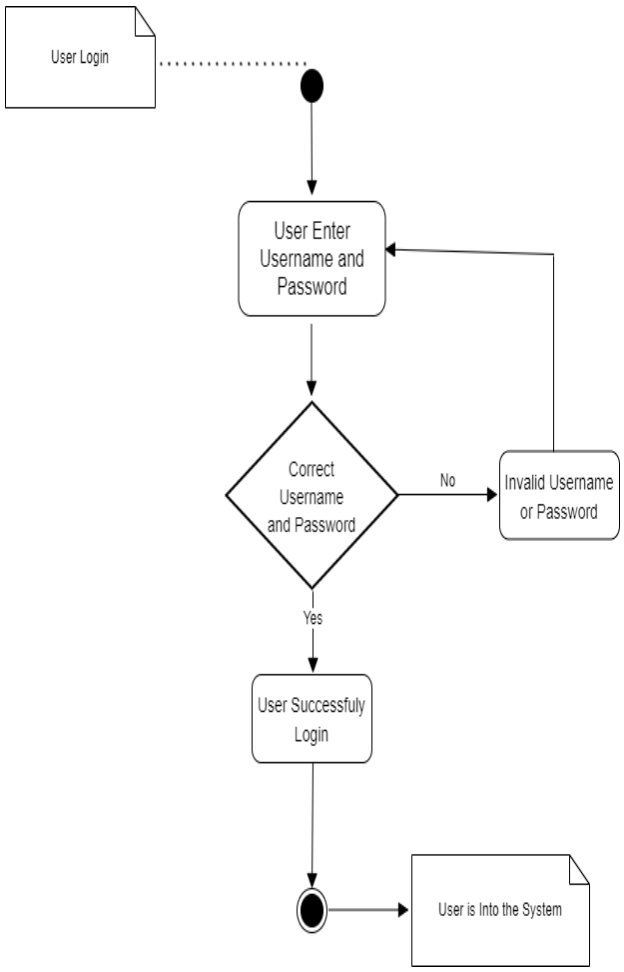


Figure 3 Login Activity

2.2.3 Reset Password:

The activity diagram visually represents the flow of activities, decisions, and interactions between the user and the system during the password reset process.

It starts with the user input his username or email, if the email that user entered is correct and exists in database,

A reset password Link will be sent for this email.

After this user enter a new password, confirm it and if it's valid, the password will be reset successfully.

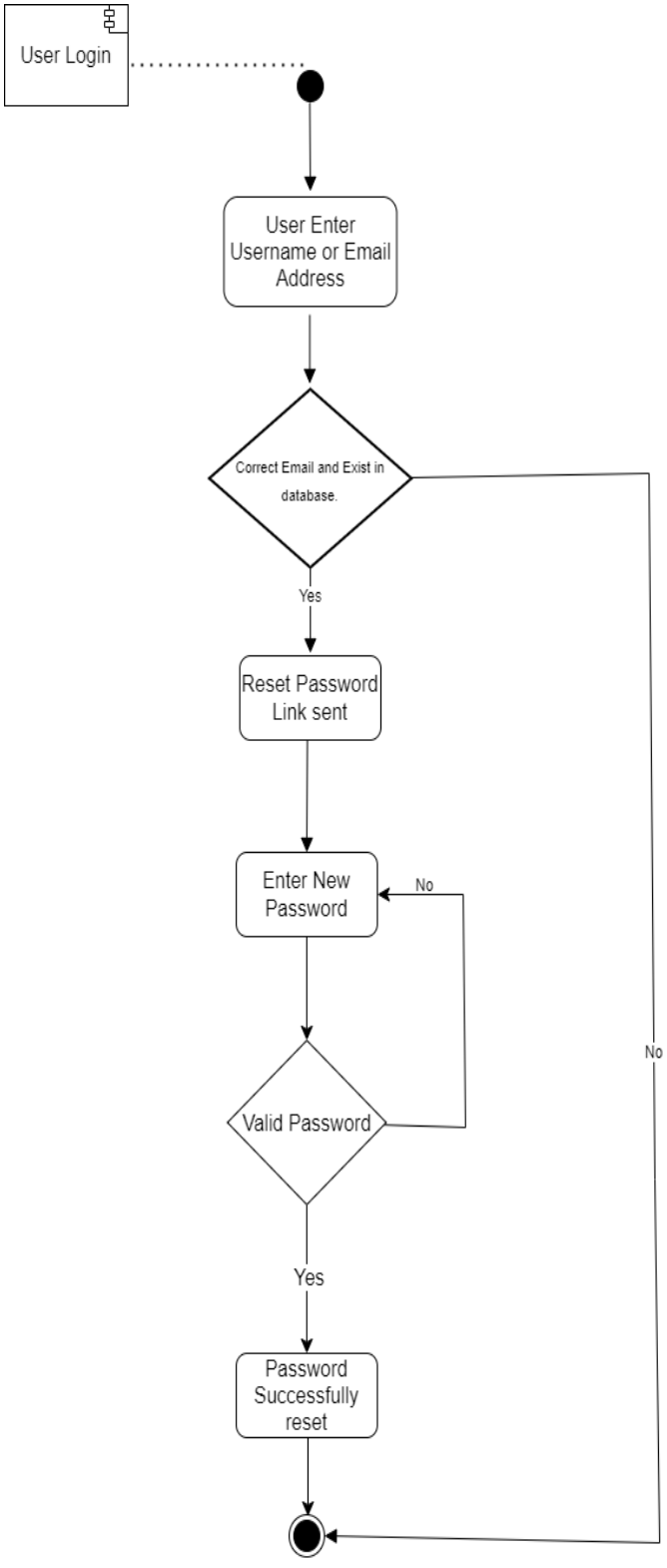


Figure 4 Reset Password Activity

2.2.4 Translate from sign to text:

As a user and he wants to translate from sign to text first he opens the translation page and begin to translate

- if as usual he does not enter any sign or a not understood one nothing will be translated so he must enter an understood sign to translate successfully

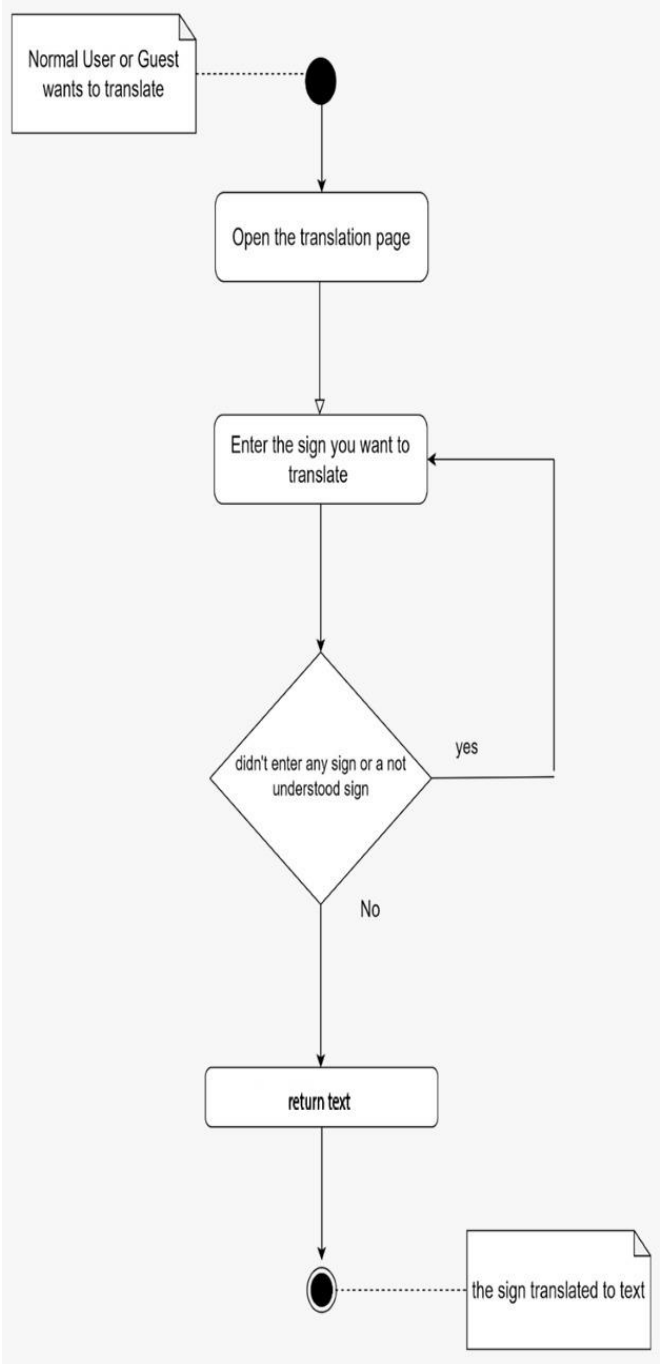


Figure 5 Translate from sign to text Activity

2.2.5 Translate from text to sign:

As a user and he wants to translate from text to sign first he opens the translation page and begin to translate

- if as usual if he did not enter any text or a entered a not understood text nothing will be translated so he must enter an understood text to translate successfully

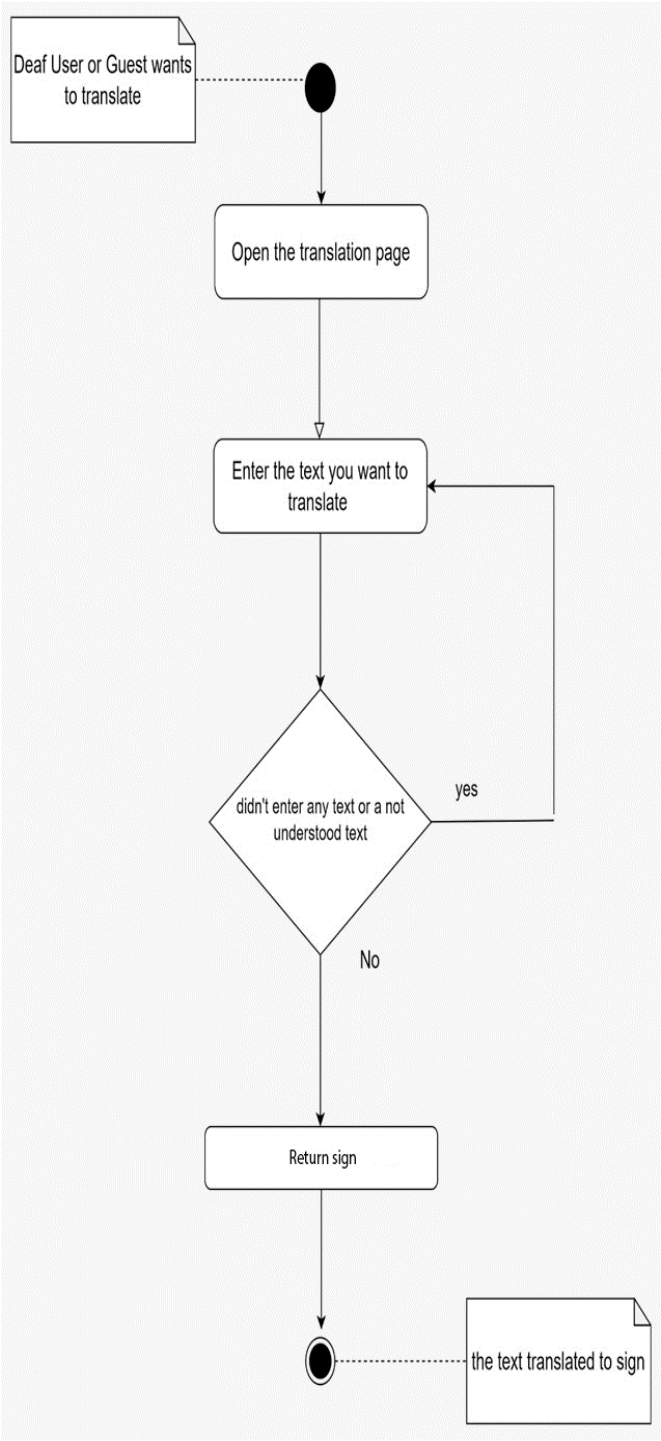


Figure 6 Translate from text to sign Activity

2.2.6 Education:

As the user explores the education section, he begins by logging in, the user first must login to view the course and start to learn.

- After that, he selects a course he is interested in. The system then checks if the chosen course is available.
- If it is, the user moves forward.
- if not, the user will be redirected to a support page for assistance. Once the course is confirmed available, the user gets access to its content.
- He can then interact with the material, participating in discussions and completing assignments, making the learning experience engaging and inclusive.

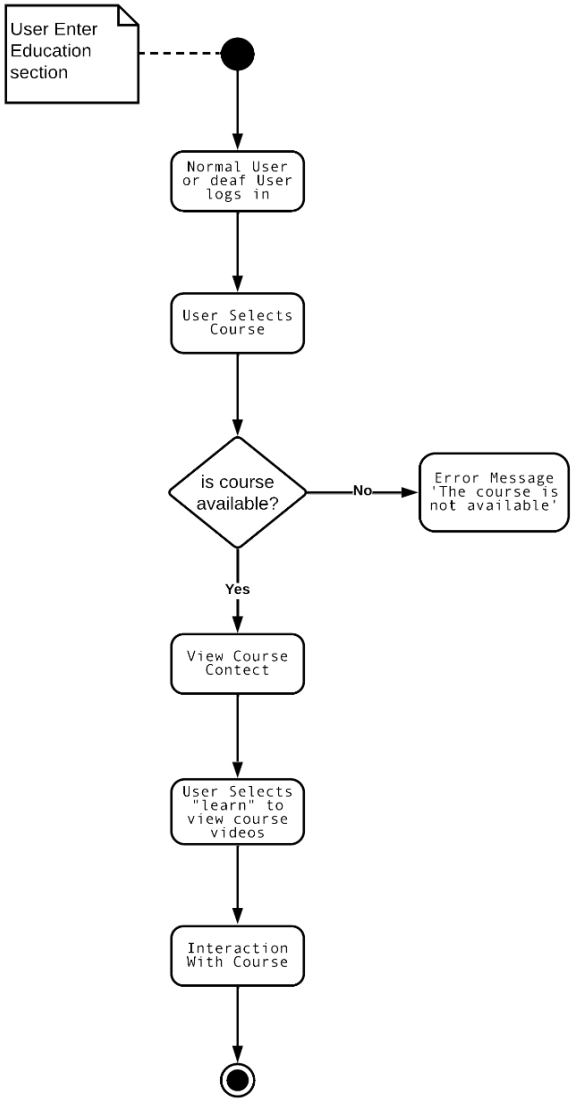


Figure 7 Education Activity

2.2.7 Add Course:

The course addition process starts with the admin logging into the course management system. The admin then goes to the section where courses can be added and provides necessary details like the course name and description. Clicking on "Add Course" saves this information.

The system checks for errors.

- If there are any, it asks the admin to fix them.
- If everything is error-free, the system stores the course details in the database.
- The activity concludes with a confirmation message, indicating that the course has been successfully added to the system.

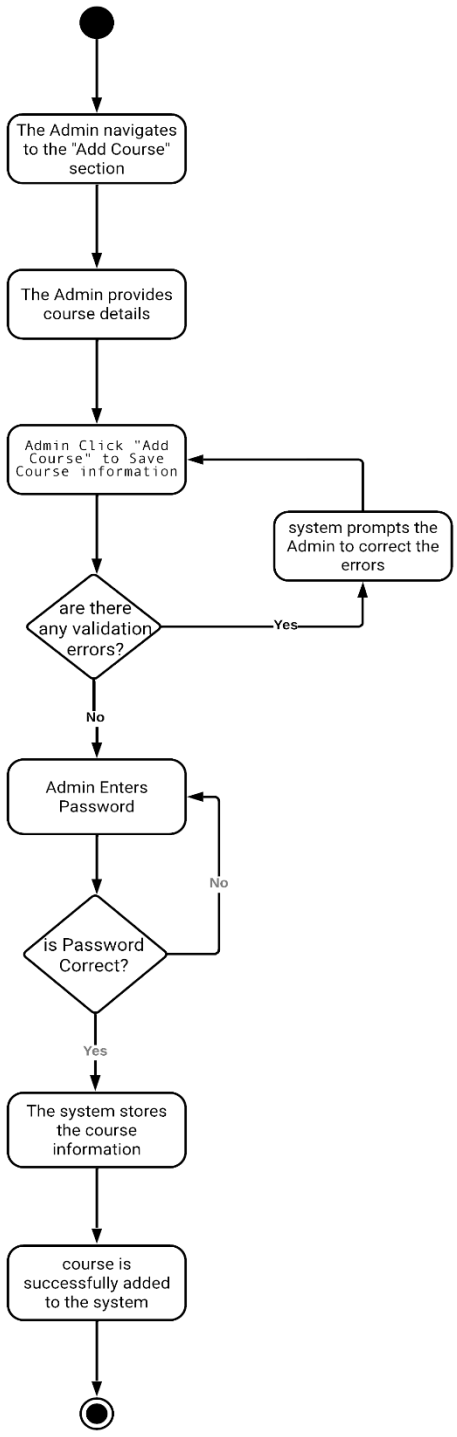


Figure 8 Add Course Activity

2.2.8 Remove Courses

The course Remove process starts with the admin logging into the course management system. The admin then goes to the section where courses can be removed and provides necessary details like the course id.

- Clicking on "Delete Course" removes this information.
- The system checks for errors. If there are any, it asks the admin to fix them.
- If everything is error-free, the system removes the course from the database.
- The activity concludes with a confirmation message, indicating that the course has been removed.

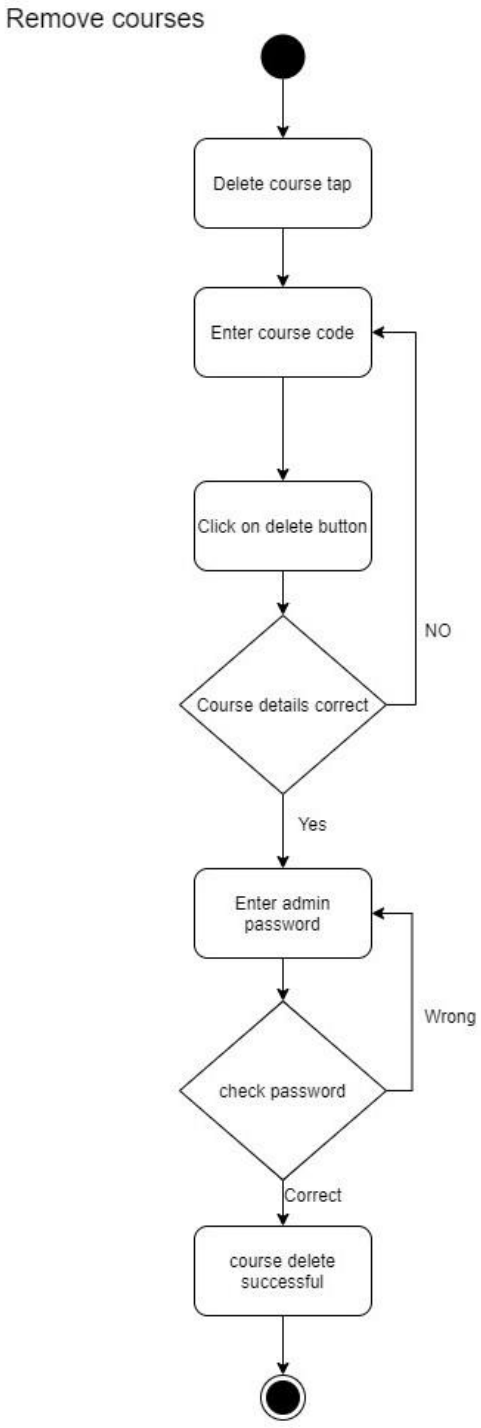


Figure 9 Remove Course Activity

2.2.9 Modify Courses

The course modify process starts with the admin logging into the course management system. The admin then goes to the section where courses can be modified and provides necessary details like the course id, name and description.

- Clicking on "Modify Course" modifies course information.
- The system checks for errors. If there are any, it asks the admin to fix them.
- If everything is error-free, the system modifies the course in the database.
- The activity concludes with a confirmation message, indicating that the course has been modified.

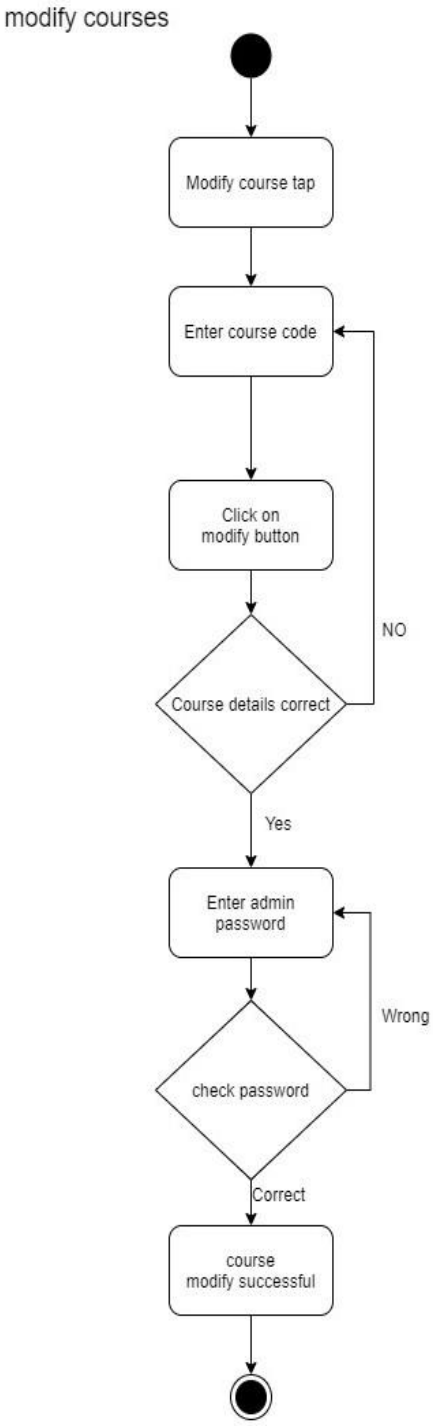


Figure 10 Modify Courses Activity

2.2.10 Chatbot

1- User Interaction:

The chatbot interacts with users through a chat interface embedded on the website. Users can type messages or questions, and the chatbot responds with relevant information or actions.

2- Task Assistance:

This chatbot are designed to assist users in completing specific tasks, such as answer about new courses or how to use the website.

- If the user needs more info, he can go to the admin email and ask him everything he wants.
- This chatbot are designed to assist users in completing specific tasks, such as answer about new courses or how to use the website.

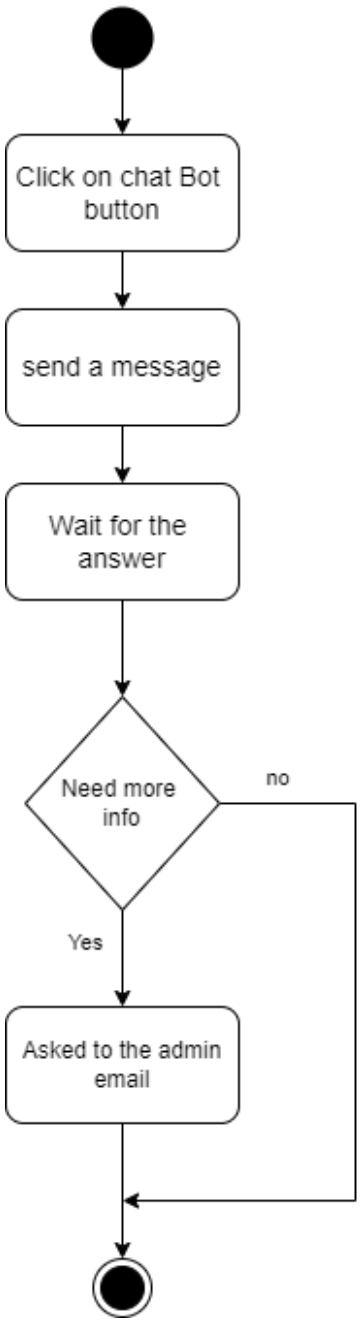


Figure 11 Chatbot Activity

2.2.11 Progress in Course

Normal and deaf users can see their progress in learning the courses.

progress in courses means how much user has learned from the entire course.

In the beginning, you must show the available courses.

then choose the appropriate course for you so that you can learn from it and follow in its footsteps.

By continuing your learning, you can see the extent of your progress in the course and how much you have accomplished from it.

Normal and deaf users can see their progress

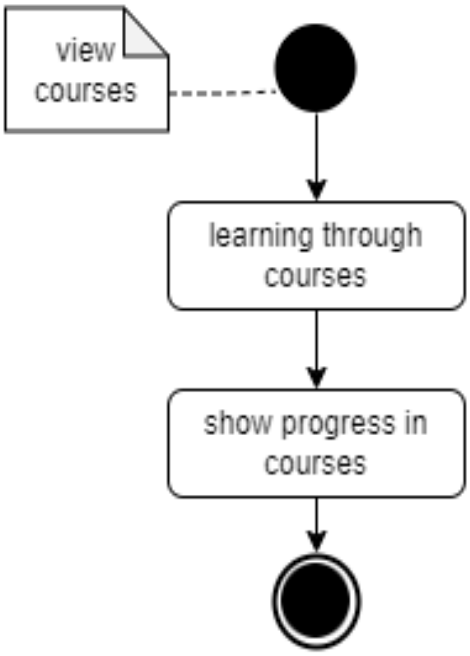


Figure 12 Progress in Course Activity

2.2.12 view courses

One of the goals of the project is to teach deaf and normal users sign language, which is done through courses. so, the user must view these courses. The user logs into his account

, Then he chooses view courses, through which all the courses in the system that are available to the user are displayed.

, The user chooses the course he wants.

, The details of this course and the educational path it will follow are displayed.

One of the goals of the project is to teach deaf and normal users sign language, which is done through courses. so, the user must view these courses. The user logs into his account

, Then he chooses view courses, through which all the courses in the system that are available to the user are displayed.

, The user chooses the course he wants.

, The details of this course and the educational path it will follow are displayed.

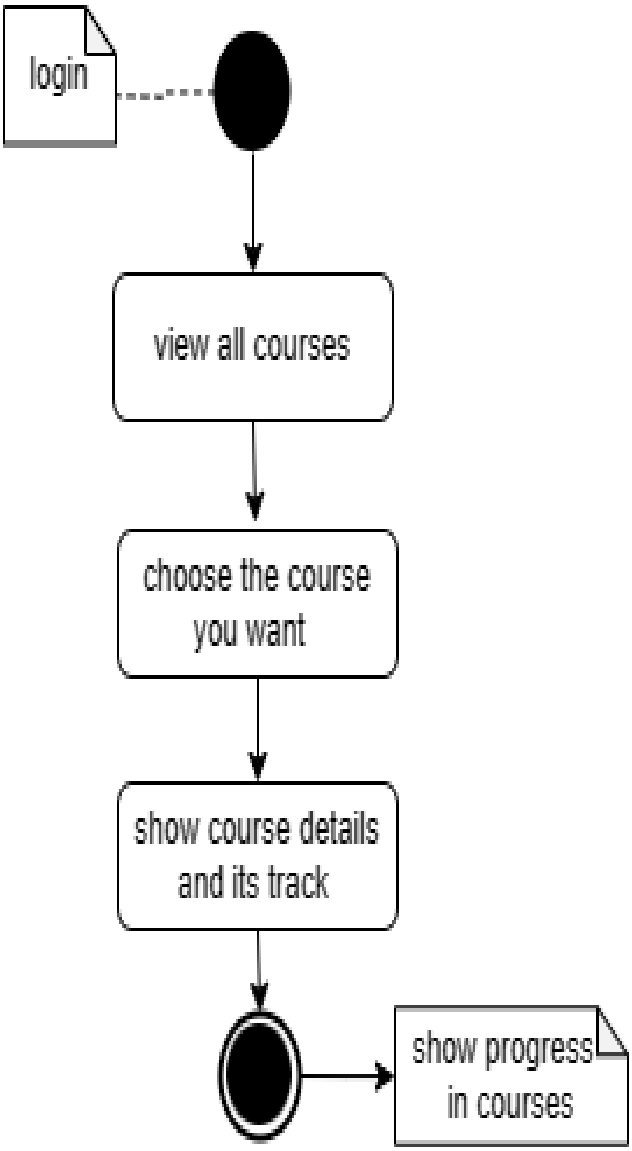


Figure 13 view courses Activity

2.2.13 Manage Account:

The Admin Can Manage User Accounts such as viewing, editing or deleting account.

The admin can view user information like:

- Name
- Username
- Profile photo.
- Gender
- And Enrolled courses

To view the information The admin must search for the user using his username.

If the username exists, the system will show the user information.

If Ther is no user with This username the system will show a message That The user doesn't Exist and Back again to Search page.

After viewing account admin can edit or delete account

To delete or edit Any user Account The admin must Enter his password to confirm deletion:

- If the password isn't correct the system will return to user profile Again
- If the password is correct the user will be deleted permanently

Deleting user Account is permanent and will delete any data related to the user like enrolled courses, progress in courses, profile photo, etc...

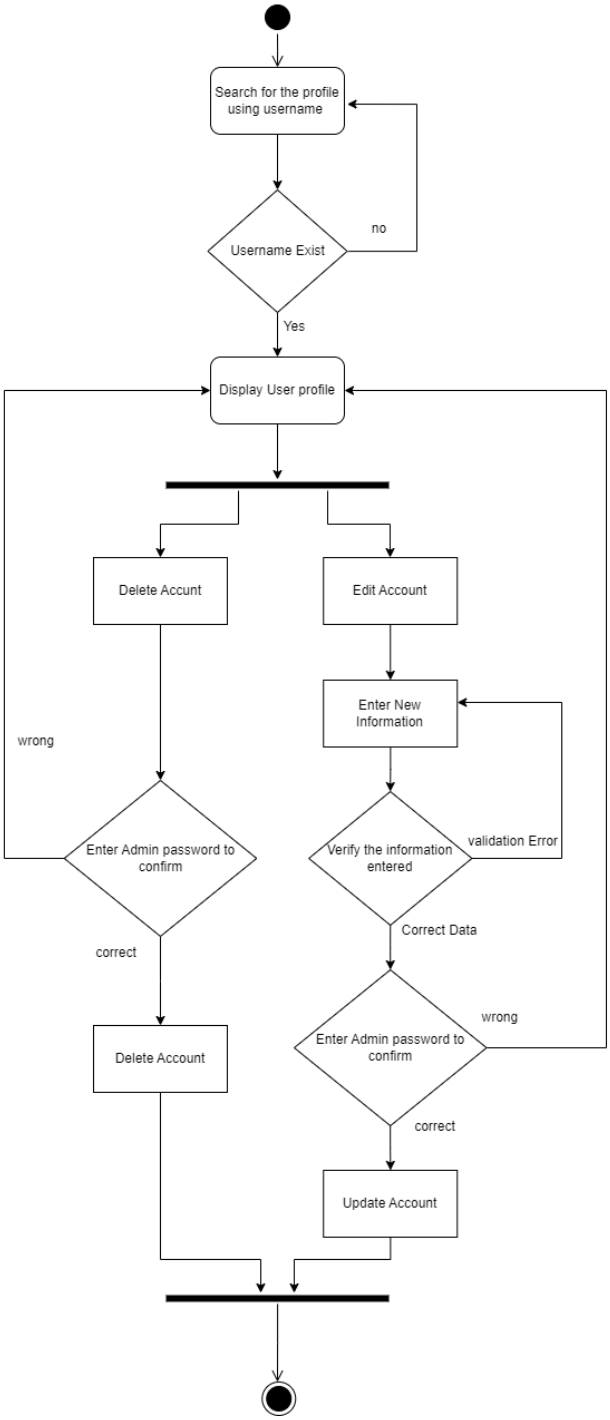


Figure 14 Manage Account Activity

2.3 Class Diagram:

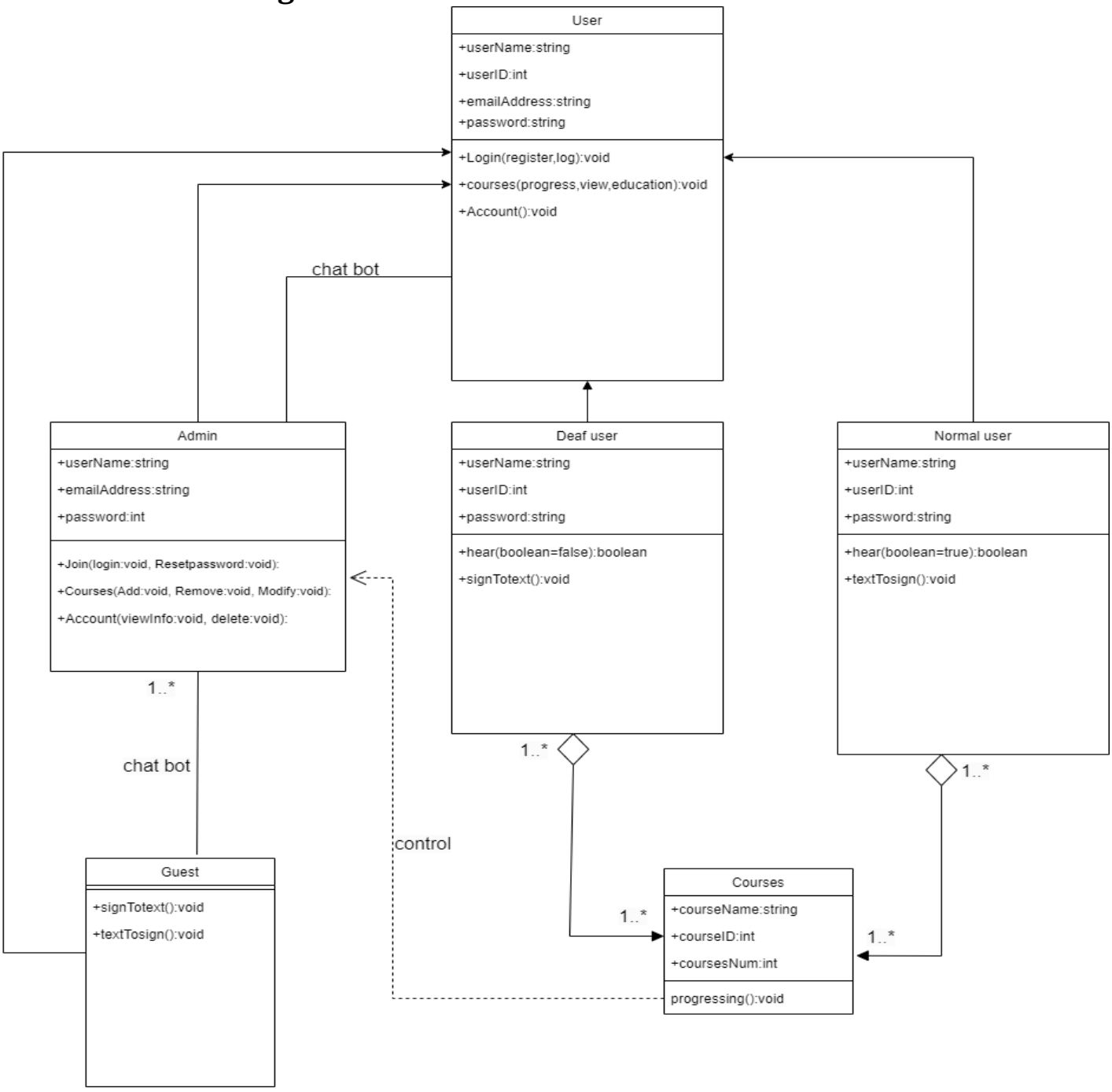
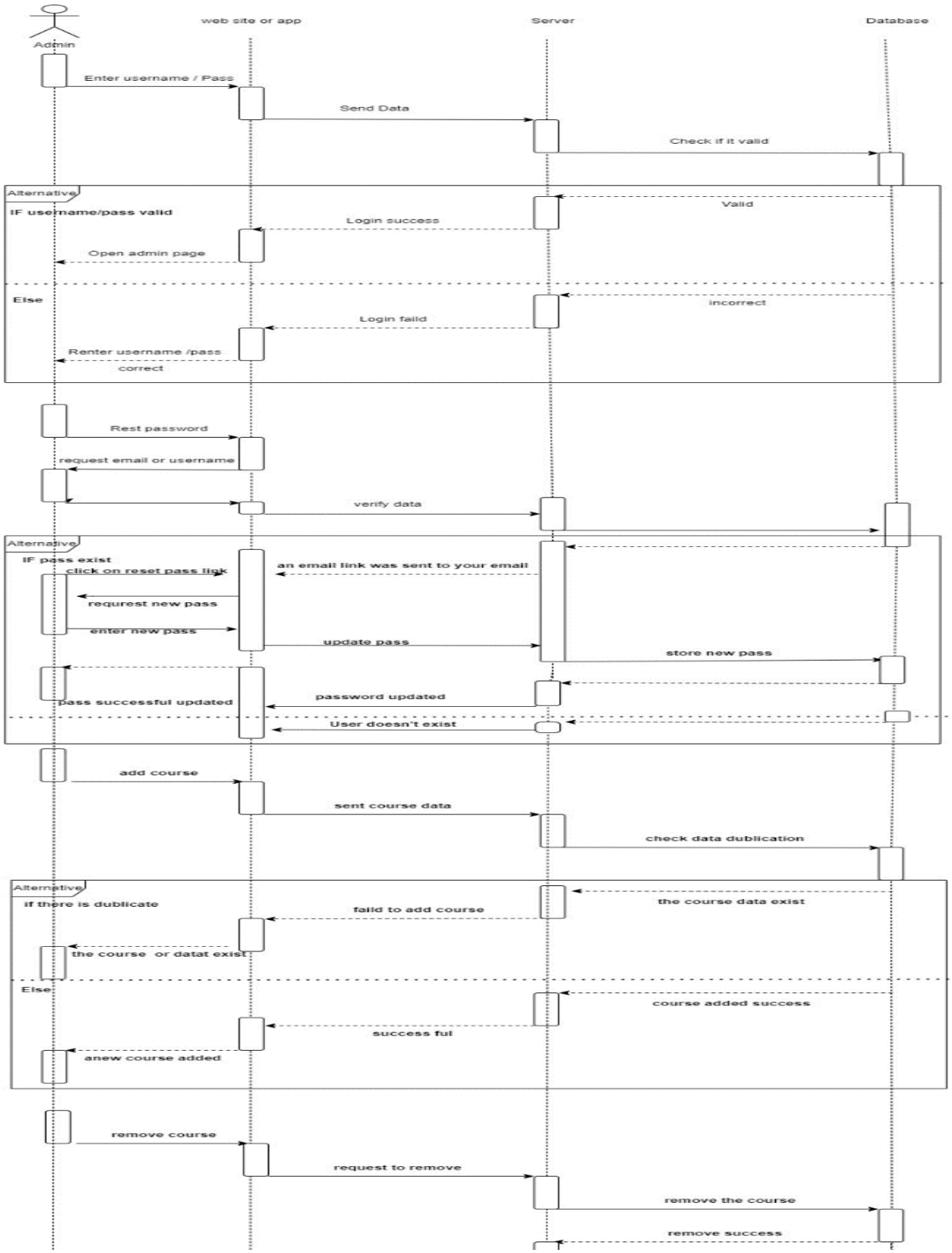


Figure 15 Class Diagram Activity

2.4 Sequence Diageam



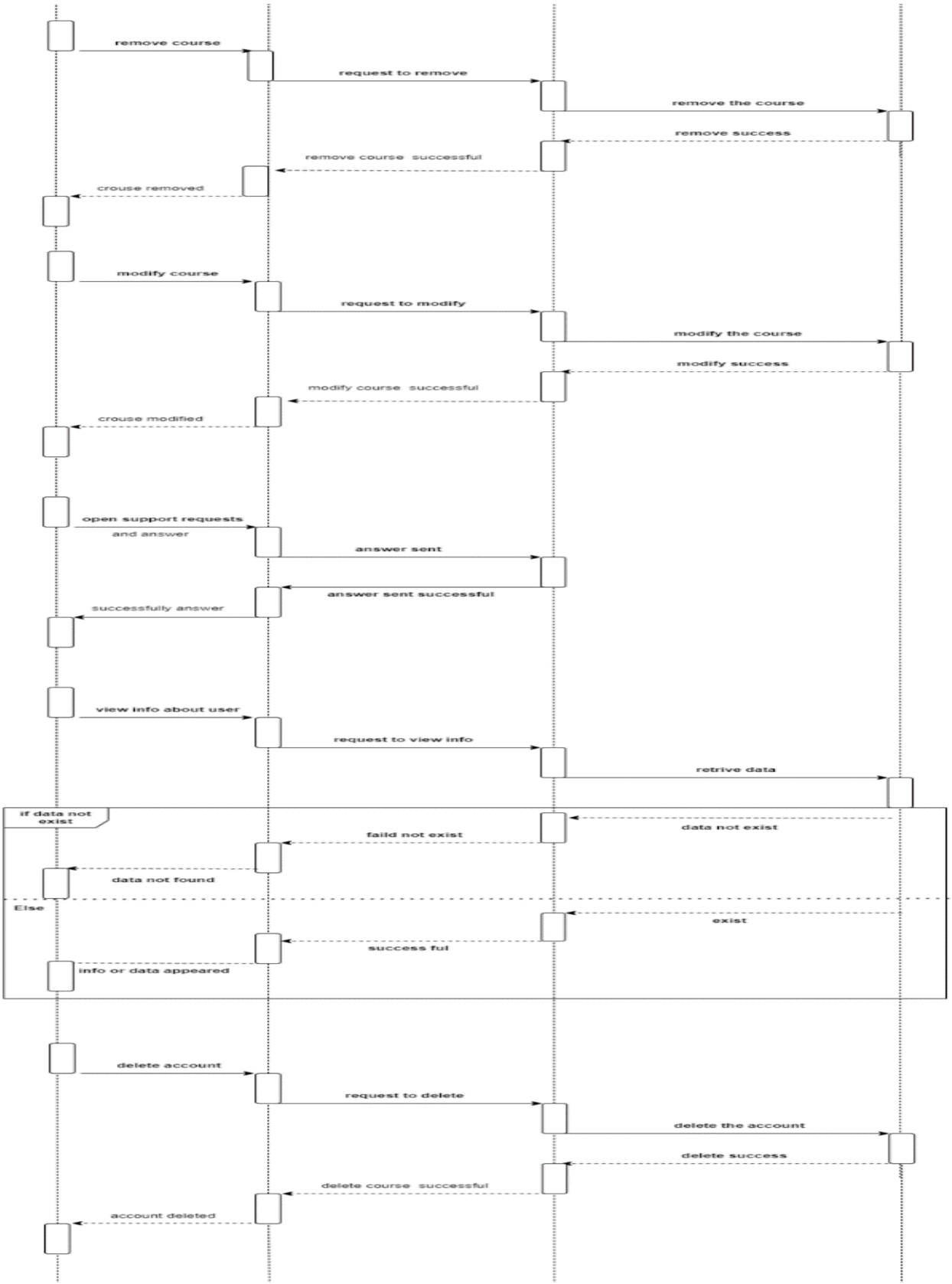
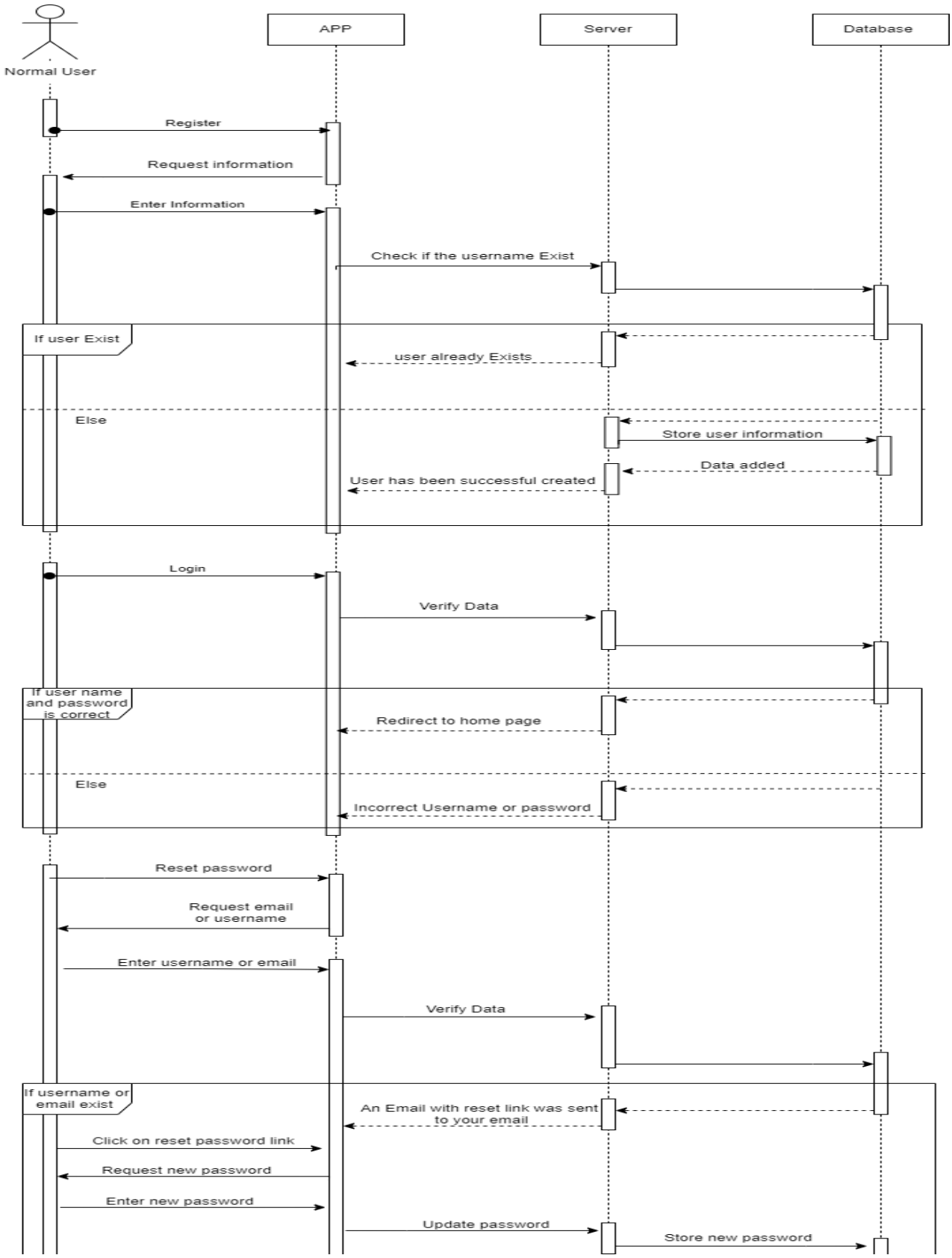


Figure 16 Admin Sequence


Normal User
APP
Server
Database
Register
Request information
Enter Information
Check if the username Exist
If user Exist
user already Exists
Else
Store user information
Data added
User has been successful created
Login
Verify Data
If user name and password is correct
Redirect to home page
Else
Incorrect Username or password
Reset password
Request email or username
Enter username or email
Verify Data
If username or email exist
An Email with reset link was sent to your email
Click on reset password link
Request new password
Enter new password
Update password
Store new password

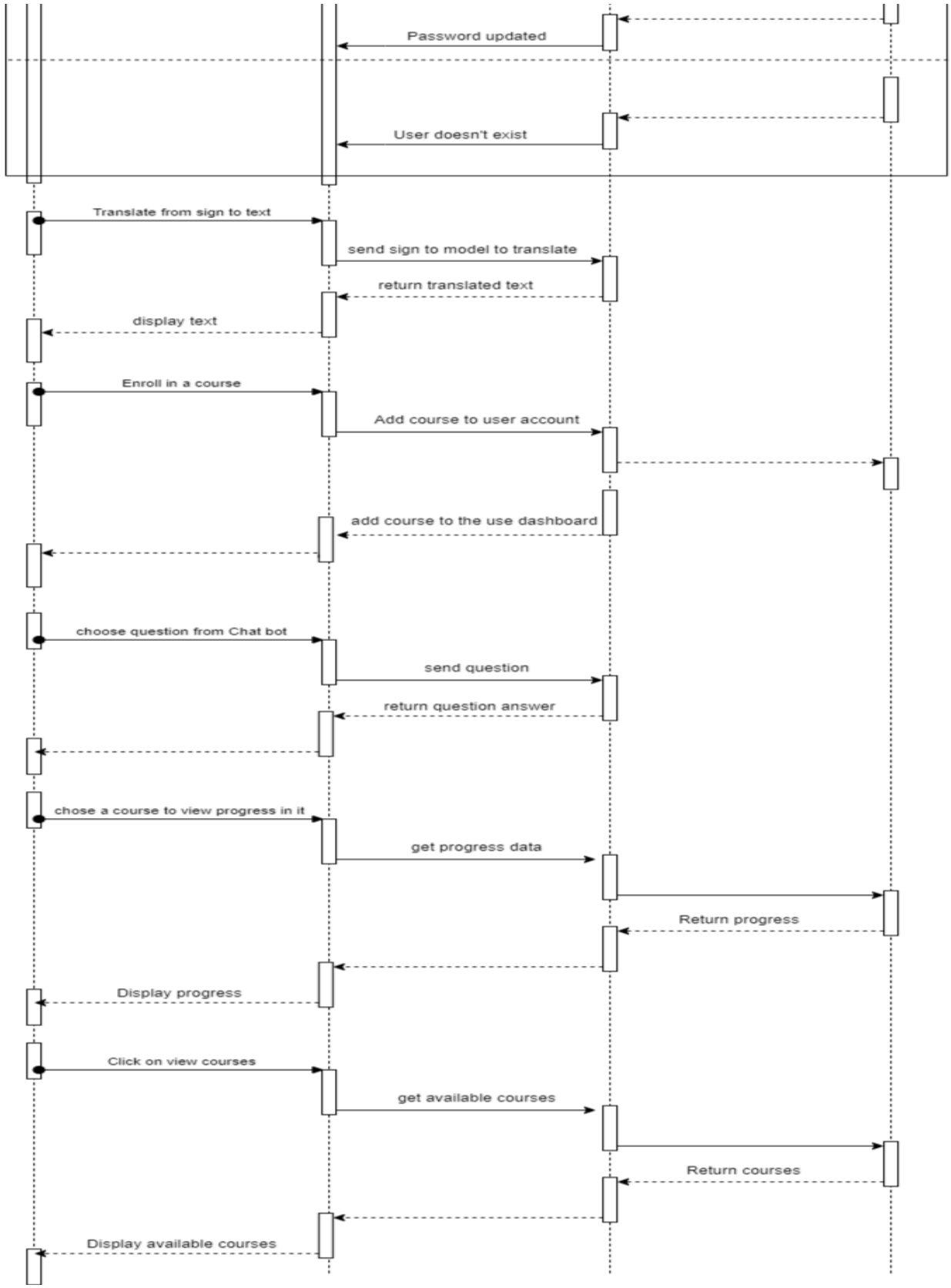
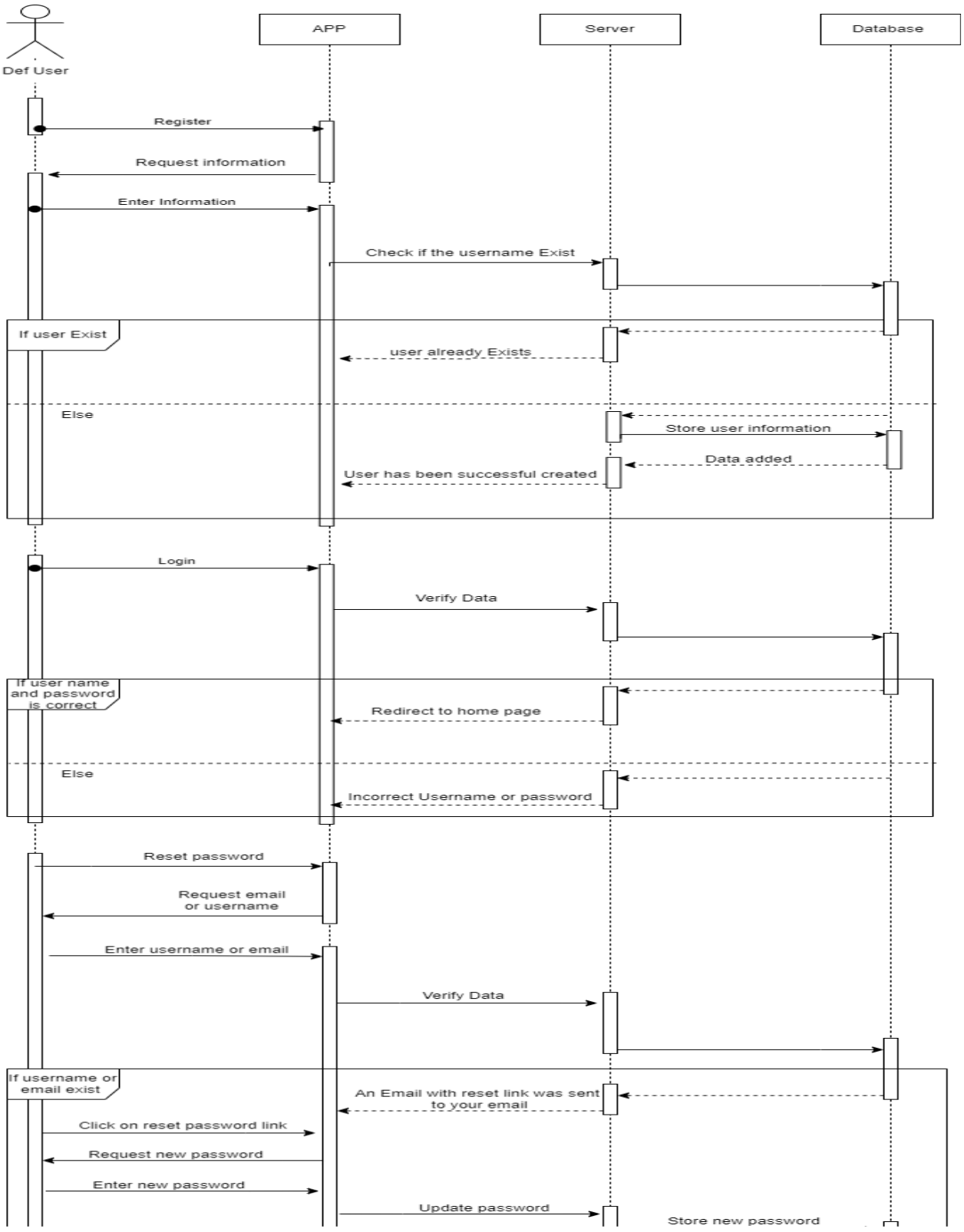


Figure 17 Normal user Sequence


Def User
APP
Server
Database
Register
Request information
Enter Information
Check if the username Exist
If user Exist
user already Exists
Else
Store user information
Data added
User has been successful created
Login
Verify Data
If user name and password is correct
Redirect to home page
Else
Incorrect Username or password
Reset password
Request email or username
Enter username or email
Verify Data
If username or email exist
An Email with reset link was sent to your email
Click on reset password link
Request new password
Enter new password
Update password
Store new password

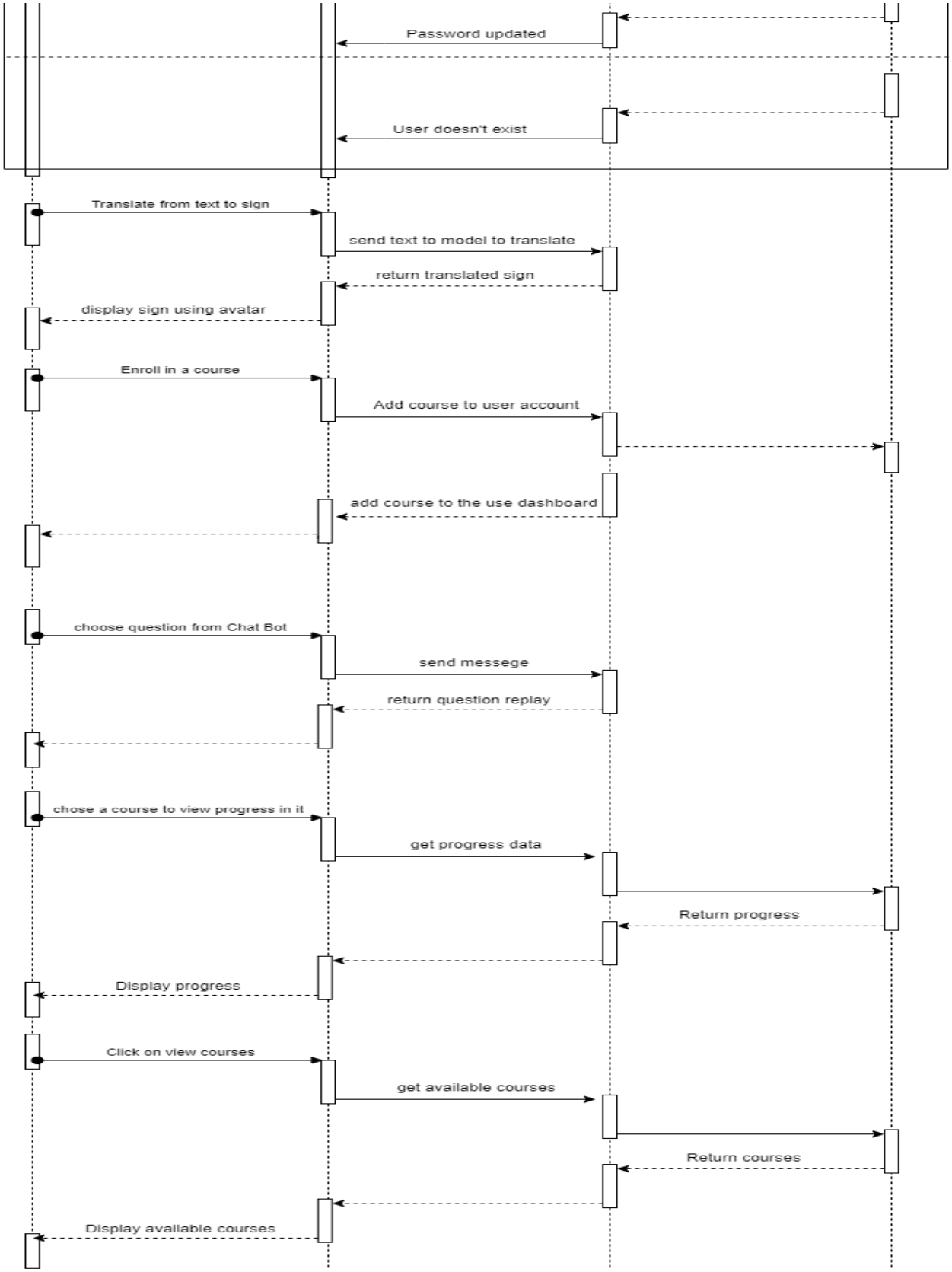


Figure 18 Deaf user Sequence

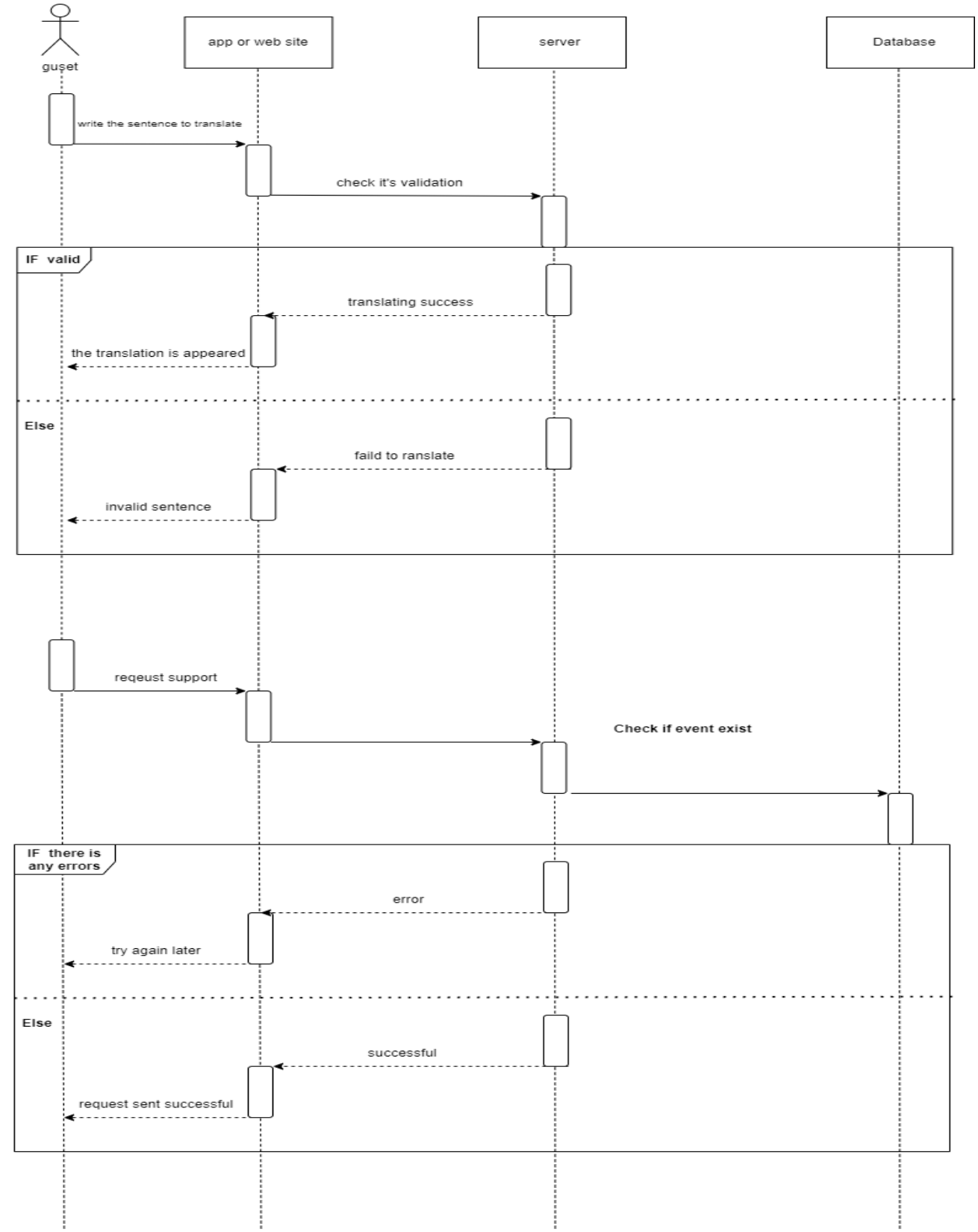


Figure 19 Guest Sequence

2.5 Data Flow Diagrams

The Data Flow Diagrams are used to graphically represent the flow of data in a business information system. DFD describes the processes that are involved in a system to transfer data from the input to the file storage and report generation.

2.5.1 Context Diagram – Level 0

This Context Diagram in **Error! Reference source not found.**…., shows The most prominent users are Normal User , Deaf User , Guest and Admin, and Normal User , Deaf User , Guest can use the Application or the Website to create a new account if they don't have one or log in directly if they have, then after the system verifies the data that they enter and make sure that it's valid data they can move to the home screen and access to the complete processes.

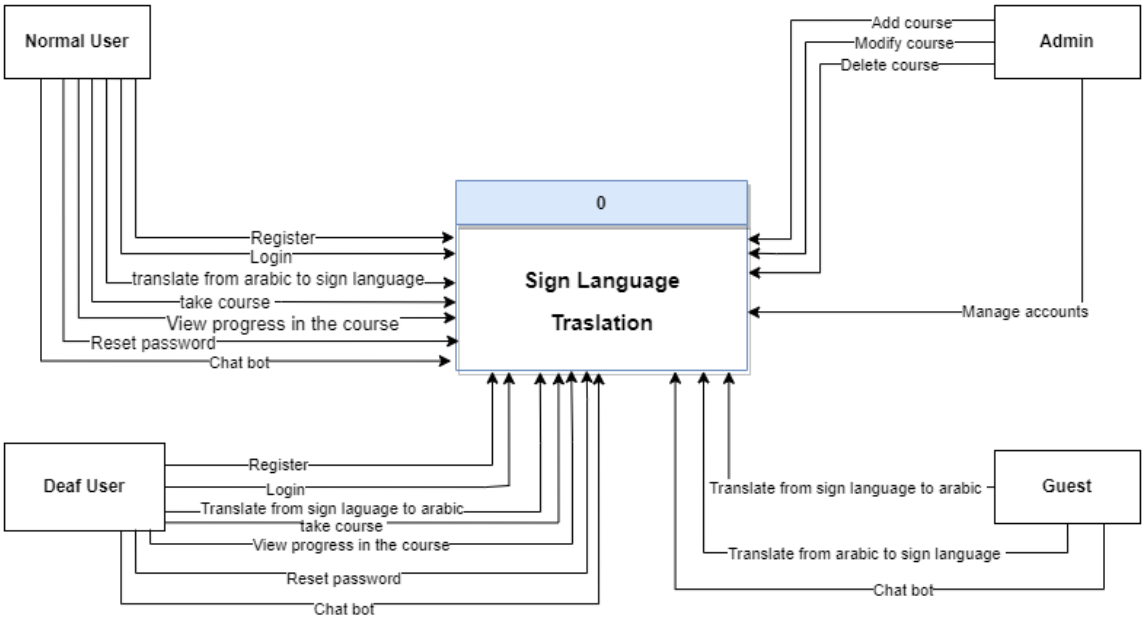


Figure 20 Context Diagram

2.5.2 Level 1

This Level in Figure …, shows in detail more than Context Diagram.

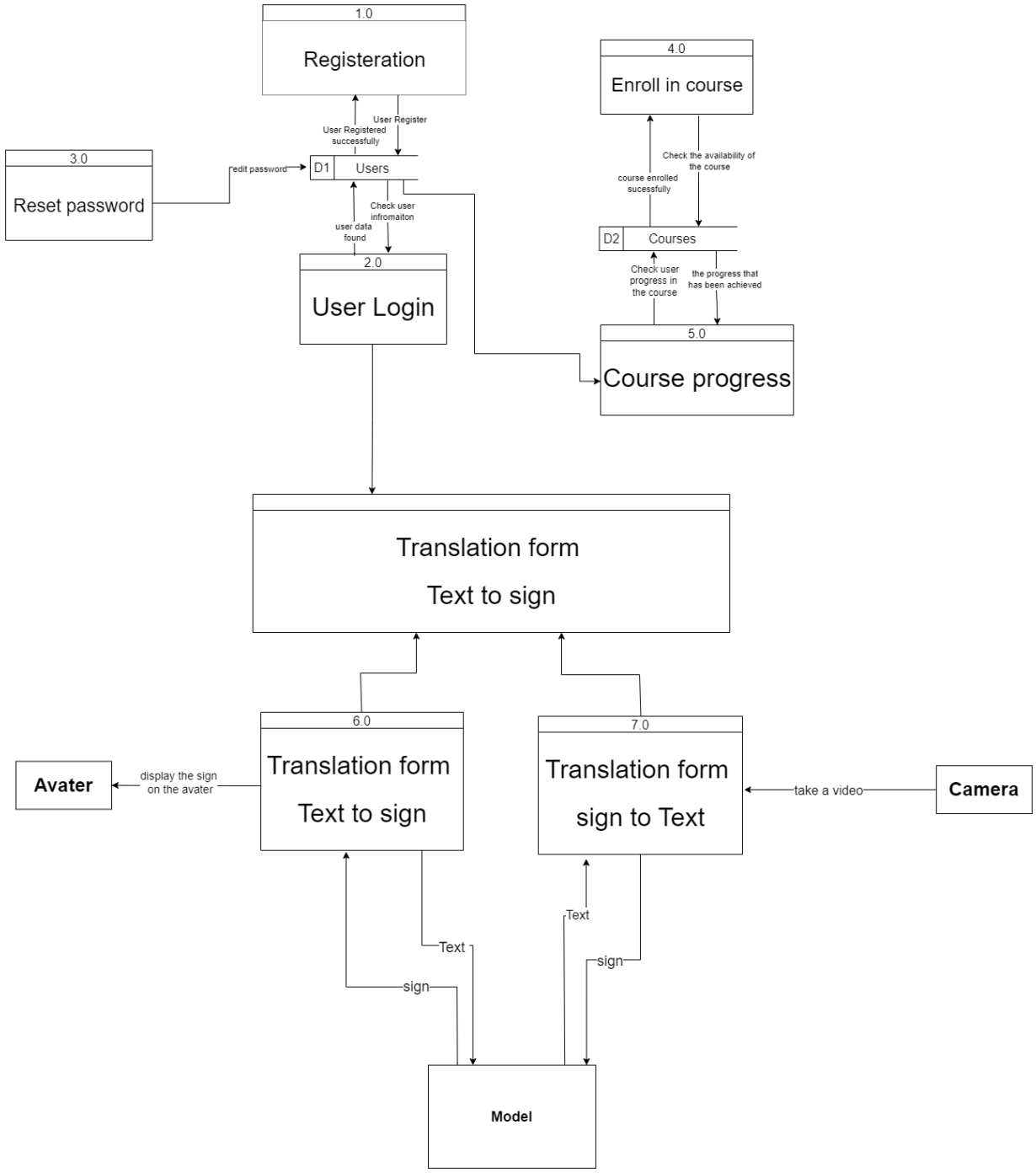


Figure 21 DFD

2.6 ER Diagram:

The ERD displays the entities in the system, which are user, admin, guest, course, chatbot, and translation system, along with their attributes.

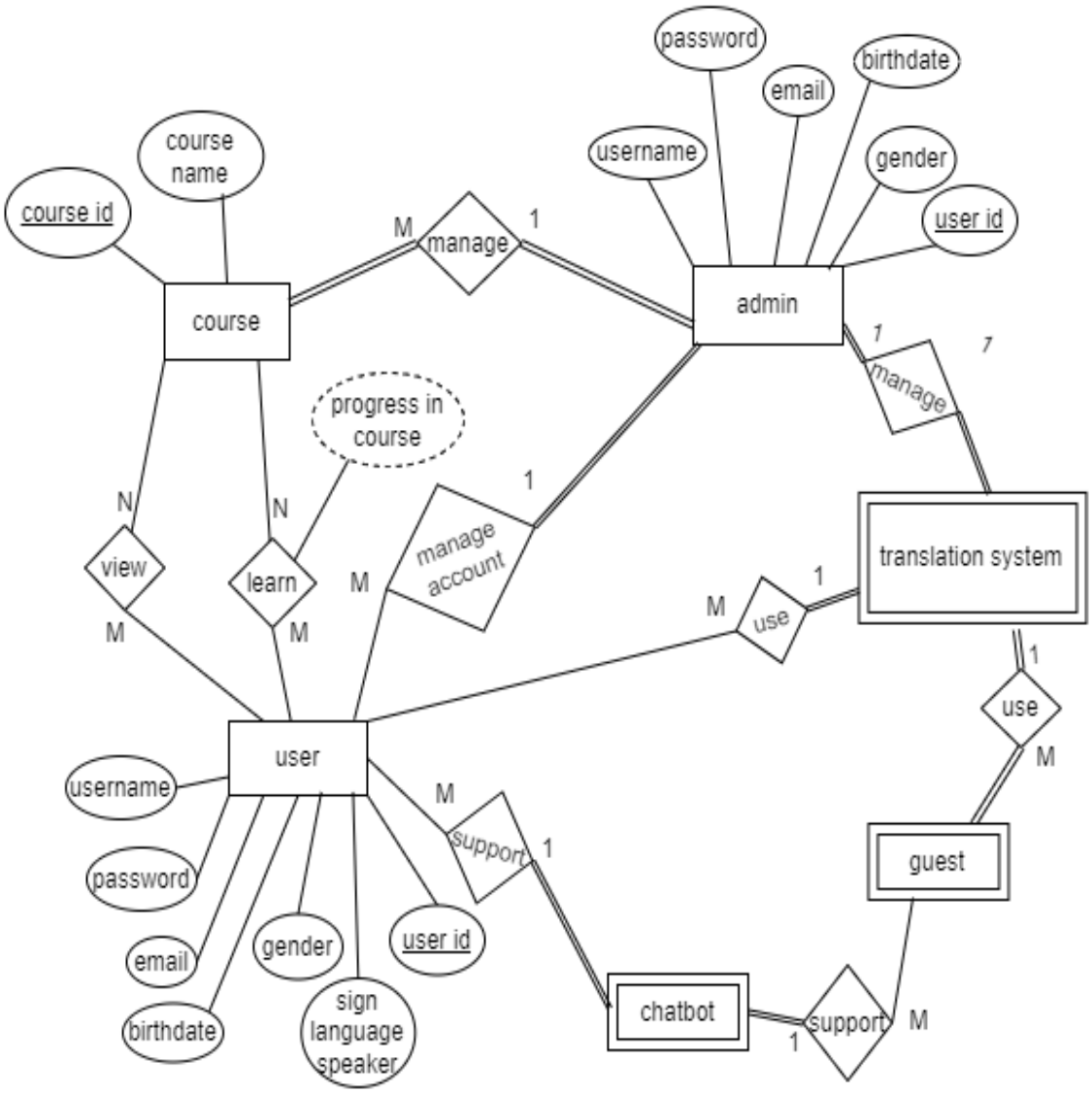


Figure 22 ER Diagram

3. IMPLEMENTATION AND EVALUATION

3.1 Current solutions

Currently, there are several solutions available in the market that focus on translating sign language to text and vice versa. These solutions mainly cater to widely spoken languages like English, American Sign Language (ASL), and others. However, there is a significant gap in solutions that support the translation of Arabic Sign Language (ArSL) to text and from text to animated avatars. Below, we outline some of the existing solutions, their capabilities, and their limitations, particularly in terms of supporting Arabic and Egyptian Sign Language.

1.1.1 3.1.1 Existing Solutions

- **Hand talk:**

  Description: Hand talk provides a comprehensive solution for translating American Sign Language (ASL) and Brazilian Sign Language (Libras) into text. It uses a combination of computer vision and natural language processing to interpret ASL gestures and convert them into written text.

  Strengths: Effective use of avatars for sign language translation Accurate ASL and Libras translation, real-time processing

  Weaknesses: no support for Arabic.

- **Sign Translateα:**

  Description: Sign Translateα provides a comprehensive solution for translating text into American, German and French Sign Language. It uses animated avatars to display sign language translations.

  Strengths: Effective use of avatars for sign language translation.

  Weaknesses: does not support Arabic.

- **SIGNARA:**

  Description: SIGNARA provides a comprehensive solution for translating text into Arabic Sign Language.

  Strengths: support Arabic Sign Language.

  Weaknesses: represent 10 words of Arabic and represent other words by letters

3.2 DATASET

3.2.1 Introduction:

Sign language can be divided into three main categories to be recognized: alphabets and numbers, words and sentences. Alphabets and numbers are mainly static gestures. Words are mainly dynamic gestures. But sentences are a mixture between static and dynamic gestures that must be separated to be correctly translated. The recognition of the first two categories is called isolated recognition as one gesture is performed at a time. While recognition of the sentences is called continuous recognition as more than one gesture is performed in a sequence without boundaries.

A sign language recognition system has one essential target which is interpreting sign language to spoken or written language. In the case of spoken languages, translating between two different languages requires the availability of a huge equivalent vocabulary from both languages to make this translation applicable. For example, in the Google Translate engine, it was clear that as more data become available for the training model the more efficient the Arabic-English statistical machine translation system works. To reach an accuracy of nearly 53% more than 200 billion words are used. On the other hand, in the case of sign language small-scale equivalent vocabulary is available. There are a few benchmark databases for different sign languages, but they still have small-scale vocabulary as the largest database available is RWTH-Phoenix-Weather which includes only 1,558 sign words. This lack of databases makes any available system inapplicable and not suitable for real-time applications.

3.2.2 ArabicSL-Net:

The data was captured by mobile camera in four main organization namely Bank, Cafe, Hospital, and Train station. The ArabicSL-Net initially consists of 307 words recorded in approximately 30,000 videos. For each organization, capture the most representative words that are used in those places. For Bank data, we have a total of 76 words, while Cafe data contains 54 words. For Hospital, we have videos for 102 words, and videos for 71 words in Train station.

After Removing duplicate Words between the 4 Main Classes (Bank, Cafe, Hospital, and Train station) The total number of words in the data set is 200 words as in the following Table:

ساعة	الفيروسات	ممرضة	الدم	سؤال	الإسكندرية	تغيير	المشروع
المكان	المرض	الحقن	التهاب الزائدة الدودية	لا	تأشيرة	جديد	التمويل
حيث	التعب	صداع	الخدمة	برتقال	الحصول على	نقود	البنك
بعد	الحمل	الحرق	الإمساك	حليب	الدعم	العنوان	الاسم
عندما	التحليل	العينين	الرعاية الصحية	الحلبة	النافذة	مدير البنك	لأن
المنصورة	تذكرة	شاش	مرهم	زجاج	مترجم لغة الإشارة	الشقة	معاش

التأخير	القطن	الضغط	نموذج	شراب	عام	الاستمارة	المحافظة
في وقت متأخر	الولادة	المرض	الأذن والأنف والحنجرة	النعناع	الإيرادات	الدفعة الشهرية	و
بني سويف	الاحتقان	ارتفاع درجة الحرارة	المغص	البيريل	النفقات	التفويض	التوقيع
محطة القطار	الكانولا	الإجازة	الأشعة	ملعقة	صحتك بخير؟	لي	انتظر
أسيوط	ممنوع	فقر الدم	القلب	قهوة	في	القيام	رخصة المحل
سوهاج	القناع	التأمين الصحي	التسمم	فراولة	أنت	المزايا المالية	بطاقة الخدمات المتكاملة
معًا	الرضاعة	الباطنية	السكري	بارد	بلدي	الرقم	الشروط
السويس	كورونا	المستشفى	أمراض النساء والتوليد	بيبسي	كيف حالك	على	السكن
الجيزة	الإسهال	الاستقبال	الشراب	يانسون	كم	الرفض	المؤهلات
الإسماعيلية	الكسر	الطوارئ	الورق	القرفة	رخصة القيادة	ماذا	نعم
الإسماعيلية	خصم	الكبد	البول	شاي	المساعدة	أو	الحمد لله
سيأتي	طنطا	الجلد	طبيب	ليمون	شهادة الوفاة	شهادة الميلاد	دفتر التوفير
بورسعيد	دمياط	مقص	علامة تبويب	عصير	الحساب	مرحبًا	الفرع
قبل	دمنهور	دماغ	براز	ساخن	سحب الأموال	الحاجة	أصم
الفيوم	قنا	نوبة قلبية	الأطفال	نسكافيه	الإيجار	الملكية	بطاقة الهوية
كفر الشيخ	بنها	العلاج الطبيعي	أخبرني	كركديه	خاص	مفتوح	حسناً
المنيا	أسوان	دوار	القسم	الموز	غرامة	عقد	قرض
أسوان	مرسى مطروح	غرفة	الجراحة	ماء	حسابك	القاهرة	إيداع
القطار	الأقصر	عظام	الأعصاب	المانجو	مشكلة	الراتب	تحويل الأموال

3.3 Modeling

3.3.1 OpenHands Sign Language Recognition:

OpenHands is an open-source software toolkit designed to facilitate research in Sign Language Recognition (SLR), particularly with a focus on pose-based recognition techniques. It aims to democratize SLR research by providing accessible tools and resources, fostering collaboration, and establishing standardized methods within the field.

Key functionalities of OpenHands include:

- **Pose-Based SLR Framework:** Enables researchers to leverage existing hand pose estimation methods (e.g., MediaPipe) to represent signs for real-time processing capabilities.
- **Standardized Datasets:** Offers carefully curated datasets for six sign languages (American, Argentinian, Chinese, Greek, Indian, and Turkish). These datasets contribute to consistency and comparability across research efforts.
- **Pre-trained Models:** Provides baseline pre-trained models for pose-based SLR on the aforementioned sign languages. Researchers can utilize these models as a foundation for further development and experimentation.
- **Self-supervised Pretraining Techniques**: Introduces methods for training models on unlabeled data. This addresses the challenge of limited availability of labeled sign language data, a common bottleneck in SLR research.

Experimental Setup in OpenHands

The OpenHands project employs an experimental setup that evaluates four deep learning models for sign language recognition: Long Short-Term Memory (LSTM), Bidirectional Encoder Representations from Transformers (BERT), Spatial-Temporal Graph Convolutional Network (ST-GCN), and Sign Language Graph Convolutional Network (SL-GCN). Their approach leverages PyTorch Lightning for streamlined data processing and training pipelines. All models are optimized using the Adam optimizer.

Model Training Parameters in OpenHands

OpenHands defines the following training parameters for each model:

- **LSTM:** Batch size: 32, Initial learning rate: 0.005
- **BERT:** Batch size: 64, Initial learning rate: 0.0001
- **ST-GCN & SL-GCN:** Batch size: 32, Initial learning rate: 0.001

Hardware and Training Data Usage in OpenHands

OpenHands utilizes a single NVIDIA Tesla V100 GPU for training. Notably, they train exclusively on the provided training sets for each dataset. This contrasts with some

previous works (e.g., AUTSL) that incorporate both training and validation sets for final test accuracy reporting.

Open-Source Availability through OpenHands

OpenHands offers public access to the trained models and their corresponding training configurations within the OpenHands toolkit.

The Accuracy of different models across datasets was as in the following Table:

Dataset	Language	State-of-the-art (pose) model		Model available in 🙌 OpenHands			
		Model (Params)	Accuracy	LSTM	Transformer	ST-GCN	SL-GCN
INCLUDE	Indian	Pose-XGBoost	63.10	83.0	90.4	91.2	**93.5**
AUTSL	Turkish	Pose-SL-GCN^2 (4.9M)	**95.02**	77.4	81.0	90.4	91.9
GSL	Greek	Pose-Attention (2.1M)	83.42	86.6	89.5	93.5	**95.4**
DEVISIGN_L	Chinese	RGB-iRDML	56.85	37.6	48.9	55.8	**63.9**
CSL	Chinese	RGBD-I3D (27M)	95.68	75.1	88.8	94.2	**94.8**
LSA64	Argentinian	Pose-LSTM (1.9M)	93.91	90.2	92.5	94.7	**97.8**
WLASL2000	American	Pose-TGCN (5.2M)	23.65	20.6	23.2	21.4	**30.6**
		Average accuracy	→	69.38	73.47	77.43	80.69

Based on the results we decided to choose SL-GCN as Our Model for Training on ArabicSL-Net Dataset.

To reduce the training time, we applied OpenHands mediapipe_extract script to extract poses from videos and add it to .pkl file.

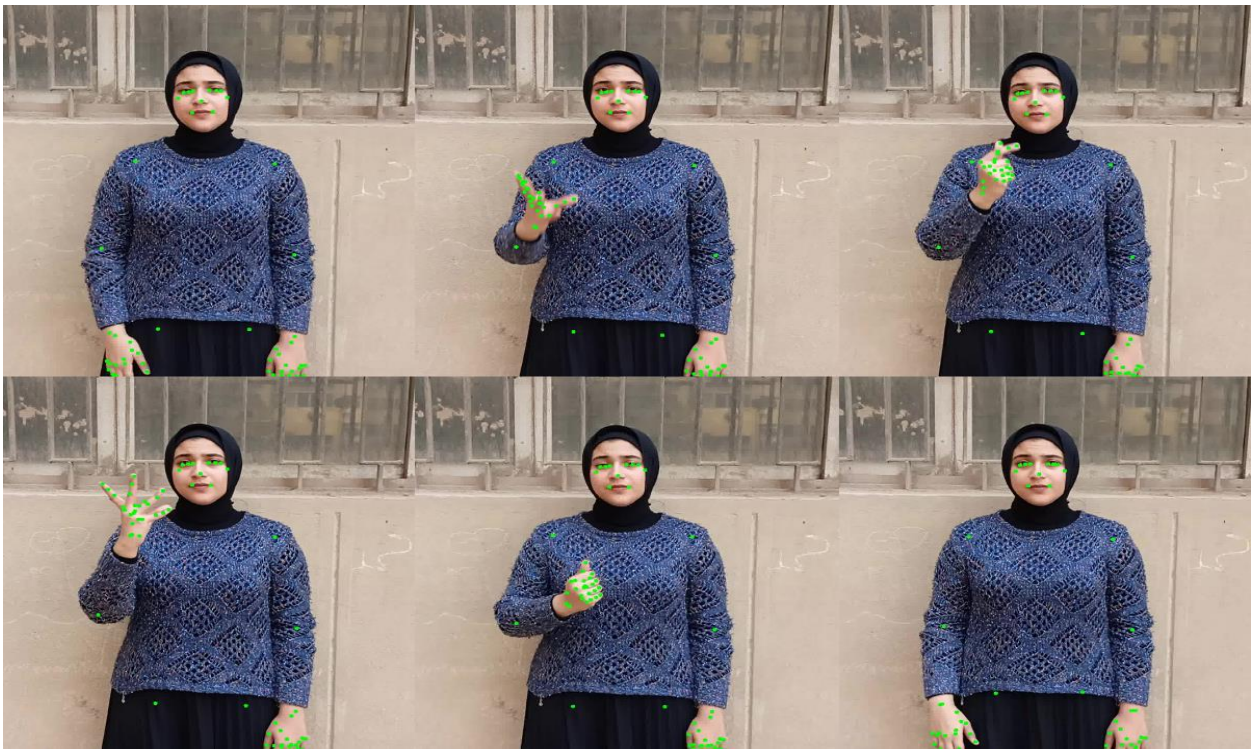


Figure 23 video after pose estimation

For the training process we needed to create a class for the dataset, so we created our class called arasl.py and added the data classes names to arasl.md that contains each word name with its index.

After the training process the model achieved the highest accuracy on epoch 91 with the following results for the training and validation.

Training Accuracy: 0.9942

Validation Accuracy: 0.985

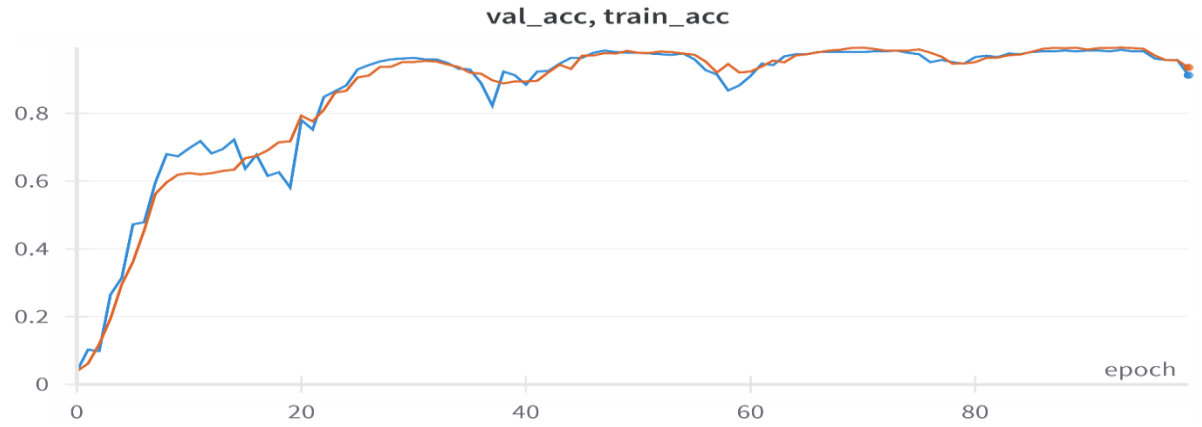


Figure 24 Training and validation Accuracy

And the following loss:

Training Loss: 0.00423

Validation Loss: 0.05288

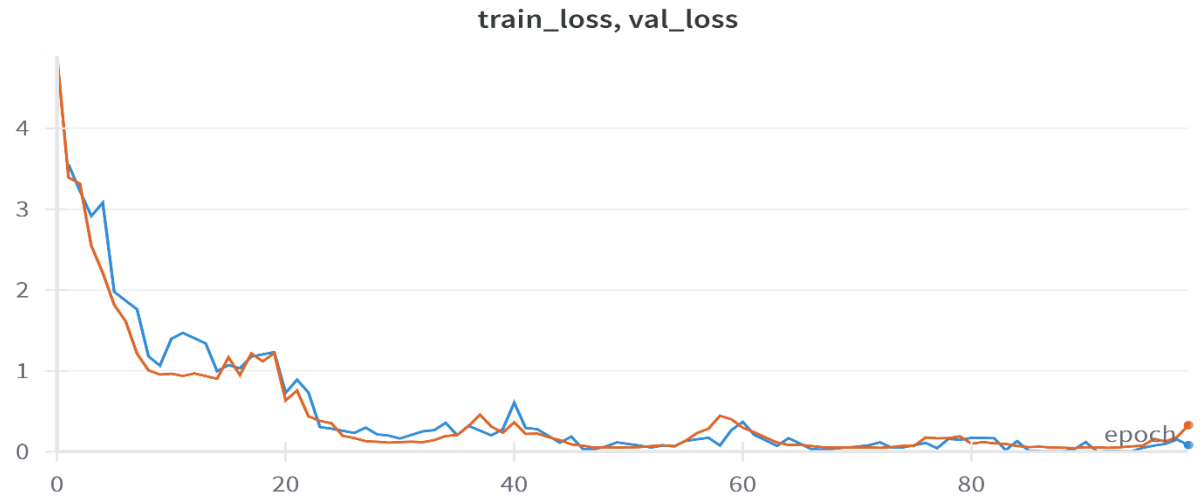


Figure 25 training and validation loss

For the inference process OpenHands has an inference script but the script expect to get the test data from data Loader that loads the data from the test path specified in the decoupled_gcn.yaml configuration file. So, we developed our own function that takes the pose file directly and apply the classification then return the result.

To reduce inference time we used multiprocessing with the following strategy:

Initialization

1. **Multiprocessing Pool Initialization**:
    - A global multiprocessing pool is initialized at the startup event. The pool size is determined by the number of available CPU cores, up to a maximum of four. This pool will be used to distribute the workload across multiple processes.

Loading and Processing Video Frames

2. **Loading Video Frames**:
    - Video frames are loaded using OpenCV and converted from BGR to RGB format. This conversion is necessary because the mediapipe holistic model processes frames in RGB format.
3. **Dividing Frames into Segments**:
    - The loaded video frames are divided into segments based on the number of cores in the multiprocessing pool. This ensures that each core gets an approximately equal share of the frames to process, distributing the computational load evenly.

Parallel Processing

4. **Generating Keypoints for Each Segment**:
    - Each segment of frames is processed in parallel to extract keypoints. The extraction process involves using the mediapipe holistic model to detect and extract keypoints for the body, face, and hands from each frame.
    - The keypoints for each frame in the segment are collected into arrays, which are then combined into a single result for that segment.
5. **Combining Results from All Segments**:
    - After all segments have been processed, the results from each segment are collected and combined to form the final arrays of keypoints and confidence scores.
    - This combination step merges the keypoints and confidences extracted from each segment into comprehensive datasets representing the entire video.

3.3.2 Text Processing with Octopus

We utilize the Octopus library, which provides a variety of natural language processing (NLP) capabilities. The main tasks supported include:

- **Text Diacritization:** Adding diacritical marks to Arabic text.
- **Grammatical Correction:** Identifying and correcting grammatical errors.
- **Title Generation:** Generating titles for given text.
- **Paraphrasing:** Rewriting text in a different form while maintaining the same meaning.
- **Question Answering:** Providing answers to questions based on a given context.
- **Question Generation:** Creating questions from given text.
- **Translation:** Translating text from one language to another.

The "process_text" function is the core of this modeling part. It maps each task to its corresponding prefix and uses the Octopus model to generate the desired output. The model's generation process is configured using specific options such as beam search to ensure high-quality results, we only use Paraphrasing task to make the sentence that is coming from the translate model more Understandable.

3.3.3 Audio Transcription with SpeechRecognition

We focus on audio transcription using the SpeechRecognition library. The application accepts audio files in ".wav" format and processes them to extract the spoken content as text. The transcription process involves the following steps:

- File Upload: Users upload audio files through an HTTP request.
- File Validation: The application checks if the uploaded file is valid and allowed (only.wav files).
- Transcription: Using the SpeechRecognition library, the application reads the audio file and transcribes it using Google's speech recognition API, specifically for the Arabic language (Egyptian dialect).

3.4 Avatar:

3.4.1 Overview:

We created an avatar character to translate written text into sign language using a sequence of animations.

Figure 26 Avatar character

3.4.2 Challenges:

Designing the avatar.

Animating the avatar and creating animations for each word in sign language.

3.4.3 Technologies Used:

- Unity.
- mixamo.com for characters designs.

3.4.4 Steps to Overcome Challenges:

- **Designing the Avatar:**
  - We used the Mixamo website to freely select a design from hundreds of characters and download it with its textures.
  - The downloaded avatar was then imported and opened using the Unity program.
- **Animations:**
  - Initially, we attempted to create animations using a Python library called Mediapipe by extracting poses from videos. However, this approach did not work.
  - Consequently, we manually created animations by moving the avatar's hands and recording each movement or change. This method allowed us to create approximately 150 animations for various words in sign language.

Figure 27 developing animation using unity

3.4.5 Weakness:

- **Animation:**
  - Achieving natural and fluid animations can be very difficult.
  - To improve, study animation principles, use motion capture if possible, and refine animations through iterative testing and feedback.
- **Programming Proficiency:**
  - Writing efficient and bug-free scripts requires good programming skills in C#, which is used to manage and synchronize animations.

3.5 Web Application

3.5.1 Web Directory Structure:

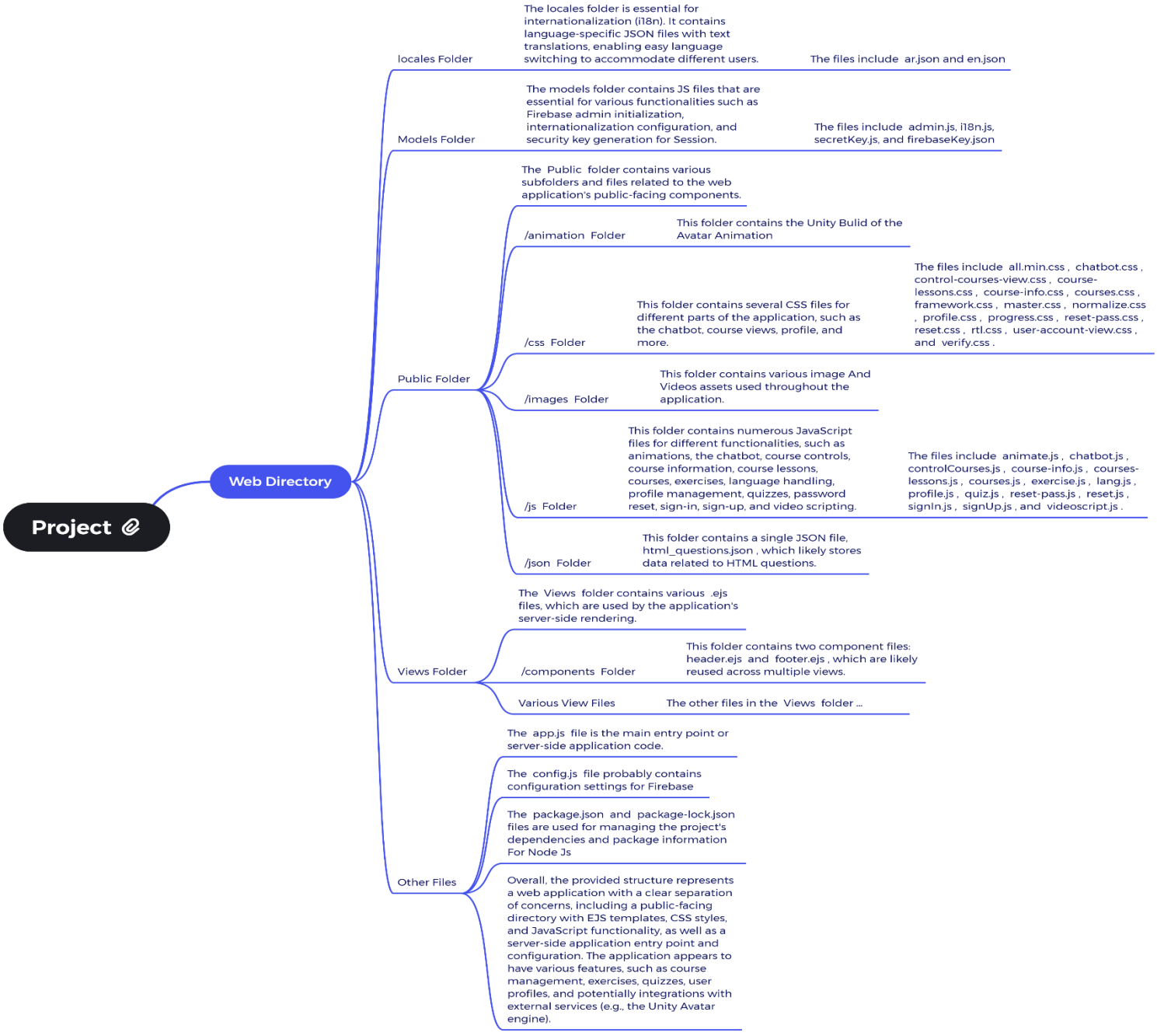


Figure 28 Web Directory Structure

3.5.2 Front End

Overview

The front-end of this website is designed to provide an intuitive and seamless user experience. It is built using HTML, CSS, and JavaScript.

- **User Interface (UI) Components**

-Header: Contains the logo, navigation menu, and user account links.

- Footer: Includes contact information, social media links, and quick links to important sections.

-Home page: The main page of the website contains all features.

-Translate page: This page contains AI models to convert from text to sign and from sign to text.

-Education page: This page contains courses for adults and children to learn sign language and contain progress of the courses and have exercise and quiz.

-Support: we have a chatbot to ask any questions of the users.

- **Responsiveness**

-The website employs responsive design techniques to ensure a consistent user experience across various devices:

- Media Queries: CSS media queries adjust the layout for different screen sizes (e.g., mobile, tablet, desktop).

- Flexible Grid Layout: The grid system adapts to the screen size, ensuring content is appropriately displayed.

- Touch-Friendly Interactions: Buttons and interactive elements are designed for touch interfaces on mobile devices.

- **Interactivity**

- Forms: Include validation messages and tooltips for user inputs.

- Buttons: Provide immediate visual feedback on click or touch.

- **Accessibility**

  To make the website accessible to all users, including those with disabilities, the following features are implemented:

  - Keyboard Navigation: All interactive elements are accessible via keyboard.

  - Contrast and Font Size: High contrast colors and scalable fonts enhance readability.

- **Styling and Theming**

  - CSS Framework: The website uses Tailwind CSS for utility-first styling.

  - Custom Themes: Users can switch between light and dark themes using the toggle in the header.

- **Performance Optimization**

  - Code Splitting: Splits the JavaScript bundle for faster initial load times.

- **User Experience Considerations**

  - Intuitive Navigation: Clear, easy-to-use navigation menus.

  - Consistent Design: Uniform styling and layout across the website.

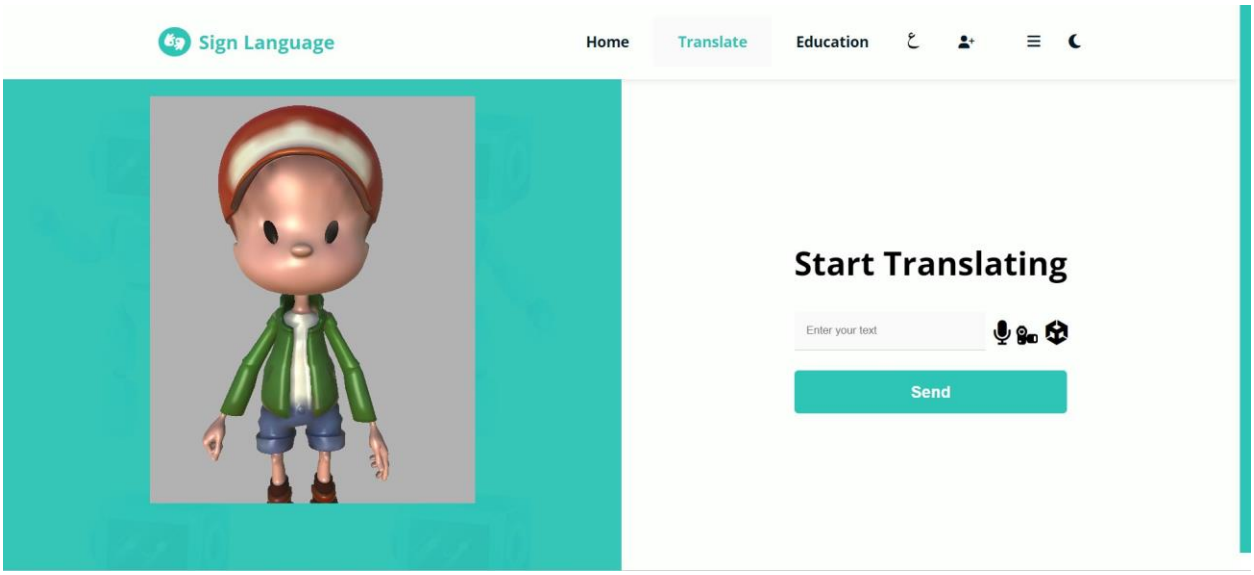
  - Feedback Mechanisms: Instant feedback for user actions such as form submissions and button clicks.

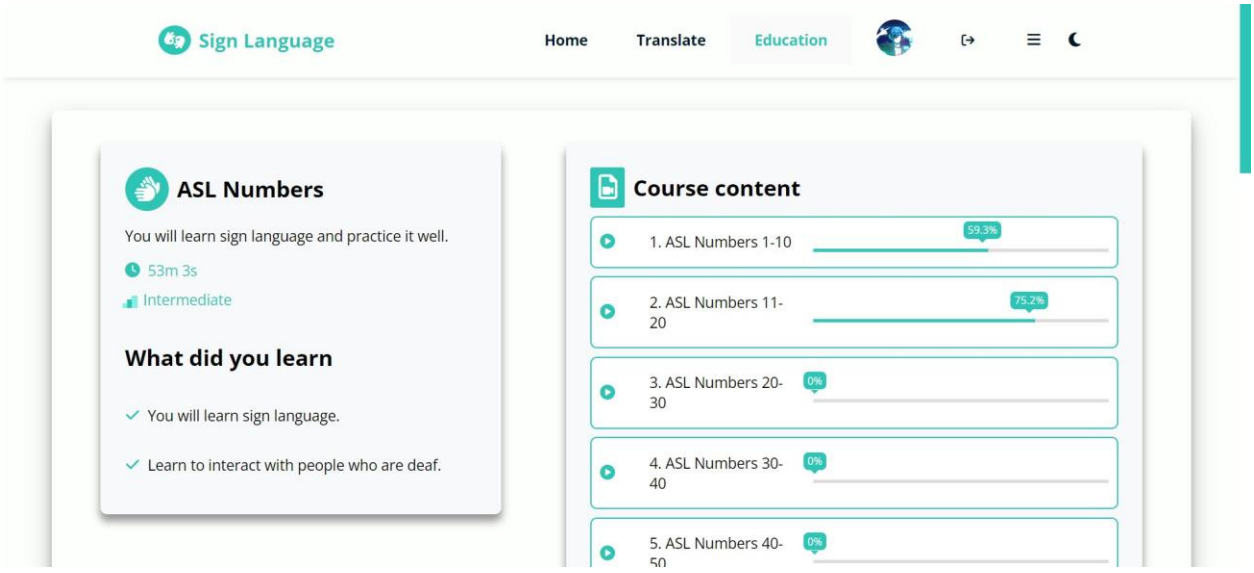
- **Troubleshooting**

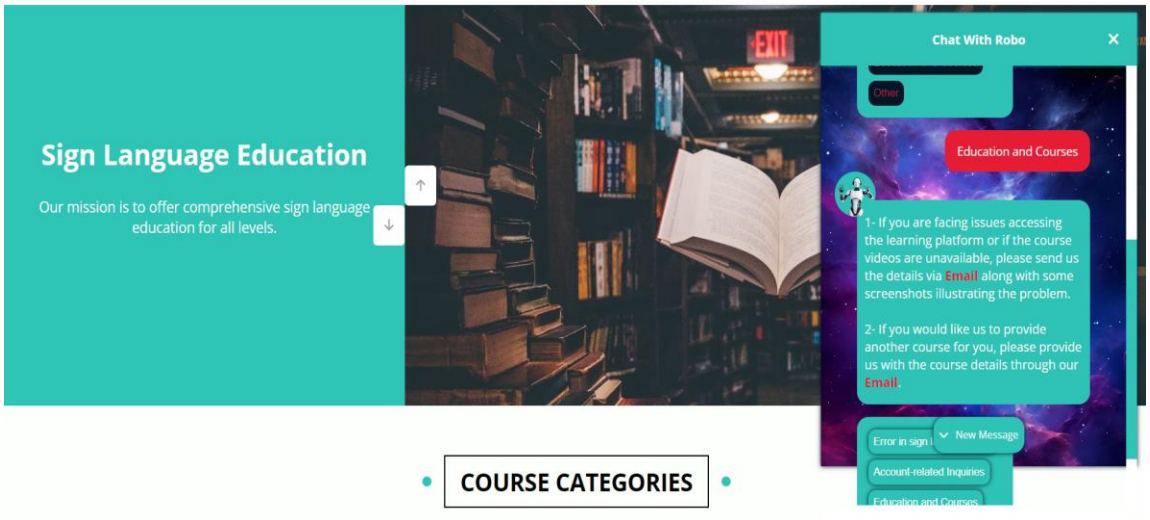
  - Layout Issues: Ensure the browser is up to date and try clearing the cache.

  - Interactivity Problems: Check if JavaScript is enabled in your browser.

  - Responsive Design Problems: Resize the browser window or switch to a different device to test responsiveness.


Sign Language
Home
Translate
Education
Start Translating
Enter your text
Send


Sign Language
Home
Translate
Education
ASL Numbers
You will learn sign language and practice it well.
53m 3s
Intermediate
What did you learn
You will learn sign language.
Learn to interact with people who are deaf.
Course content
1. ASL Numbers 1-10
59.3%
2. ASL Numbers 11-20
75.2%
3. ASL Numbers 20-30
0%
4. ASL Numbers 30-40
0%
5. ASL Numbers 40-50
0%


Sign Language Education
Our mission is to offer comprehensive sign language education for all levels.
EXIT
Chat With Robo
Other
Education and Courses
1- If you are facing issues accessing the learning platform or if the course videos are unavailable, please send us the details via Email along with some screenshots illustrating the problem.
2- If you would like us to provide another course for you, please provide us with the course details through our Email.
New Message
Account-related Inquiries
COURSE CATEGORIES

3.5.3 Back End

backend of the application, including its setup, endpoints, middleware, firebase setup and other crucial components.

Technologies Used:

- Node.js
- Express.js
- Firebase
- Nodemailer
- Redis
- Multer
- Axios
- EJS
- Express session
- livereload
- i18n

Configuration:

Environment Variables:

The application uses environment variables to manage configurations.

- PORT: The port on which the application runs.
- FIREBASE_CONFIG: Firebase configuration details.
- REDIS_HOST: Redis host address.
- REDIS_PORT: Redis port.
- REDIS_PASSWORD: Redis password.

Firebase Initialization:

The Firebase configuration is initialized in config.js. Ensure the Firebase SDK is properly set up and initialized.

Redis Initialization:

Redis is used for session management. Initialize Redis in the application and connect it using the provided credentials.

Endpoints:

User Endpoints:

- GET '/' Render the home page.
- GET '/translate' Render the translation page.
- GET '/unity' Render the Unity page.
- GET '/courses' Handles fetching courses based on the user's language preference then Render the courses page, redirects to sign-in if the user is not authenticated or sends an email verification if the user's email is not verified.
- GET '/courses/course_info' Fetches detailed information about a specific course then Render course information page, manages email verification and error handling if the course doesn't exist.
- GET '/courses/course_info/course_lessons' Manages lessons for a specific course and Handles progress tracking and initialization for course videos then Render course lessons, redirects to sign-in or sends email verification if necessary.
- GET '/courses/course_info/course_lessons/exercise' Renders an exercise page for authenticated users or redirects to sign-in otherwise
- GET' /courses/course_info/course_lessons/quiz' Renders a quiz page for authenticated users or redirects to sign-in otherwise.
- GET '/signIn' Render the sign-in page.
- GET '/signUp' Render the sign-up page.
- GET '/profile' Fetches user progress data for displayed courses then Render the profile page.
- GET '/reset' Render the reset page.
- GET '/reset/reset_password' Render the reset password page and Handles password reset with user ID.
- GET '/logout' Destroys the session and redirects to the main page.
- GET '/getProgress' Get progress course for user
- GET '/translations' GET method to fetch translations based on language, retrieves language from query parameters or defaults to English, uses i18n.getCatalog(lang) to get translations for the specified language then Sends translations as a JSON response to frontend

- POST ‘/signIn’ Authenticate the user and set session data than Redirects to the home page if the user is normal user or to the control page if the user is an admin Sets an error message in the session and redirects back to the signIn page if authentication fails.
- POST ‘/signUp’ Create a new user account authenticate and set initial user data to firestore Database, redirects to the signIn page upon successful user creation or sets an error message in the session and redirects back to the signIn page if sign-up fails (Email Exist).
- POST ‘/reset’ Sends a password reset email to the user, then Redirects to the home page after attempting to send the reset email.
- POST ‘/reset/reset_password’ Resets the user's password then Redirects to the signIn page upon successful password reset.
- POST ‘/Upload_Avatar’ Uploads a user's profile picture to Firebase Storage and updates the avatar URL in Firestore Storage then Redirects to the profile page after successful avatar upload.
- POST ‘/Update_User’ Updates the user's profile information in Firestore then Redirects to the profile page after successfully updating user information.
- POST ‘/Update_Password’ Updates the user's password after verifying the current password that user will enter then Redirects to the profile page upon successful password update or sets an error message in the session and redirects back to the profile page if password update fails.
- POST ‘/updateProgress’ Updates the progress of a user in a specific course in firestore then Returns a JSON response indicating success or failure of the progress update
- POST ‘/proxy-correct’ POST method to proxy data to an external API (pythonanywhere/correct_text_Model), Uses axios to send text to the external API to correct it Sends back the corrected text from the external API as a JSON response then send it to frontend, handles errors by sending a 500 status and error message if the request fails
- POST ‘/proxy-process’ **P**OST method to proxy data to another external API (ngrok/process_text_model), Uses axios to send text to the external API to rephrase the Arabic text Sends back the response data from the external API as a JSON response then send it to frontend, handles errors by sending a 500 status and a JSON object with an error message if the request fails
- POST ‘/uploadAudio’ Uploads an audio file to Speech Model (ngrok Api) by using axios then get the text form model and send back to front End.

- POST ‘/uploadVideo’ Uploads a video file to sign Language Model (ngrok Api) by using axios then get the text of the sign form model and send back to front End.

Admin Endpoints:

- GET ‘/control’ first Handles access denial and redirects to sign-in otherwise then enders admin control panel if authenticated as admin
- GET ‘/control/controlAccount’ first Handles access denial and redirects to sign-in otherwise then Render the control account page for admins.
- GET ‘/control/controlCoursesView’ first Handles access denial and redirects to sign-in otherwise then Render the control courses page for admins.
- GET ‘/play’ Renders a video player for a specific video for admin.
- POST ‘/setAdmin’ Set a user as admin by email.
- POST ‘/create-course’ Create a new course with code, language, name, duration, lesson, Lessondescription, categories, description, thumbnails, intro video, and playlist videos then Redirects to the admin control panel with success or error message.
- POST ‘/updateCourse’ Update details or files (thumbnails, intro video, playlist videos) of an existing course then Redirects to the admin control panel with success or error message.
- POST ‘/deleteCourse’ Delete a course from the database and associated files then Redirects to the admin control panel with success or error message.
- POST ‘/editVideo’ Edit or replace a video in a course's playlist then Redirects to the admin control panel with success or error message
- POST ‘/deleteVideo’ Delete a specific video from a course's playlist then Redirects to the admin control panel with success or error message.
- POST ‘/deleteUser’ Delete a user and associated data then Redirects to the admin control panel with success or error message.

Middleware:

- express.json(): Parse incoming requests with JSON payloads.
- cors(): Enable CORS for all routes.

- cookieParser(): Parse cookies attached to the client request object.
- i18n.init: Initialize internationalization.
- express.urlencoded(): Parse incoming requests with URL-encoded payloads.
- express.static(): Serve static files from the public directory.
- connect-livereload(): Enable live reload functionality.

Functions:

Email Functions:

- sendEmail(userEmail): Send an email verification link using Nodemailer.

Admin Functions:

- setAdmin(email): Set a user as an admin by email.

Firebase Functions:

- getCourses(lang): Retrieve courses based on the specified language.
- getUsers(): Retrieve all users and their details from Firebase.
- Various helper functions for generating messages based on the language.

Session Management:

Sessions are managed using express-session and stored in Redis. The session configuration includes secure cookie settings and a session store connected to Redis.

Language Preference Middleware:

This middleware sets the language preference for each user based on query parameters, cookies, or a default value. It ensures that the application responds in the user's preferred language and sets a cookie to remember this preference.

Live Reload:

Live reload functionality is enabled using livereload and connect-livereload This allows the application to automatically refresh the browser when changes are detected in the public directory
Note: Live reload functionality is only used during local development and should not be enabled in production

3.6 Mobile Application:

3.6.1 Introduction:

Sign language is a vital mode of communication for individuals who are deaf or hard of hearing. However, the language barrier between sign language users and those who do not understand it often leads to communication challenges. To bridge this gap, we are developing a sign language translator mobile application. This mobile app with Kotlin language with smoothy UI uses advanced technologies like avatar with unity and firebase storage to translate sign language gestures into spoken or written language and vice versa, facilitating effective communication directly on users' smartphones.

3.6.2 Purpose

The primary purpose of the sign language translator mobile application is to:

1. Enable seamless communication between sign language users and non-sign language users.
2. Promote inclusivity and accessibility in various social, educational, and professional environments.
3. Leverage mobile technology to provide a portable and user-friendly solution for real-time translation.

3.6.3 Key Features

1. **Real-Time Translation**:
    - The application uses the camera of the mobile device to capture sign language gestures and translates them into text in real-time.
2. **Voice-to-Sign Translation:**
    - It converts spoken language into corresponding sign language animations, helping sign language users understand spoken words.
3. **Text-to-Sign Translation:**
    - Users can type text, which the application translates into sign language through animations (Avatar).
4. **Learning Mode:**
    - The application includes a mode for learning and practicing sign language, featuring tutorials and interactive lessons.
5. **Customization:**
    - Users can customize the language preferences, including different dialects and regional variations of sign language.
6. **Offline Mode:**
    - The application offers basic translation functionalities without requiring an internet connection, ensuring accessibility even in areas with poor connectivity.

- **Technologies Used**

1. **Kotlin Language:**
    - In android studio as IDE using Kotlin as our programming language.
2. **Camera X:**
    - Utilizes the device's camera and computer vision algorithms to detect and recognize hand gestures.
3. **UnityPlayer apk**:
    - That helps to connect avatar which created on unity, with our mobile application.
4. **Machine Learning Models:**
    - Connecting the models (speech, prediction or classification model) helps to make this mobile application interactive.
5. **Firebase**

3.6.4 Development Challenges

1. **Gesture Recognition Accuracy:**
    - Ensuring high accuracy in recognizing diverse and complex gestures across different users within the constraints of mobile device hardware and varying lighting conditions.
2. **Real-Time Processing:**
    - Achieving low-latency translation to enable fluid and natural communication, considering the limited processing power and memory of mobile devices.
3. **User Interface (UI) Design**:
    - Designing an intuitive and accessible UI that caters to both sign language users and non-sign language users.

3.6.5 Impact and Benefits

1. **Enhanced Communication:**
    - Breaks down communication barriers, allowing deaf and hard of hearing individuals to interact more freely with the hearing community.
2. **Educational Tool:**
    - Serves as an educational resource for both sign language learners and educators.
3. **Inclusivity:**
    - Promotes inclusivity in workplaces, public services, and social settings by making communication more accessible.
4. **Social Integration:**
    - Helps in the social integration of sign language users by facilitating smoother interactions in daily life.

3.6.6 Conclusion

The sign language translator mobile application represents a significant advancement in assistive technology, leveraging modern mobile and AI technologies to foster a more inclusive society. By providing a reliable and user-friendly platform for sign language translation, the application aims to bridge the communication gap and enhance the quality of life for sign language users.

3.6.7 Implementation View

After signing up or log in this is the view of translate page and education page, translate page: which avatar exist and input a text, picture or voice and avatar will help you to know the sign word.
education page: which courses is there to help learning the sign language for kids and adults.

Figure 30 Android Translation page

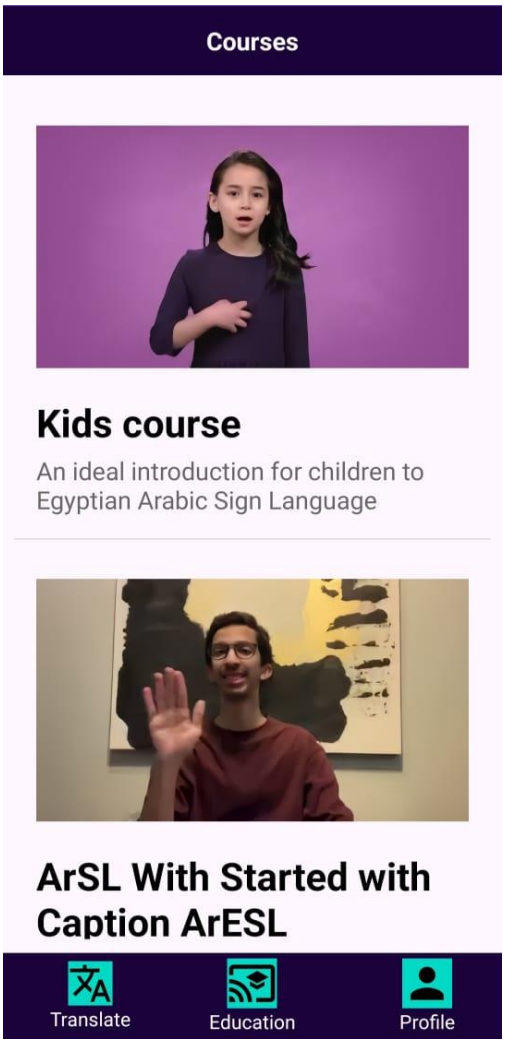


Figure 29 Android Education page

4 DISCUSSION AND CONCLUSION

4.2 Introduction

Deaf individuals face numerous challenges in communicating with others, leading to difficulties in social interaction and understanding daily information. In today's busy world, it is challenging to provide a constant interpreter for them as everyone is occupied with their own responsibilities. Enhancing communication between people is very important to us, so how can we facilitate understanding between deaf individuals and hearing people? Hence, we conceived the idea of developing an application to address the communication challenges faced by the deaf and to simplify interactions between them and others through artificial intelligence, which will serve as a translator between them. This application offers its service through both a web platform and an Android app, allowing users to interact easily and conveniently anytime and anywhere. The system translates sign language into text and speech, and vice versa, ensuring an effective and seamless communication experience for all parties involved

4.3 Main Findings

4.3.3 Why Is This Project Important

The importance of this system lies in facilitating communication between deaf individuals and hearing people. Deaf individuals face significant challenges in interacting with others, leading to difficulties in social communication and understanding daily information. The system translates sign language into text and speech, and vice versa, ensuring an effective and seamless communication experience for all parties involved. The importance of this system includes:

1. Improving understanding between deaf individuals and hearing people.

2. Providing a quick and effective means of communication that helps reduce social barriers.Additionally, the application offers the ability to learn sign language through the web platform and Android app, enabling users to enhance their sign language skills and communicate better with deaf individuals. This way, I can interact effectively with deaf friends and colleagues while continuously learning sign language anytime and anywhere

4.3.4 Practical Implementations

- Data Collection and Preparation :
  Video data from diverse locations (bank, café, hospital, train station) yielded 30,000 clips with 307 words each. After deduplication, a dataset of 200 unique words was finalized.
- Modeling with OpenHands:
  Utilized OpenHands for pose-based sign language recognition across six languages. Evaluated LSTM, BERT, ST-GCN, and SL-GCN models in PyTorch Lightning, optimizing with Adam.
- Text Processing with Octopus:
  Employed Octopus for NLP tasks: diacritical marks, grammar correction, title generation, paraphrasing, question answering, and translation. Configured for quality using beam search.
- Audio Transcription with SpeechRecognition:
  Used SpeechRecognition for ".wav" files, integrating Google's API for accurate Arabic speech-to-text conversion.
- Avatar Development:
  Created an avatar in Unity via Mixamo.com, overcoming animation challenges for sign language words through manual creation.
- System Integration and Deployment:
  Integrated components into ArabicSL-Net for robust sign language recognition, tested and optimized for real-world efficiency and compatibility.
- Web and Mobile Application Development:
  - An interactive user interface was developed using HTML, CSS, and JavaScript, featuring key components such as the header, footer, and various pages. Responsive design techniques were incorporated to ensure a consistent user experience across different devices. Technologies like Node.js, Express.js, Firebase, and Redis were used to build the backend, with environment variables configured and integration with Firebase and Redis for secure session management. The avatar was integrated to enhance user experience.
  - For the mobile application, it was developed using Android Studio and Kotlin. Camera X and UnityPlayer APK were integrated to utilize the mobile device’s camera for gesture recognition and avatar animation, with real-time translation implemented using Kotlin and Unity.

- The integration of web and mobile applications enhances accessibility and ensures a seamless user experience across all devices, allowing users to easily interact with the application's content.

4.4 Limitations

- Challenges in gathering video data from diverse sources such as banks, cafes, and hospitals, requiring significant resources to ensure data diversity and quality representation.

- Complexity in developing accurate models for efficient sign language recognition across different languages, necessitating high computational power for model training and optimization.

- Difficulties in text processing for translation and question answering, with challenges in adapting the system to linguistic and cultural variations.

- Challenges in achieving accurate Arabic speech-to-text conversion in noisy environments or diverse dialects using external services, exacerbating connectivity and cost issues.

- Complications in creating precise animations for sign language words and gestures, with challenges in delivering a flexible and engaging user experience.

- Integration complexities of multiple system components and ensuring their effective performance, requiring rigorous testing to ensure optimal responsiveness in real-world environments.

- Challenges in providing a consistent user experience across all devices and platforms, alongside additional complexities related to supporting and maintaining multiple technologies.

- High costs associated with system development and maintenance, including software, hardware, and human resources expenses.

- Challenges in persuading users to adopt the system and build confidence in its accuracy and efficiency, necessitating effective strategies for user education and training.

4.5 Future Recommendation

- Expand Use in Educational Institutions: The system can be integrated into schools and universities to teach students sign language, enhancing cultural understanding and interaction between deaf and hearing students.

- Offer Community Sign Language Courses: Organize public courses to teach sign language to the general community, increasing awareness and respect for the needs of deaf individuals in society.

- Integrate Technology with Education: Develop innovative applications that integrate technology with learning and teaching sign language, making the process more engaging and effective.

- Enhance Technical Support and Training: Strengthen technical support for users and provide regular training sessions for teachers and staff interacting with deaf individuals, ensuring effective use of the system that meets everyone's needs.

- Collaborate with Medical and Healthcare Community: Foster collaboration with hospitals and medical centers to provide medical support for deaf individuals relying on the system, ensuring prompt and appropriate responses to their health needs.

4.6 Summary

The project idea revolves around translating between sign language and text/audio, bridging the communication gap between hearing individuals and the deaf. Often, communication breakdowns occur where neither party can understand each other, necessitating urgent communication with the deaf community.

Additionally, the system educates in sign language and provides tests and exercises for self-assessment. It offers a web and Android application to facilitate user interaction with the system.

Ultimately, this system aims to eliminate communication barriers and enhance skills for regular individuals to communicate effectively with the deaf. Now, the deaf can feel included and communicate seamlessly with hearing individuals.

5 References:

[1] Bryan M. O'Halloran, Robert B. Stone, Irem Y. Tumer, "A failure modes and mechanisms naming taxonomy", 2012 Proceedings Annual Reliability and Maintainability Symposium, pp.1-6, 2012.

[2] M. Essam, “ArabicSL-Net: A Benchmark Video Dataset for Arabic Words Sign Language”. Zenodo, Jan. 01, 2023. doi: 10.5281/zenodo.7771372.

[3] P. Selvaraj, G. NC, P. Kumar, and M. Khapra, "OpenHands: Making Sign Language Recognition Accessible with Pose-based Pretrained Models across Languages," arXiv:2110.05877, 2021.

[4] G. NC, M. Ladi, S. Negi, P. Selvaraj, P. Kumar, and M. M. Khapra, "Addressing Resource Scarcity across Sign Languages with Multilingual Pretraining and Unified-Vocabulary Datasets," in Thirty-sixth Conference on Neural Information Processing Systems Datasets and Benchmarks Track, 2022. Available: https://openreview.net/forum?id=zBBmV-i84Go

[5] N. Ibrahim, H. Zayed, and M. Selim, "Advances, Challenges, and Opportunities in Continuous Sign Language Recognition," Journal of Engineering and Applied Sciences, vol. 15, pp. 1205-1227, 2019, doi: 10.36478/jeasci.2020.1205.1227.

[6] T. Starner and A. Pentland, "Visual Recognition of American Sign Language Using Hidden Markov Models," Perceptual Computing Section, The Media Laboratory, Massachusetts Institute of Technology.

[7] A. Moryossef, M. Müller, and R. Fahrni, "pose-format: Library for viewing, augmenting, and handling .pose files," 2021. [Online]. Available: https://github.com/sign-language-processing/pose

[8] Neto, G.M.R., Junior, G.B., de Almeida, J.D.S., de Paiva, A.C. (2018). Sign Language Recognition Based on 3D Convolutional Neural Networks. In: Campilho, A., Karray, F., ter Haar Romeny, B. (eds) Image Analysis and Recognition. ICIAR 2018. Lecture Notes in Computer Science(), vol 10882. Springer, Cham. https://doi.org/10.1007/978-3-319-93000-8_45

[9] P. Neto, D. Pereira, J. N. Pires, Member, IEEE, and A. P. Moreira, Member, IEEE, "Real-Time and Continuous Hand Gesture Spotting: an Approach Based on Artificial Neural Networks."

[10] Zhao, ZC., Cai, AN. (2006). Shot Boundary Detection Algorithm in Compressed Domain Based on Adaboost and Fuzzy Theory. In: Jiao, L., Wang, L., Gao, X., Liu, J., Wu, F. (eds) Advances in Natural Computation. ICNC 2006. Lecture Notes in Computer Science, vol 4222. Springer, Berlin, Heidelberg. https://doi.org/10.1007/11881223_76

[11] Choudhury, Ananya, Kumar Talukdar, Anjan, Kamal Bhuyan, Manas and Kumar Sarma, Kandarpa. "Movement Epenthesis Detection for Continuous Sign Language Recognition" *Journal of Intelligent Systems*, vol. 26, no. 3, 2017, pp. 471-481. https://doi.org/10.1515/jisys-2016-0009

[12] S. Sarkar, R. Yang, and B. Loeding, "Enhanced Level Building Algorithm for the Movement Epenthesis Problem in Sign Language Recognition," in *2007 IEEE Conference on Computer Vision and Pattern Recognition*, Minneapolis, MN, 2007, pp. 1-8. doi: 10.1109/CVPR.2007.383347.