National University Of Computing & Emerging Sciences

Final Year Project (CS-491)

Software Design Specification (SDS) Document

Fall '23, BS(CS)

CogniAssess

Team Candidates

Sumsam Ali - 20K-1075

Mukand Krishna - 20k-0409

Bahadur Khan - 20k-1081

Supervisor

Dr. Nouman Durrani

Project Code

F23-109D

From the Department Of Computer Science

Faculty of Computer Sciences and Engineering

National University Of Computing And Emerging Sciences

Document History

Revision history will be maintained to keep track of changes done by anyone in the document.

Version	Name of Person	Date	Description of change
1.0	Sumsam Ali	2023-11-01	Document Created
1.1	Sumsam Ali	2023-11-23	Added Class Diagrams
1.2	Sumsam Ali	2023-12-05	Added State Diagrams
1.3	Sumsam Ali	2023-12-05	Added Sequence Diagrams

Distribution List

The following table will contain a list of people to whom the document will be distributed after every sign-off

Name	Role
Name of Supervisor	Dr. Nouman Durrani
Name of Co-Supervisor	None

Document Sign-Off

The following table will contain sign-off details of the document. Once the document is prepared and revised, this should be signed off by the sign-off authority. Any subsequent changes in the document after the first sign-off should again get a formal sign-off by the authorities.

Version	Sign-off Authority	Sign-off Date
1.0	Dr. Nouman Durrani	12-10-23

Document Information

Category	Information
Customer	FAST-NU
Project	CogniAssess
Document	Software Design Specification
Document Version	1.4
Status	Draft
Author(s)	Sumsam Ali , Mukand Krishna and Bahadur Khan
Approver(s)	
Issue Date	
Document Location	
Distribution	
Advisor	
Project Coordinator's Office (through Advisor)	

Definition of Terms, Acronyms, and Abbreviations

This section should provide the definitions of all terms, acronyms, and abbreviations required to interpret the terms used in the document properly.

Term	**Definition**
ASP	Active Server Pages
DD	Design Specification

Contents

1 Introduction

1.1 Purpose of Document

This Software Design Specification (SDS) document is crafted to provide a comprehensive blueprint for the development of COGNIASSESS, an innovative platform tailored for the assessment and recruitment of candidates in non-technical roles. The document's primary purpose is to guide the development team through a structured approach in building the software, ensuring that all functionalities and requirements are met with precision. Additionally, it serves as a reference point for stakeholders to understand the design process and architectural decisions. The project will employ an Object-Oriented Design Methodology, which is well-suited for handling the complex, modular, and scalable nature of the COGNIASSESS system.

1.2 Intended Audience

The intended audience for this SDS document encompasses a broad spectrum of stakeholders including Supervisors, Jury and students. This document aims to provide these professionals with a clear understanding of the design principles, system architecture, and functional components of COGNIASSESS, facilitating effective collaboration and alignment across different phases of the development process.

1.3 Document Convention

This document adheres to a standardized format for consistency and clarity. The primary text is set in Times New Roman, with a font size of 12 points, ensuring readability and professionalism. Headings are formatted in bold with a size of 14 points to distinguish them clearly from the main text. Key technical terms, code snippets, and important notes are highlighted in italics to draw attention and aid in quick reference.

1.4 Project Overview

COGNIASSESS is envisioned as a state-of-the-art platform designed to revolutionize the recruitment process for non-technical roles. It integrates advanced technologies such as AI-driven large language models (LLMs) and AI proctoring to evaluate and rank candidates based on their non-technical skills. The system will offer a seamless, intuitive interface for candidates to register, upload their CVs, choose desired roles, and undergo personalized assessments. For recruiters, the platform will provide robust tools for candidate search, filtering, and selection based on comprehensive assessment data. The overarching design goal is to create a secure, efficient, and user-centric solution that enhances the recruitment experience for both candidates and recruiters[1].

1.5 Scope

The COGNIASSESS project's scope is designed to create a streamlined, efficient platform for non-technical skill assessment in recruitment[1]. Key inclusions:

- Web-based platform development for non-technical skill assessments.
- AI integration for dynamic assessments and secure proctoring.
- User-friendly interfaces for candidate profile creation, CV upload, and role selection and Feedback.
- Diverse assessment modules and personalized feedback mechanisms.
- Adherence to data privacy and security standards.
- Ongoing system updates and user experience enhancements.

Exclusions from the scope:

- Assessment of technical skills or roles.
- Real-time video interviewing and advanced technical analysis tools in the initial launch.
- Job posting or application functionalities.
- Integration with external job boards or recruitment platforms in the initial version.
- Multilingual support beyond English in the initial release.

2 Design Considerations

This section explores critical issues to be addressed or resolved before establishing a complete design solution for the COGNIASSESS project. It aims to set a solid foundation for the system design, taking into account various influencing factors.

2.1 Assumptions and Dependencies

This section highlights new issues related to design, distinct from those captured in the System Requirements Specification (SRS) document. Key points include:

- The assumption that the selected technology stack, including the fine-tuned LLM, will remain stable, supported, and available throughout the development cycle.
- Dependency on external services for AI proctoring and integration of the fine-tuned LLM, assuming consistent access and performance levels.
- Expectation of users interacting with the platform in a manner consistent with the predefined user journey, particularly in relation to interactions with the LLM-driven assessment tools.
- Reliance on robust and scalable cloud infrastructure for hosting the platform, which is critical for the high-performance demands of the LLM.
- Presumption of continuous internet connectivity, essential for the LLM's real-time processing and data analysis capabilities.

2.2 Risks and Volatile Areas

This subsection addresses potential sources of change and risks that could impact the system's design, with a focus on the fine-tuned LLM:

- Dependence on the LLM provider's stability and service continuity, with potential impacts on system functionality and user experience.
- Contingency measures include a modular design approach to facilitate LLM updates, regular reviews of AI technology trends, and flexible data policies to manage privacy and regulatory compliance.

3 System Architecture

This section provides a high-level overview of the system's architecture, delineating how the functionalities and responsibilities are partitioned and then allocated to various subsystems or components. It aims to offer a clear understanding of the system's decomposition and the interplay between individual parts to deliver the required functionality.

3.1 System Level Architecture

The system is decomposed into several key elements, each responsible for a specific set of functions. The following points outline the main elements of the system:

- **User Registration and Verification Module:** Handles user registration and verifies their status upon each visit to the website.
- **Role and Domain Selection Module:** Allows users to select their non-technical roles and relevant domains for assessment.
- **CV Analysis and Question Generation Engine:** Analyzes the uploaded CVs and prompts the Large Language Model (LLM) to generate tailored assessment questions.
- **Personality Assessment Module:** Utilizes the LLM to generate questions for evaluating personality traits.
- **Job Scenario Assessment Module:** Generates sample job scenario questions based on the candidate's past experiences.
- **Evaluation and Feedback Engine:** Evaluates user responses and generates personalized feedback, contributing to the user's overall ranking on the platform.
- **Recruiter Interface:** Allows recruiters to access candidate rankings and connect with potential candidates.

The relationship between these elements is orchestrated through a series of interconnected workflows, ensuring a seamless user experience from registration to assessment completion.

Interfaces to External Systems: The system interfaces with external databases for storing user data and CVs, and integrates with external APIs for additional functionalities like email notifications.

Physical Design Considerations: The system is designed to be cloud-based, ensuring scalability and reliability.

Global Design Strategies: The system employs robust error handling and security measures to ensure data integrity and user privacy.

3.2 Software Architecture

The software architecture of the system is layered to separate concerns and enhance maintainability. It consists of the following layers:

- **User Interface Layer:** The front-end of the system, designed for user interaction. It includes the web pages for registration, role and domain selection, CV upload, and assessment modules.
- **Middle Tier:** This layer hosts the business logic and controls the application flow. It acts as a bridge between the user interface and the data access layers.
- **Data Access Layer:** Responsible for data persistence. It handles all interactions with the database, ensuring data is stored and retrieved efficiently.

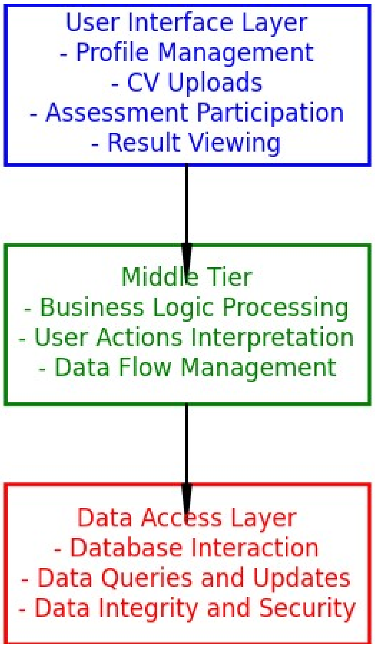

Figure 1: Software Architectural Diagram

Additional Architectural Considerations: Depending on the specific requirements and technologies used, other architectural elements such as service-oriented architecture (SOA), microservices, or a message-oriented middleware might be incorporated into the system design.

4 Design Strategy

This section delves into the design strategies or decisions that significantly impact the overall organization and high-level structure of the system. These strategies provide insights into the key abstractions and mechanisms employed in the system's architecture. Each strategy is chosen based on previously stated design goals and principles, with considerations for potential trade-offs.

4.1 Future System Extension or Enhancement

The system is designed with scalability and flexibility in mind to accommodate future extensions or enhancements. Modular design and loose coupling are emphasized to ensure that adding new features or updating existing ones does not require a significant overhaul of the system.

- **Modular Architecture:** The system's architecture is compartmentalized into distinct modules, each handling specific functionalities. This approach facilitates easy integration of new modules or enhancements to existing modules without disrupting the entire system.
- **Use of APIs:** External functionalities are integrated through APIs, allowing for easy updates and the addition of new features.

4.2 System Reuse

The design incorporates principles that promote reusability, aiming to maximize the efficiency and cost-effectiveness of the system development.

- **Standardized Interfaces:** Interfaces between modules are standardized, facilitating the reuse of components in different contexts or projects.
- **Code Reusability:** Commonly used functions and classes are abstracted into reusable libraries, reducing duplication and simplifying maintenance.

4.3 User Interface Paradigms

The user interface (UI) is designed to be intuitive and user-friendly, adhering to established UI paradigms to ensure a seamless user experience.

- **Responsive Design:** The UI is responsive, making the system accessible across various devices and screen sizes.
- **Consistent Design Language:** A consistent design language is used throughout the system to provide a cohesive user experience.

4.4 Data Management

Data management strategies focus on storage, distribution, and persistence to ensure efficient and secure handling of data.

- **Database Optimization:** Databases are optimized for performance and scalability.
- **Data Distribution:** Data is distributed across multiple servers to balance the load and ensure high availability.
- **Data Persistence:** Persistent storage mechanisms are employed to ensure data integrity and durability.

4.5 Concurrency and Synchronization

Given the multi-user nature of the system, concurrency and synchronization are critical aspects of the design.

- **Concurrency Control:** Mechanisms are in place to handle concurrent accesses and modifications, ensuring data consistency.
- **Synchronization:** Synchronization techniques are employed to maintain the integrity of data across different modules and layers of the system.

These design strategies collectively contribute to a robust, scalable, and user-friendly system, aligning with the overarching goals and principles initially set for the project.

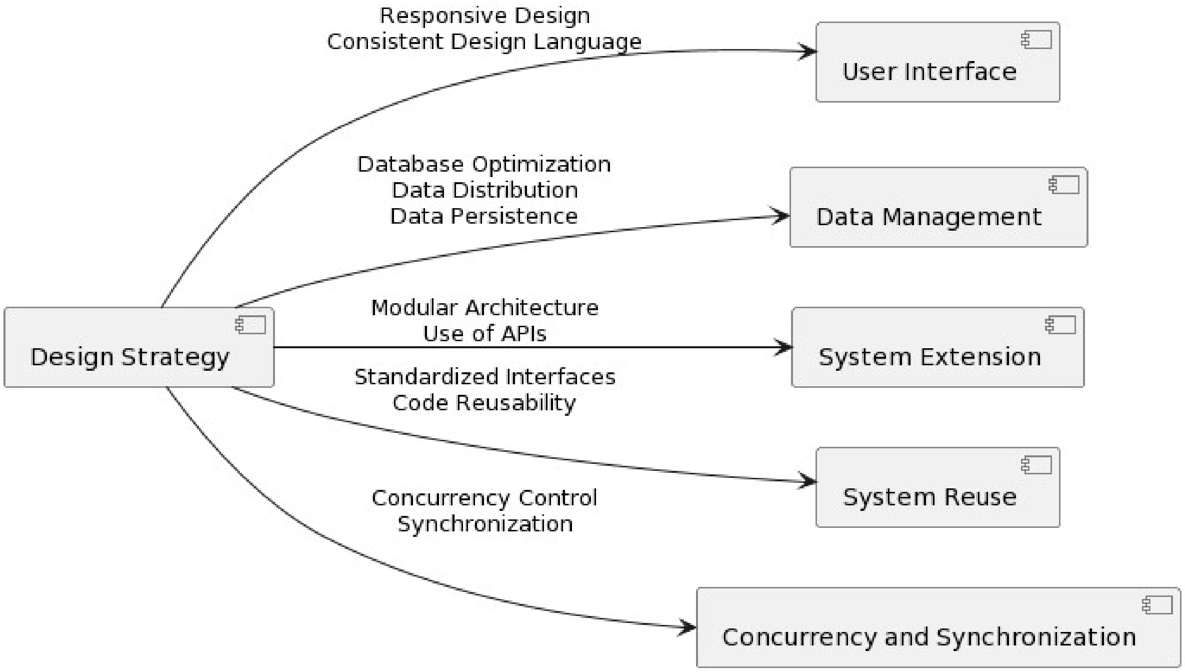

Figure 2: Design Strategy

5 Detailed System Design

This section provides a comprehensive overview of the COGNIASSESS system, detailing the class diagrams, logical data models (E/R model), and the design of the Graphical User Interfaces (GUIs)[2].

5.1 Class Diagram and Descriptions

The class diagram of COGNIASSESS consists of several integral classes, each encapsulating specific attributes and methods necessary for the system's functionality. The key classes include User, Role, Domain, CV, Question, Assessment, Ranking, and Recruiter.

5.1.1 User Class

The User class is central to the system, representing the end-users. Attributes include userID, name, email, password, registeredStatus, userRole, among others. Key methods encompass login, register, updateProfile, selectRole, selectDomain, uploadCV, startAssessment, completeAssessment, getFeedback, viewAssessmentHistory, calculateCurrentRank, and logout. This class interacts with almost all other classes, playing a pivotal role in system operations.

5.1.2 Role Class

The Role class defines the various roles a user can assume within the system (e.g., candidate, recruiter). It includes attributes like roleID and roleName, and methods for role management such as addRole and removeRole.

5.1.3 Domain Class

This class represents different domains or specializations that users can select. Attributes include domainID and domainName. Methods like addDomain and removeDomain facilitate dynamic domain management.

5.1.4 CV Class

The CV class handles the user's curriculum vitae, storing and processing it for further analysis. Key attributes are cvID, userID, and cvContent. It includes methods for uploading and analyzing the CV.

5.1.5 Question Class

This class is responsible for generating and managing questions for the assessments. Includes questionID, domainID, content, and type as attributes. Methods like generate and display are crucial for assessment functionality.

5.1.6 Assessment Class

The Assessment class deals with the evaluation of users based on their responses. Attributes include assessmentID, userID, questionIDs, responses, and score. Evaluate and generateFeedback are key methods.

5.1.7 Ranking Class

This class manages the ranking of users based on their assessment scores. Attributes are rankingID, userID, and rankScore. The calculate method computes the user's rank.

5.1.8 Recruiter Class

Specific to the recruiter's role, this class provides functionalities for searching and reviewing candidate profiles. It has recruiterID, userID, and includes methods like searchCandidates and viewCandidateProfile.

5.1.9 Streaks Class

The Streaks class is designed to track and incentivize user engagement over consecutive periods. Encouraging regular use of the system. Attributes include streakID, currentStreak, and longestStreak.

5.1.10 Points Class

The Points class is a gamification element that rewards users for completing assessments. It includes attributes like userID, and totalPoints. This class typically interacts with multiple other classes, enhancing user engagement and satisfaction.

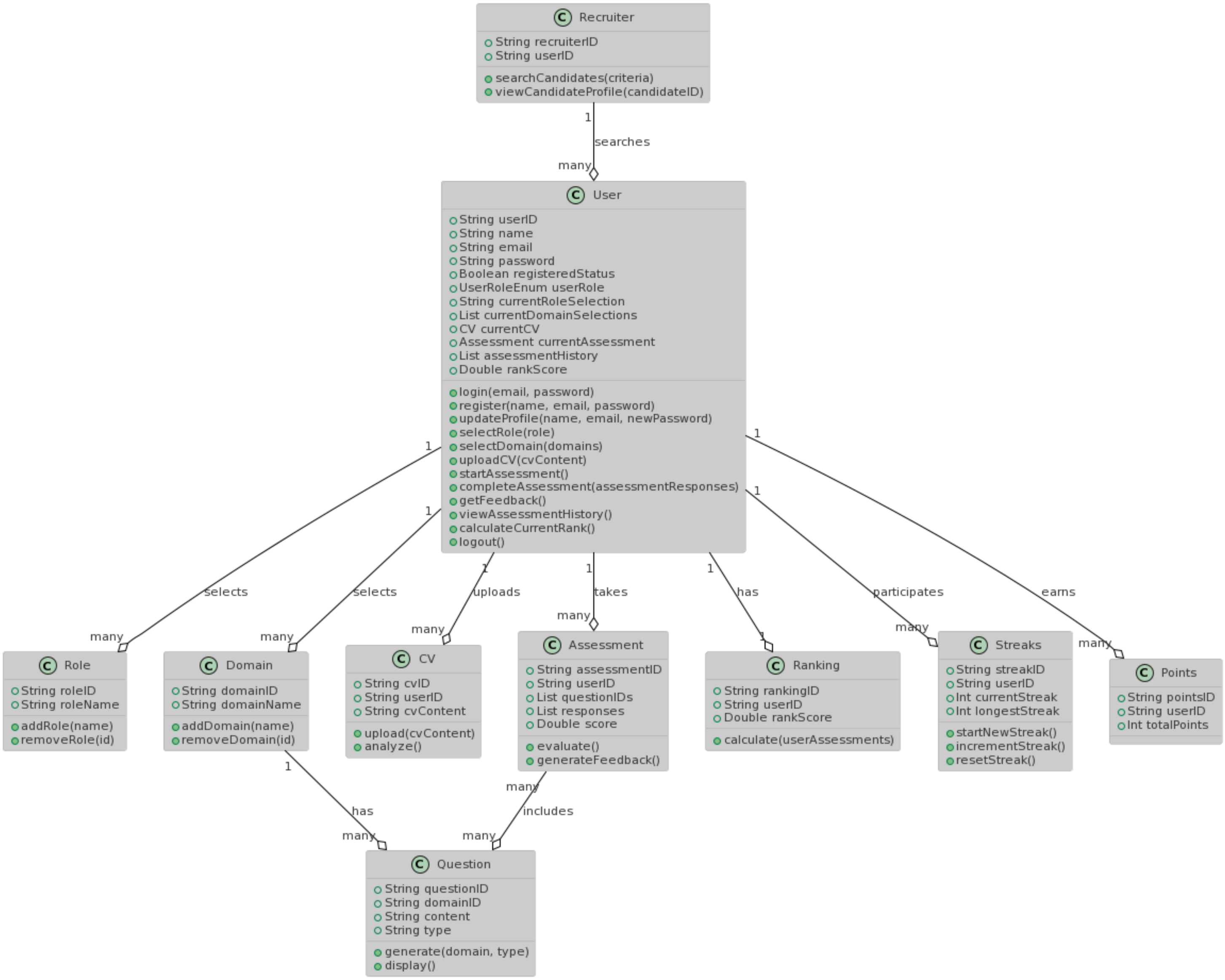

Figure 3: Class Diagram

5.2 Logical Data Model (E/R Model)

The E/R model of COGNIASSESS illustrates the relationships between different entities such as users, roles, domains, CVs, questions, assessments, and rankings. - Users can have multiple roles and select multiple domains. - Each domain can relate to multiple questions, which in turn are part of various assessments. - Users have a one-to-one relationship with their rankings. - The Recruiter entity interacts with the User entity for candidate search and profile viewing.

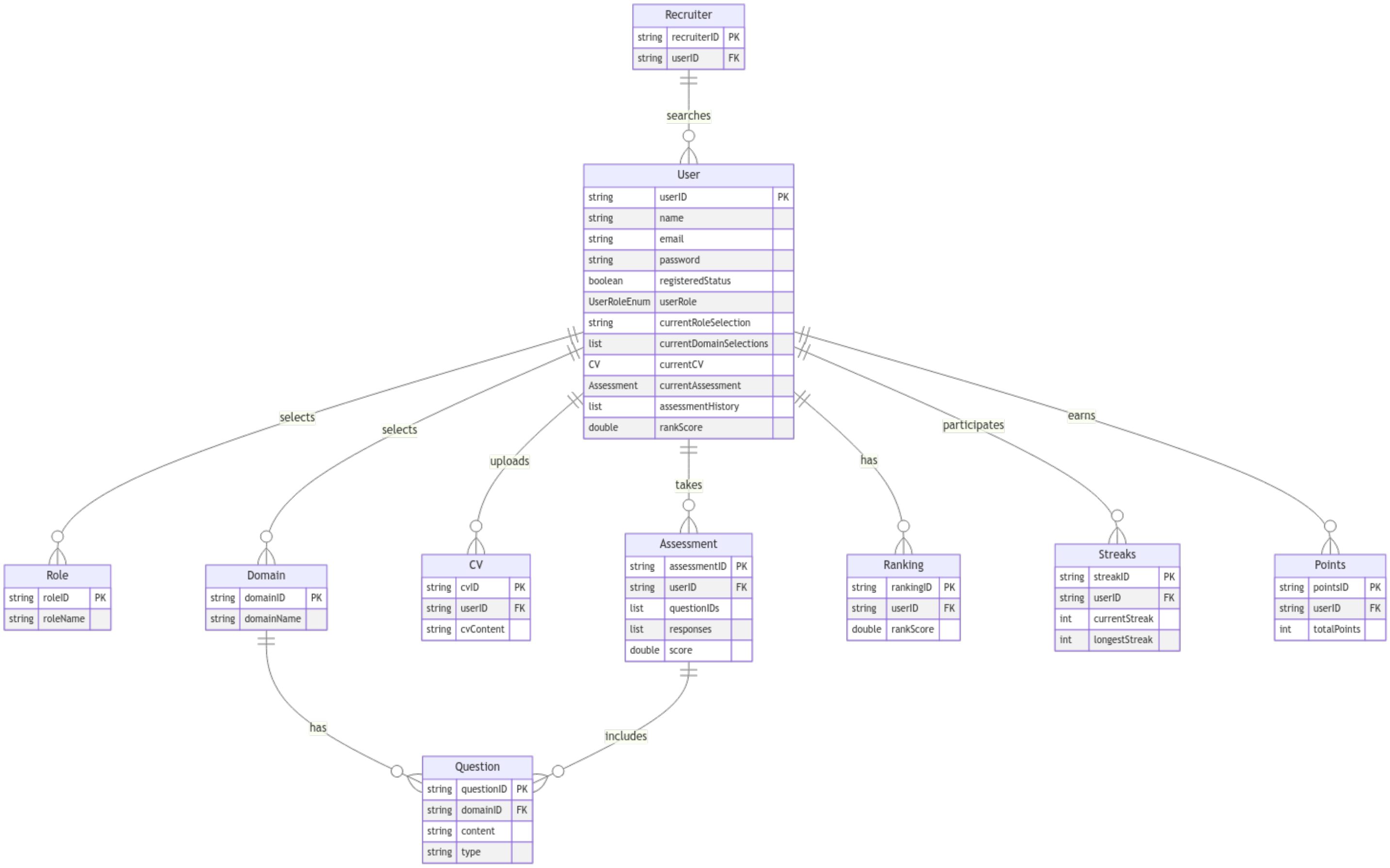

Figure 4: ER Diagram

5.3 Detailed GUIs

The GUI of COGNIASSESS is designed to offer a user-friendly and intuitive interaction for both candidates and recruiters. - **Registration and Login Screens:** Facilitate user access to the system. - **Role and Domain Selection Interfaces:** Allow users to choose roles and domains for personalized assessments. - **CV Upload and Analysis:** Users can upload and manage their CVs. - **Assessment Interface:** Where users engage with the assessment questions. - **Feedback and Ranking Screens:** Display personalized feedback and user rankings post-assessment. - **Recruiter Dashboard:** Enables recruiters to search, view, and manage candidate profiles.

These GUIs ensure a seamless and efficient user experience, crucial for the system's effectiveness. Please click the following link to access the GUI: Figma UI link

6 Data Dictionary

6.1 User Class

User	
Name	User
Alias	Account, Member, Participant
Where-used/how-used	The User class is used for user authentication, storing personal and login information, managing roles, and tracking assessment interactions within the system.
Content description	This class contains information about the users, including credentials, profile details, roles, and assessment data.

Column Details						
Column Name	**Description**	**Type**	**Length**	**Nullable**	**Default Value**	**Key Type**
userID	Unique identifier for the user	Integer	10	No	None	Primary Key
name	Full name of the user	Varchar	255	No	None	
email	Email address used for login	Varchar	255	No	None	Unique
password	Hashed password for user authentication	Varchar	255	No	None	
registeredStatus	Status of user's completion of the registration process	Boolean		No	False	
userRole	Role of the user in the system (e.g., Candidate, Recruiter)	Enum		No	Candidate	
currentRoleSelection	Currently selected role for assessment	Varchar	255	Yes	None	
currentDomainSelections	List of selected domains for the user's assessment	Text		Yes	None	
currentCV	Reference to the user's uploaded CV	Text		Yes	None	
currentAssessment	Reference to the ongoing assessment instance for the user	Text		Yes	None	
assessmentHistory	List of past assessments taken by the user	Text		Yes	None	

rankScore	Numeric score representing the user's ranking	Decimal	5,2	Yes	0.00	

6.2 Role Class

Role	
Name	Role
Alias	UserRole, Position
Where-used/how-used	The Role class is used to assign and manage roles for users within the system, impacting access control and available actions.
Content description	This class represents the roles that can be assigned to users, such as Candidate or Recruiter.

Column Details						
Column Name	**Description**	**Type**	**Length**	**Nullable**	**Default Value**	**Key Type**
roleID	Unique identifier for the role	Integer	10	No	None	Primary Key
roleName	Name of the role	Varchar	100	No	None	

6.3 Domain Class

Domain	
Name	Domain
Alias	ExpertiseArea, Field, Specialization
Where-used/how-used	This class is utilized to categorize users and questions based on their area of expertise or interest. It is used in the assessment process to ensure that users are evaluated within relevant domains.
Content description	Contains the details of the various domains or categories that can be selected by users for assessments.

Column Details						
Column Name	**Description**	**Type**	**Length**	**Nullable**	**Default Value**	**Key Type**
domainID	Unique identifier for the domain	Integer		No	None	Primary Key
domainName	The name representing the specific domain of expertise or interest	Varchar	255	No	None	

6.4 CV Class

CV (Curriculum Vitae)	
Name	CV
Alias	Resume
Where-used/how-used	The CV class is used within the user profile to store and manage the user's curriculum vitae, which is critical for assessment and recruiter evaluation processes.
Content description	This class encapsulates the data structure for storing user CVs, including personal information, work experience, education, skills, and other relevant details.

Column Details						
Column Name	**Description**	**Type**	**Length**	**Nullable**	**Default Value**	**Key Type**
cvID	Unique identifier for the user's CV	Integer		No	None	Primary Key
userID	Reference to the user that the CV belongs to	Integer		No	None	Foreign Key
cvContent	The content of the CV, including text and possible formatting	Text		Yes	None	

6.5 Question Class

Question	
Name	Question
Alias	AssessmentQuestion
Where-used/how-used	Utilized in the assessment process, where questions are presented to users based on their selected domain to evaluate their knowledge and skills.
Content description	Encompasses the attributes and methods necessary for creating, storing, and displaying questions for user assessments.

Column Details						
Column Name	**Description**	**Type**	**Length**	**Nullable**	**Default Value**	**Key Type**
questionID	Unique identifier for each question	Integer		No	None	Primary Key
domainID	The domain with which the question is associated	Integer		No	None	Foreign Key

content	The textual content of the question itself	Text		No	None	
type	The type or category of the question (e.g., multiple choice, free text)	Varchar	50	No	None	

6.6 Assessment Class

Assessment	
Name	Assessment
Alias	Evaluation, Test, Exam
Where-used/how-used	This class is used to manage the assessment processes, storing user responses to questions, calculating scores, and providing feedback. It is fundamental to the system's evaluation mechanism.[2]
Content description	Contains user assessment attempts, responses, scores, and any related metadata necessary for evaluating the performance of users.

Column Details						
Column Name	**Description**	**Type**	**Length**	**Nullable**	**Default Value**	**Key Type**
assessmentID	Unique identifier for the assessment instance	Integer		No	None	Primary Key
userID	Reference to the user taking the assessment	Integer		No	None	Foreign Key
questionIDs	List of IDs of questions included in the assessment	Text		Yes	None	
responses	User responses to the assessment questions	Text		Yes	None	
score	The score obtained by the user on this assessment	Decimal	5,2	Yes	0.00	

6.7 Ranking Class

Ranking	
Name	Ranking
Alias	Leaderboard, Scoreboard
Where-used/how-used	The Ranking class is used to maintain a leaderboard of users, reflecting their performance and progress within the system based on assessment results.
Content description	Stores the ranking information for users, including their overall scores and positions in comparison to other users.

Column Details						
Column Name	**Description**	**Type**	**Length**	**Nullable**	**Default Value**	**Key Type**
rankingID	Unique identifier for the user's ranking	Integer		No	None	Primary Key
userID	Reference to the user whose ranking is represented	Integer		No	None	Foreign Key
rankScore	Numerical representation of the user's ranking	Decimal	5,2	No	None	

6.8 Recruiter Class

Recruiter	
Name	Recruiter
Alias	HiringManager, TalentScout
Where-used/how-used	Used by the system to manage functionalities specific to recruiters, such as accessing candidate profiles, searching for candidates, and viewing assessment results.
Content description	This class handles the attributes and operations related to recruiters, allowing them to interact with candidate information effectively.

Column Details						
Column Name	**Description**	**Type**	**Length**	**Nullable**	**Default Value**	**Key Type**
recruiterID	Unique identifier for the recruiter	Integer		No	None	Primary Key
userID	Reference to the user account associated with the recruiter	Integer		No	None	Foreign Key

7 Application Design

This section details the application design of COGNIASSESS, including sequence diagrams, state transition diagrams, and DFD level 1 diagrams to illustrate the dynamic behavior and interactions within the application.

7.1 Sequence Diagram

Sequence diagrams are utilized to showcase the interaction between different components of the system over time.

7.1.1 User Authentication Sequence Diagram

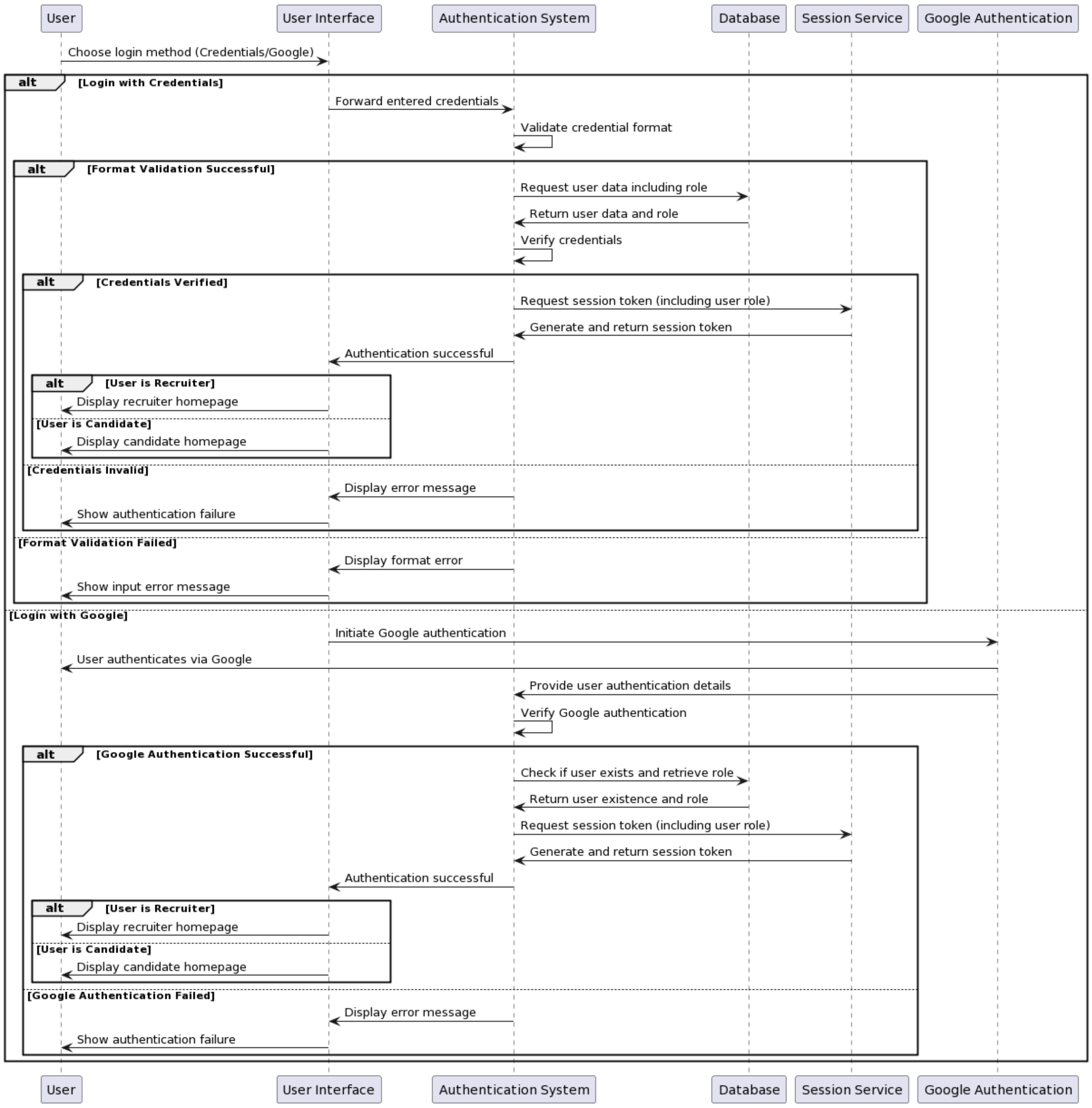
User
User Interface
Authentication System
Database
Session Service
Google Authentication
Choose login method (Credentials/Google)
alt [Login with Credentials]
Forward entered credentials
Validate credential format
alt [Format Validation Successful]
Request user data including role
Return user data and role
Verify credentials
alt [Credentials Verified]
Request session token (including user role)
Generate and return session token
Authentication successful
alt [User is Recruiter]
Display recruiter homepage
[User is Candidate]
Display candidate homepage
[Credentials Invalid]
Display error message
Show authentication failure
[Format Validation Failed]
Display format error
Show input error message
[Login with Google]
Initiate Google authentication
User authenticates via Google
Provide user authentication details
Verify Google authentication
alt [Google Authentication Successful]
Check if user exists and retrieve role
Return user existence and role
Request session token (including user role)
Generate and return session token
Authentication successful
alt [User is Recruiter]
Display recruiter homepage
[User is Candidate]
Display candidate homepage
[Google Authentication Failed]
Display error message
Show authentication failure
User
User Interface
Authentication System
Database
Session Service
Google Authentication

Figure 5: User Authentication Sequence Diagram

This diagram illustrates the process of user login, authentication, and error handling. It details the interaction between the User, Authentication system, and Database, highlighting steps like user input, credential verification, and session initialization.

7.1.2 Assessment Process Sequence Diagram

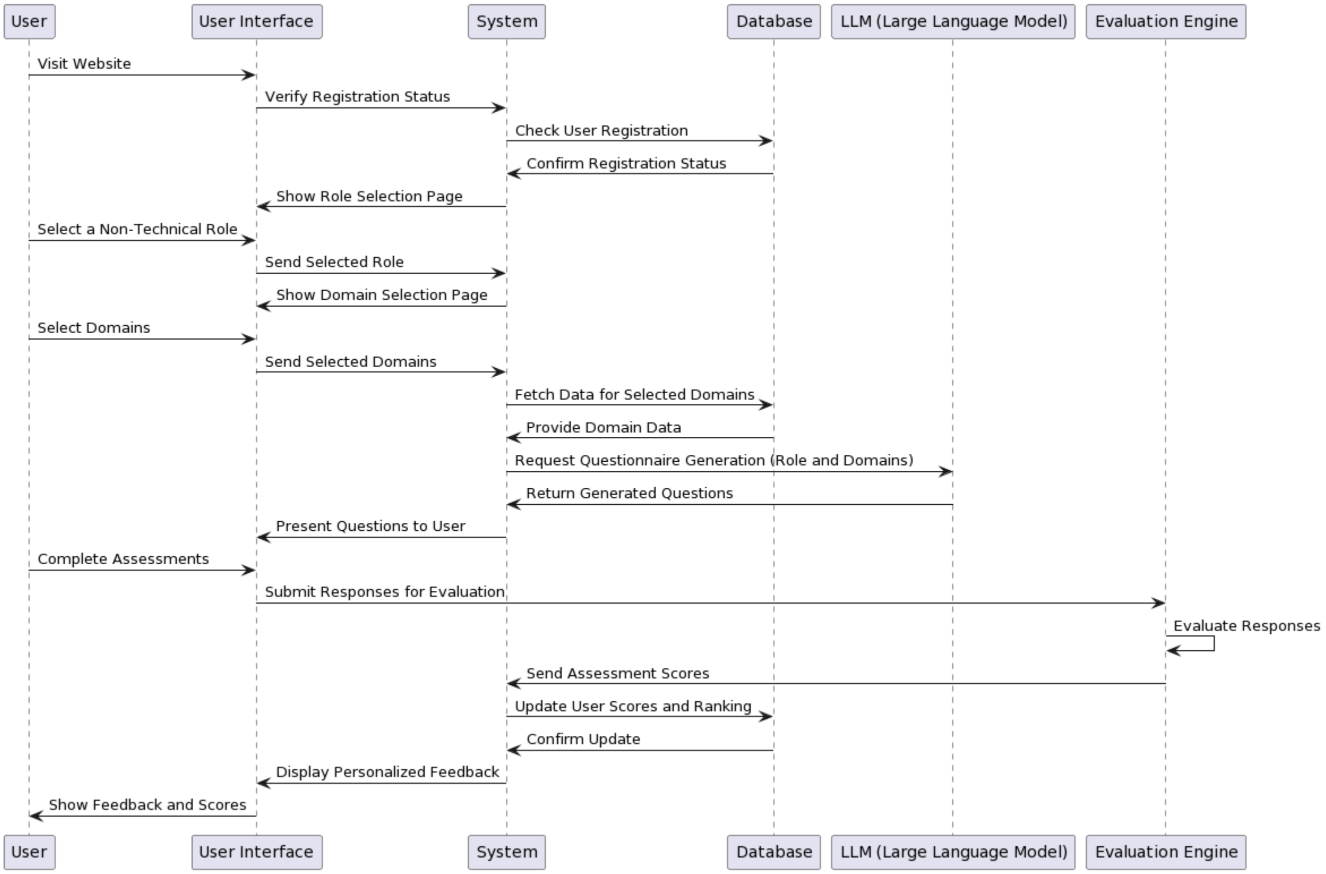
User
User Interface
System
Database
LLM (Large Language Model)
Evaluation Engine
Visit Website
Verify Registration Status
Check User Registration
Confirm Registration Status
Show Role Selection Page
Select a Non-Technical Role
Send Selected Role
Show Domain Selection Page
Select Domains
Send Selected Domains
Fetch Data for Selected Domains
Provide Domain Data
Request Questionnaire Generation (Role and Domains)
Return Generated Questions
Present Questions to User
Complete Assessments
Submit Responses for Evaluation
Evaluate Responses
Send Assessment Scores
Update User Scores and Ranking
Confirm Update
Display Personalized Feedback
Show Feedback and Scores
User
User Interface
System
Database
LLM (Large Language Model)
Evaluation Engine

Figure 6: Assessment Process Sequence Diagram

This diagram shows the sequence of events from when a user starts an assessment to when they receive their results. It describes interactions between User, Assessment Module, Questionnaire, and Evaluation Engine, covering question retrieval, answer submission, and scoring.

7.1.3 LLM Generation Process Sequence Diagram

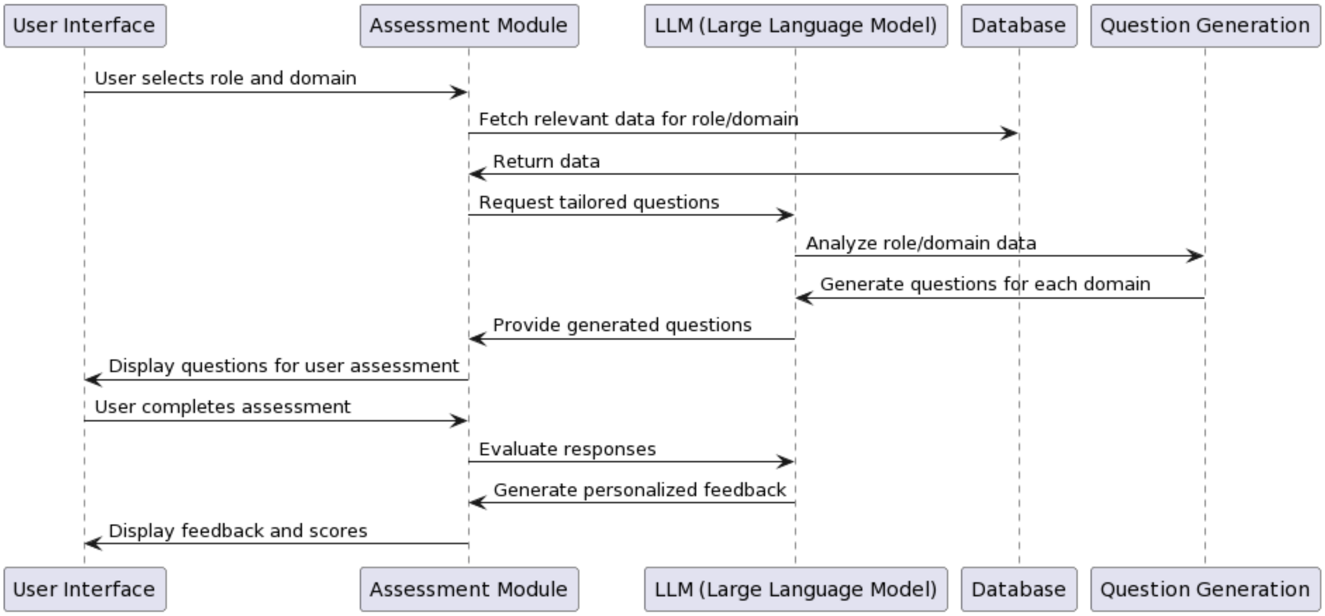
User Interface
Assessment Module
LLM (Large Language Model)
Database
Question Generation
User selects role and domain
Fetch relevant data for role/domain
Return data
Request tailored questions
Analyze role/domain data
Generate questions for each domain
Provide generated questions
Display questions for user assessment
User completes assessment
Evaluate responses
Generate personalized feedback
Display feedback and scores

Figure 7: LLM Generation Process Sequence Diagram

This diagram illustrates the process of generating assessment content using a fine-tuned Large Language Model (LLM) hosted on Hugging Face. It demonstrates the sequence of events and interactions for three distinct types of assessments: job scenario, personality, and domain-role based. The diagram covers the system's interaction with the LLM for each assessment type, detailing the steps from the initial request for content generation to the presentation of tailored questions to the user. This includes:

1. **Job Scenario Assessment**: Showcasing how the system prompts the LLM to generate realistic job scenario questions based on the user's selected role and domain.

2. **Personality Assessment**: Illustrating the generation of questions aimed at evaluating personality traits.

3. **Domain-Role Based Assessment**: Depicting the process of creating domain-specific questions that align with the user's chosen non-technical role.

The sequence diagram also highlights the integration of the LLM with the system's database and user interface, ensuring a seamless and dynamic assessment creation process.

7.1.4 LLM Fine-Tuning Process Sequence Diagram

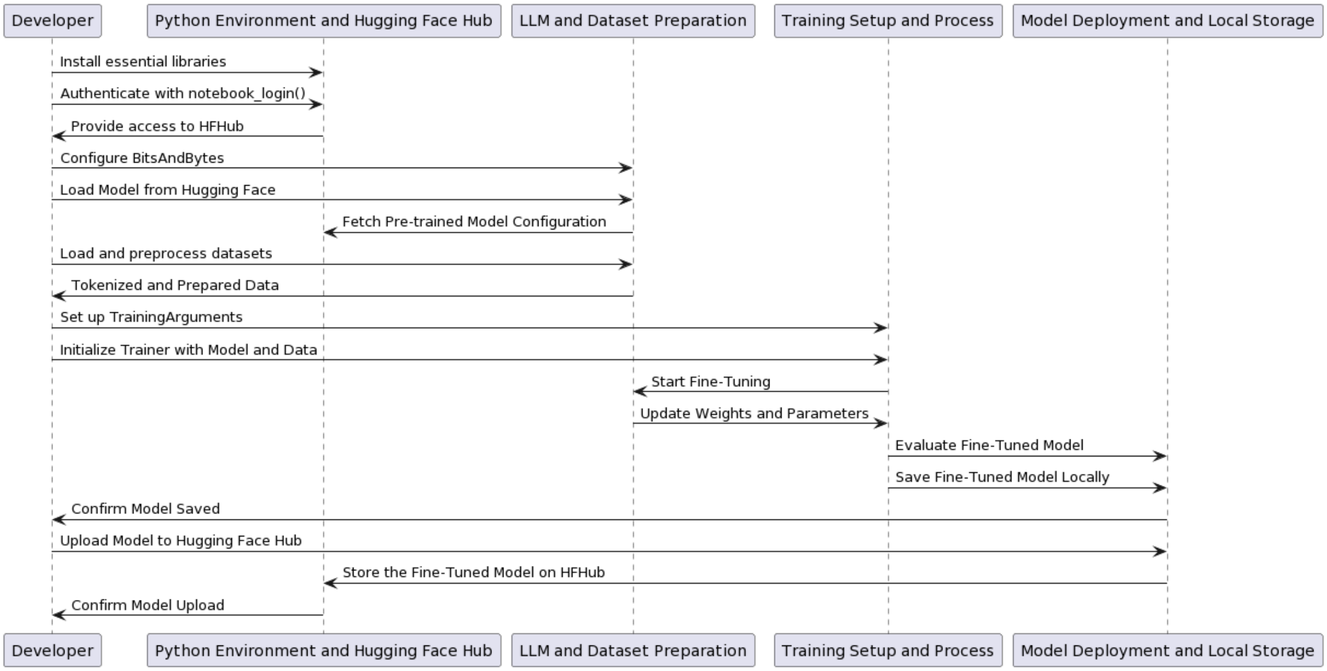
Developer
Python Environment and Hugging Face Hub
LLM and Dataset Preparation
Training Setup and Process
Model Deployment and Local Storage
Install essential libraries
Authenticate with notebook_login()
Provide access to HFHub
Configure BitsAndBytes
Load Model from Hugging Face
Fetch Pre-trained Model Configuration
Load and preprocess datasets
Tokenized and Prepared Data
Set up TrainingArguments
Initialize Trainer with Model and Data
Start Fine-Tuning
Update Weights and Parameters
Evaluate Fine-Tuned Model
Save Fine-Tuned Model Locally
Confirm Model Saved
Upload Model to Hugging Face Hub
Store the Fine-Tuned Model on HFHub
Confirm Model Upload
Developer
Python Environment and Hugging Face Hub
LLM and Dataset Preparation
Training Setup and Process
Model Deployment and Local Storage

Figure 8: LLM Fine-Tuning Process Sequence Diagram

This diagram visualizes the fine-tuning process of a Large Language Model (LLM) hosted on Hugging Face, specifically tailored for the CogniAssess project. It encapsulates the comprehensive steps involved in preparing, training, and deploying the fine-tuned LLM. The sequence includes:

1. **Module Installation and Imports**: Showcasing the installation of essential Python libraries like 'torch', 'transformers', and 'accelerate', and their subsequent imports into the environment.

2. **Model and Tokenizer Setup**: Illustrating the process of configuring and initializing the LLM with specific parameters, including the BitsAndBytes configuration for memory-efficient model loading.

3. **Dataset Preparation**: Depicting the steps for loading, preprocessing, and tokenizing datasets, specifically focused on generating prompts for non-technical role assessments.

4. **Training Parameter Configuration**: Outlining the setup of training arguments, including batch sizes, learning rates, and optimization strategies.

5. **Model Fine-Tuning**: Detailing the training process, where the LLM is fine-tuned on the prepared dataset using specific training parameters and strategies.

6. **Model Evaluation and Deployment**: Highlighting the evaluation of the fine-tuned model and its subsequent deployment, including saving and uploading the model to Hugging Face Hub.

This sequence diagram emphasizes the intricate and methodical approach to fine-tuning an LLM for the specific needs of the CogniAssess project, ensuring the model is optimally trained for assessing non-technical roles.

7.1.5 Recruiter-Candidate Interaction Sequence Diagram

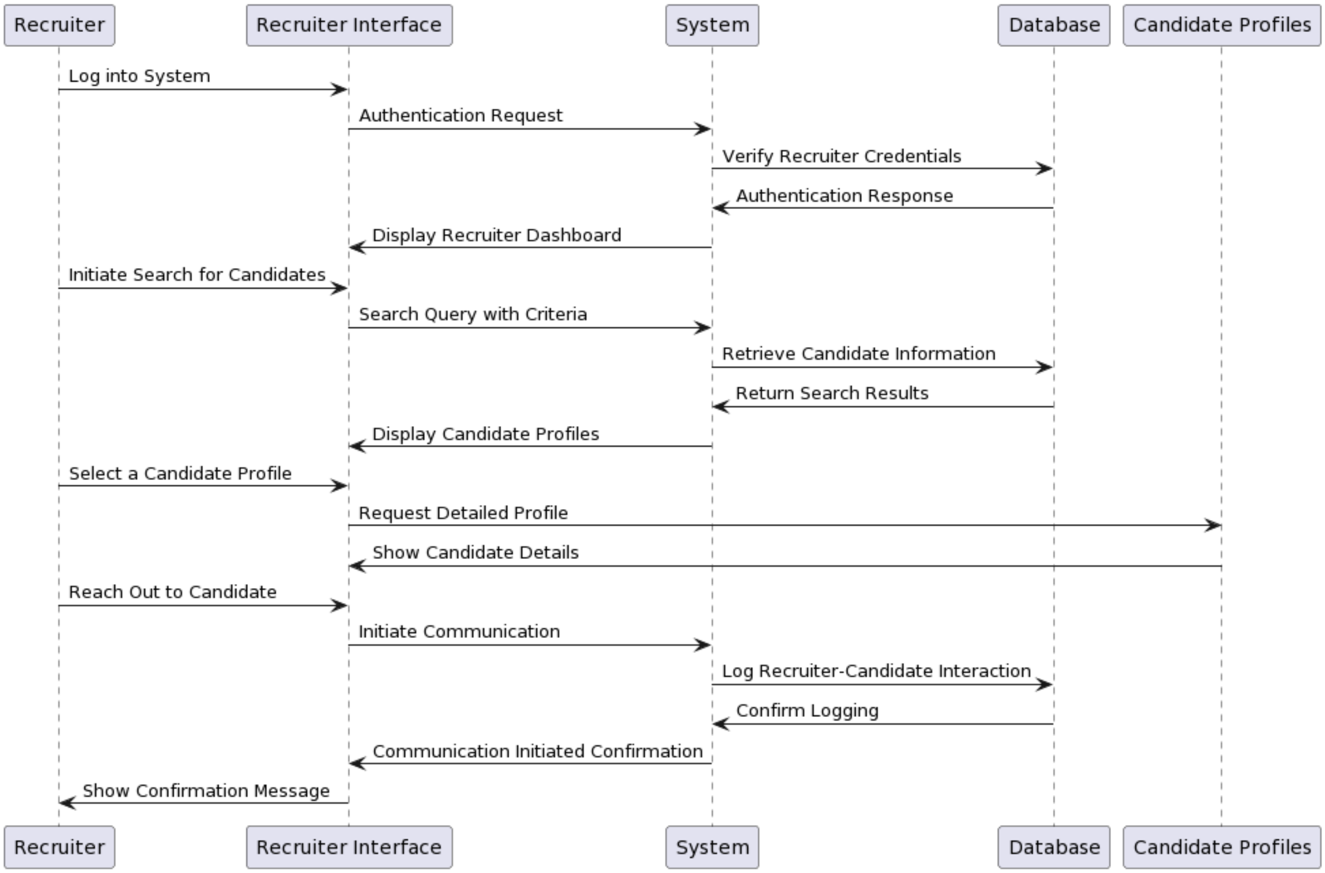

Figure 9: Recruiter-Candidate Interaction Sequence Diagram

This diagram details the steps a recruiter takes to search for, view, and contact candidates. It outlines interactions involving the Recruiter, Search Module, Candidate Database, and Communication System.

7.2 State Diagram

State diagrams provide insights into the various states of system components and how transitions occur between these states.

7.2.1 User State Diagram

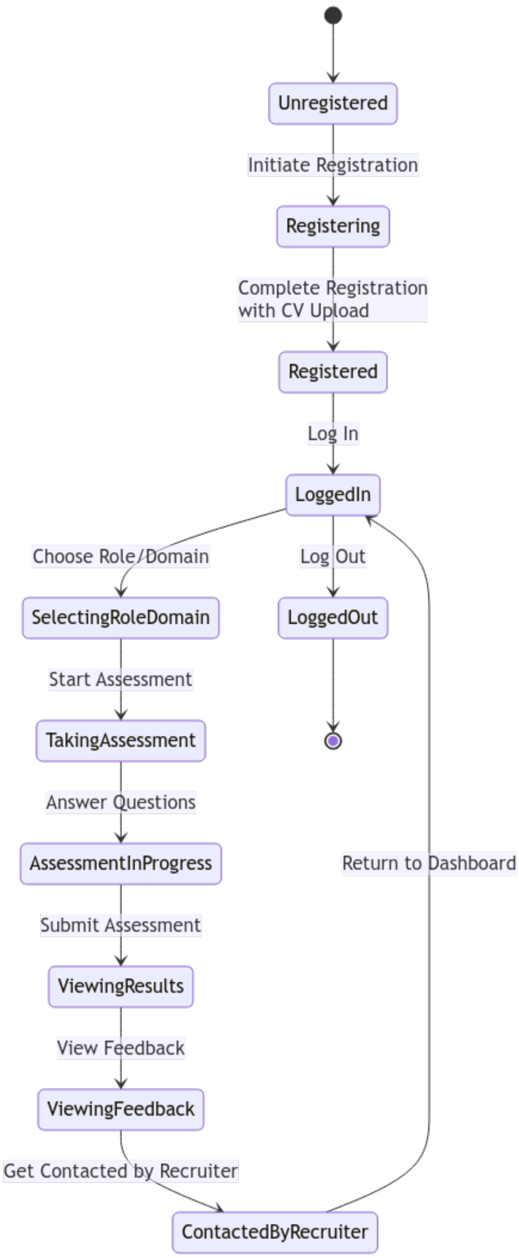

Figure 10: User State Diagram

This diagram represents the different states of a user account (e.g., Registered, Logged In, Taking Assessment) and describes what causes transitions between states, such as logging in, starting an assessment, or logging out.

7.2.2 Assessment Module State Diagram

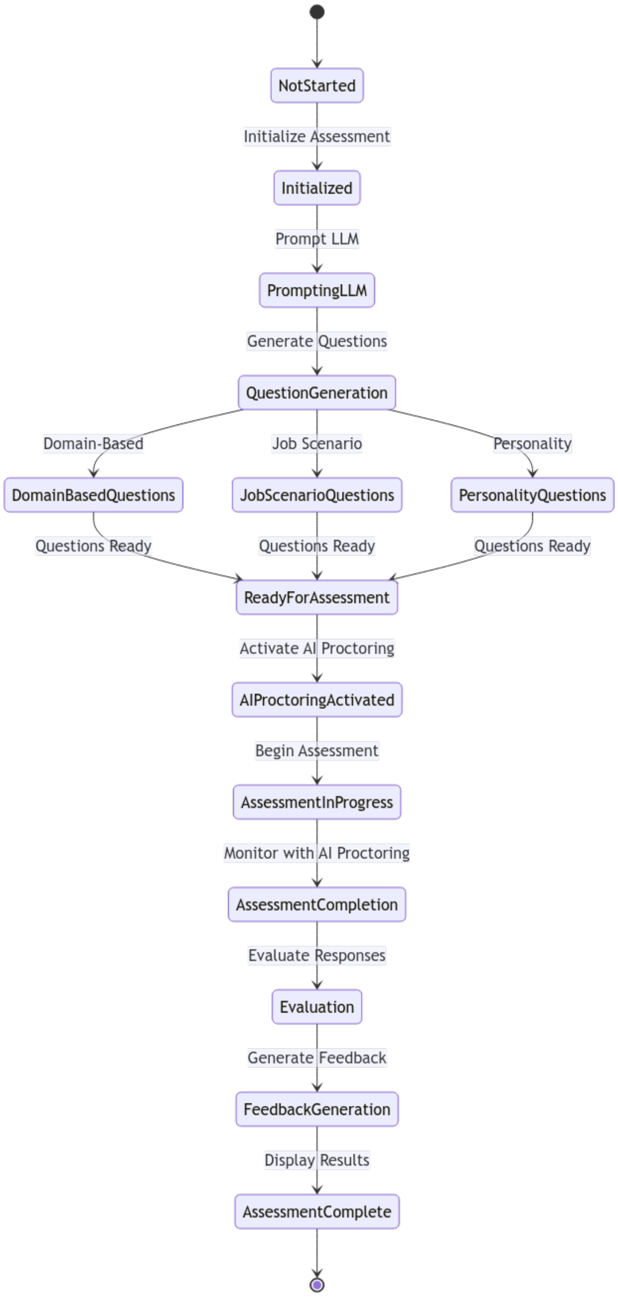

Figure 11: Assessment Module State Diagram

This diagram shows the different states of the assessment process (e.g., Not Started, In Progress, Completed) and details triggers for state transitions like starting an assessment, submitting answers, and receiving results.

7.2.3 Recruiter Dashboard State Diagram

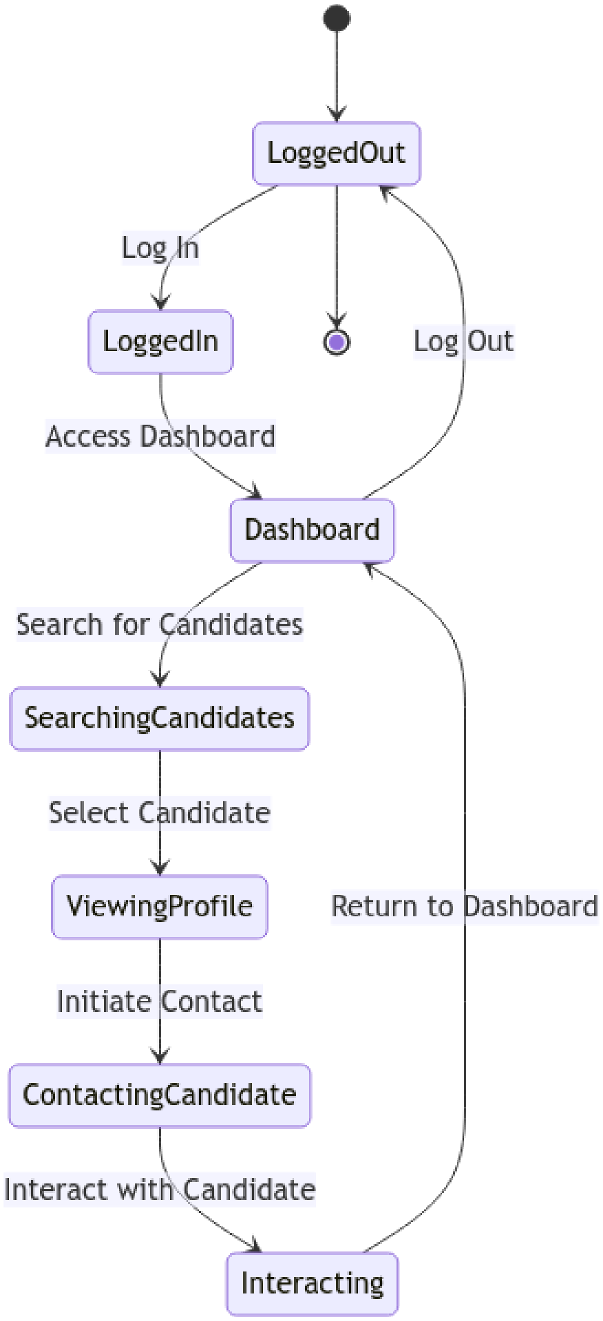

Figure 12: Recruiter Dashboard State Diagram

This diagram outlines the states involved in a recruiter's interaction with the system (e.g., Searching, Viewing Profiles) and explains transitions based on recruiter actions like initiating a search, selecting a candidate, or initiating contact.

7.3 DFD Level 1 Diagram

The Data Flow Diagram (DFD) Level 1 provides a high-level overview of data movement within the system.

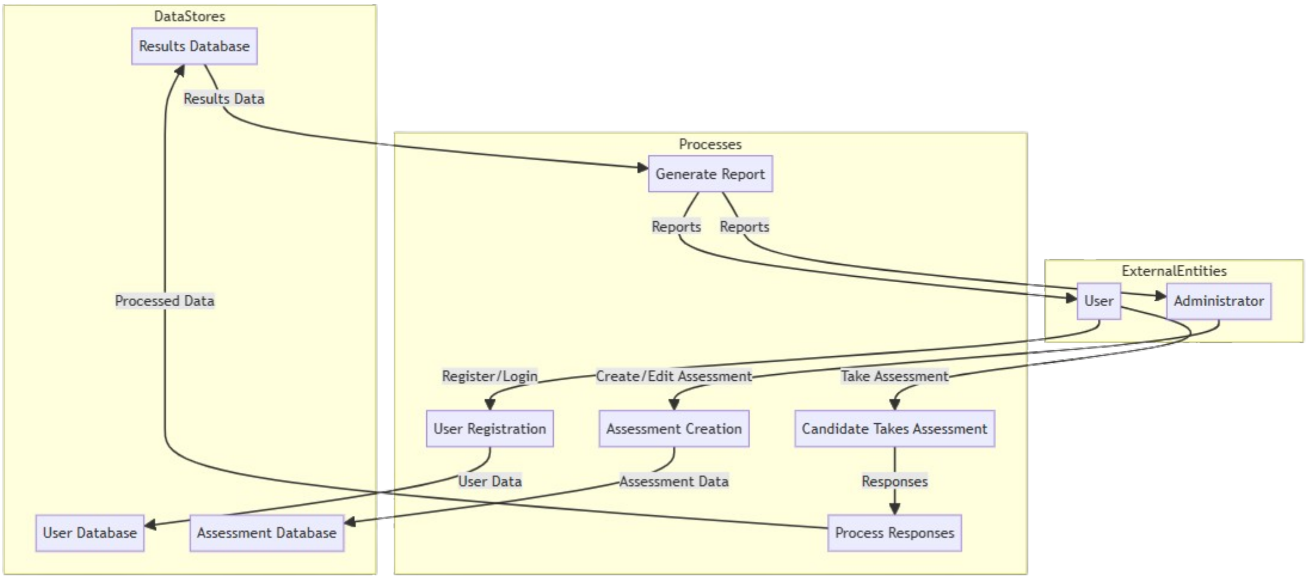

Figure 13: DFD Level 1 Diagram

This diagram includes major system components like User Management, Assessment Processing, and Reporting and describes the flow of data between these components and how they interact to provide the system's functionality.

8 Refrences

References

[1] ”Generative Job Recommendations with Large Language Model,” ar5iv. [Online]. Available: https://ar5iv.org/html/2307.02157

[2] ”Personality Prediction Via CV Analysis using Machine Learning,” IJERT. [Online]. Available: https://www.ijert.org/personality-prediction-via-cv-analysis-using-machine-learning

[3] ”Job Role and Personality Prediction Using CV and Text Analysis,” IJRASET. [Online]. Available: https://issuu.com/ijraset/docs/job_role_and_personality_prediction_using_cv_and_t

9 Appendices

1. Use Case Diagrams
2. Data Flow Diagrams
3. System Architecture Diagram
4. User Interface Mockups
5. Sample Assessment Questions
6. Code Samples
7. Glossary of Terms