|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | | **Software Design Specifications**  ***AI Candidate Assessment System***  **Version: 1.3**   |  |  | | --- | --- | | Project Code | F24-88 | | Supervisor | Ms. Nida Munawar | | Co Supervisor | - | | Project Team | *Ashesh Kumar(21K-3451)*  *Abdul Wasay(21K-4589)*  *Fahad Ahmed(21K-4926)* | | Submission Date |  | |   **Document History**   |  |  |  |  | | --- | --- | --- | --- | | Version | Name of Person | Date | Description of change | | 1.0 | Ashesh Kumar | 2/11/24 | Document Created | | 1.0 | Fahad Ahmed | 10/11/24 | Added System Architecture | | 1.0 | Ashesh Kumar | 12/11/24 | Database Design | | 1.1 | Ashesh Kumar | 25/11/24 | Updated Database Design | | 1.1 | Abdul Wasay | 2/12/24 | Added Sequence Diagrams | | 1.1 | Abdul Wasay | 3/12/24 | Added State Diagram and references | | 1.2 | Abdul Wasay | 5/12/24 | Updated Sequence Diagrams | | 1.2 | Fahad Ahmed | 5/12/24 | Added Appendices | | 1.3 | Ashesh Kumar | 10/12/24 | Made overall corrections |     **Distribution List**   |  |  | | --- | --- | | **Name** | **Role** | | Miss Nida Munawar | Supervisor | | - | Co Supervisor | |  |  |     **Document Sign-Off**   |  |  |  |  |  | | --- | --- | --- | --- | --- | | **Version** | **Sign-off Authority** | **Project Role** | **Signature** | **Sign-off Date** | | 1.3 | Miss Nida Munawar | Supervisor | A close-up of a signature  Description automatically generated | 11-12-2024 | |  |  |  |  |  | |  |  |  |  |  | |  |  |  |  |  | |  |  |  |  |  | |  |  |  |  |  | |  |  |  |  |  | |

**Document Information**

| **Category** | **Information** |
| --- | --- |
| Customer | FAST-NU |
| Project | AI Candidate Assessment System |
| Document | Software Design Specification |
| Document Version | 1.5 |
| Status | Final |
| Author(s) | Ashesh Kumar  Abdul Wasay  Fahad Ahmed |
| Approver(s) |  |
| Issue Date | 10/12/24 |
| Document Location |  |
| Distribution | Advisor  Project Coordinator’s Office (through Advisor) |

**Definition of Terms, Acronyms and Abbreviations**

| **Term** | **Description** |
| --- | --- |
| AI | Artificial Intelligence |
| SBERT | Sentence Bidirectional Encoder Representations from Transformers |
| GPT | Generative Pre-trained Transformer |
| REST API | Representational State Transfer Application Programming Interface |
| RBAC | Role-Based Access Control |
| SMTP | Simple Mail Transfer Protocol |
| UI | User Interface |

**Table of Contents**

[1 Introduction 6](#_Toc184753337)

[1.1 Purpose of Document 6](#_Toc184753338)

[1.2 Intended Audience 6](#_Toc184753339)

[1.3 Document Convention 6](#_Toc184753340)

[1.4 Project Overview 6](#_Toc184753341)

[1.5 Scope 6](#_Toc184753342)

[2 Design Considerations 8](#_Toc184753343)

[2.1 Assumptions and Dependencies 8](#_Toc184753344)

[2.2 Risks and Volatile Areas 8](#_Toc184753345)

[3 System Architecture 8](#_Toc184753346)

[3.1 System Level Architecture 8](#_Toc184753347)

[3.2 Software Architecture 10](#_Toc184753348)

[4 Design Strategy 11](#_Toc184753349)

[*4.1 Design Goals* 11](#_Toc184753350)

[*4.2 Key Strategies* 11](#_Toc184753351)

[*4.2.1 Future System Extension or Enhancement* 11](#_Toc184753352)

[*4.2.2 System Reuse* 12](#_Toc184753353)

[*4.2.3 User Interface Paradigms* 12](#_Toc184753354)

[*4.2.4 Data Management* 12](#_Toc184753355)

[*4.2.5 Concurrency and Synchronization* 12](#_Toc184753356)

[5 Detailed System Design 13](#_Toc184753357)

[5.1 Database Design 13](#_Toc184753358)

[5.1.1 ER Diagram 13](#_Toc184753359)

[5.1.2 Data Dictionary 14](#_Toc184753360)

[5.1.2.1 Data 1 14](#_Toc184753361)

[5.1.2.2 Data 2 15](#_Toc184753362)

[5.1.2.3 Data 3 16](#_Toc184753363)

[5.1.2.4 Data 4 17](#_Toc184753364)

[5.1.2.5 Data 5 18](#_Toc184753365)

[5.1.2.6 Data 6 19](#_Toc184753366)

[Application Design 20](#_Toc184753367)

[5.1.3 Sequence Diagram 20](#_Toc184753368)

[5.1.3.1 CV Upload and Analysis 20](#_Toc184753369)

[5.1.3.2 Test Submission and Evaluation 21](#_Toc184753370)

[5.1.3.3 Final Candidate Selection 21](#_Toc184753371)

[5.1.4 State Diagram 22](#_Toc184753372)

[5.1.4.1 Candidate Profile State Diagram 22](#_Toc184753373)

[5.1.4.2 Test State Diagram 23](#_Toc184753374)

[5.1.4.3 HR Decision Process State Diagram 24](#_Toc184753375)

[5.1.5 Data Flow Diagram 25](#_Toc184753376)

[6 References 25](#_Toc184753377)

[7 Appendices 27](#_Toc184753378)

[*7.1 Supporting Materials* 27](#_Toc184753379)

# Introduction

## Purpose of Document

*The purpose of this document is to define the architectural and component-level design of the AI Candidate Assessment System. This document provides a comprehensive blueprint for developers and other stakeholders to understand the system's structure and functionality. The Object-Oriented Design (OOD) methodology will be used for this project.*

## Intended Audience

* *Fast NU*
* *Jury*
* *Supervisor (Ms. Nida Munawar)*
* *Students of Fast NU*
* *Our Team(Designer, Developer, Tester)*
* *Potential Users of this product*

## Document Convention

* *Font Family = Arial*
* *Font Size = 12 for headings, 10 for the rest of the content*

## Project Overview

*The AI Candidate Assessment System automates the recruitment process for HR departments. It analyzes candidate CVs, generates skill-based tests tailored to job descriptions, evaluates test performance, and provides HR professionals with detailed reports for decision-making.*

## Scope

*This system is designed to automate the hiring process, improving efficiency and accuracy in evaluating job applicants. By leveraging AI technologies such as SBERT for CV analysis and GPT for test generation, the system ensures more objective and consistent decision-making. It aims to simplify the process for HR professionals, reducing manual effort and time while enhancing the candidate evaluation process. The system will streamline workflows, minimize biases, and ultimately accelerate the recruitment cycle.*

***Included Functionalities:***

* ***CV Scanning****: Analyzing applicants' CVs using AI to evaluate qualifications and match candidates to job requirements.*
* ***AI-Generated Test****: Creating personalized technical tests based on the candidate’s required skill set and the job role they are applying for.*
* ***Test Evaluation****: Automated generation of evaluation reports, accessible by HR for decision-making.*
* ***HR Decision Support****: Providing reports to HR to assess candidate suitability.*
* ***Job Role Support****: Initially, the system supports only the Junior Developer role, and additional roles or languages will be implemented later*

***Excluded Functionalities****:*

* ***Job Posting Management****: The system does not handle job posting creation or management by HR.*
* ***Candidate Communication****: It does not offer direct communication features such as messaging or interviews between HR and candidates.*
* ***Job Role Expansion****: Initially, the system supports only the Junior Developer role, and additional roles or languages will be implemented later.*
* ***Advanced Analytics****: The system does not provide deep predictive analytics or insights beyond the basic reports for HR decision-making.*

# Design Considerations

## Assumptions and Dependencies

***Assumptions***

*● Users have basic digital literacy to navigate the system and complete necessary tasks such as uploading CVs and taking tests.*

*● High-speed internet is available for most users to ensure smooth system access and functionality.*

*● Candidates will submit accurate and complete information in their CVs and other required documents.*

***Dependencies***

*● System performance is dependent on the reliability and scalability of Successful integration with third-party APIs for email notifications, data storage, and AI model processing is essential for system functionality.*

*● Continuous access to updated AI models (SBERT and GPT) is crucial for accurate CV analysis and test generation.*

## Risks and Volatile Areas

***Risks****:*

* ***Model Accuracy****: The AI models may yield inaccurate results if the training data is not adequate or representative.*
* ***Data Security****: Candidate data and HR details are sensitive and require strong encryption to prevent breaches.*
* ***System Downtime****: Unscheduled outages could disrupt hiring processes and lead to user dissatisfaction.*

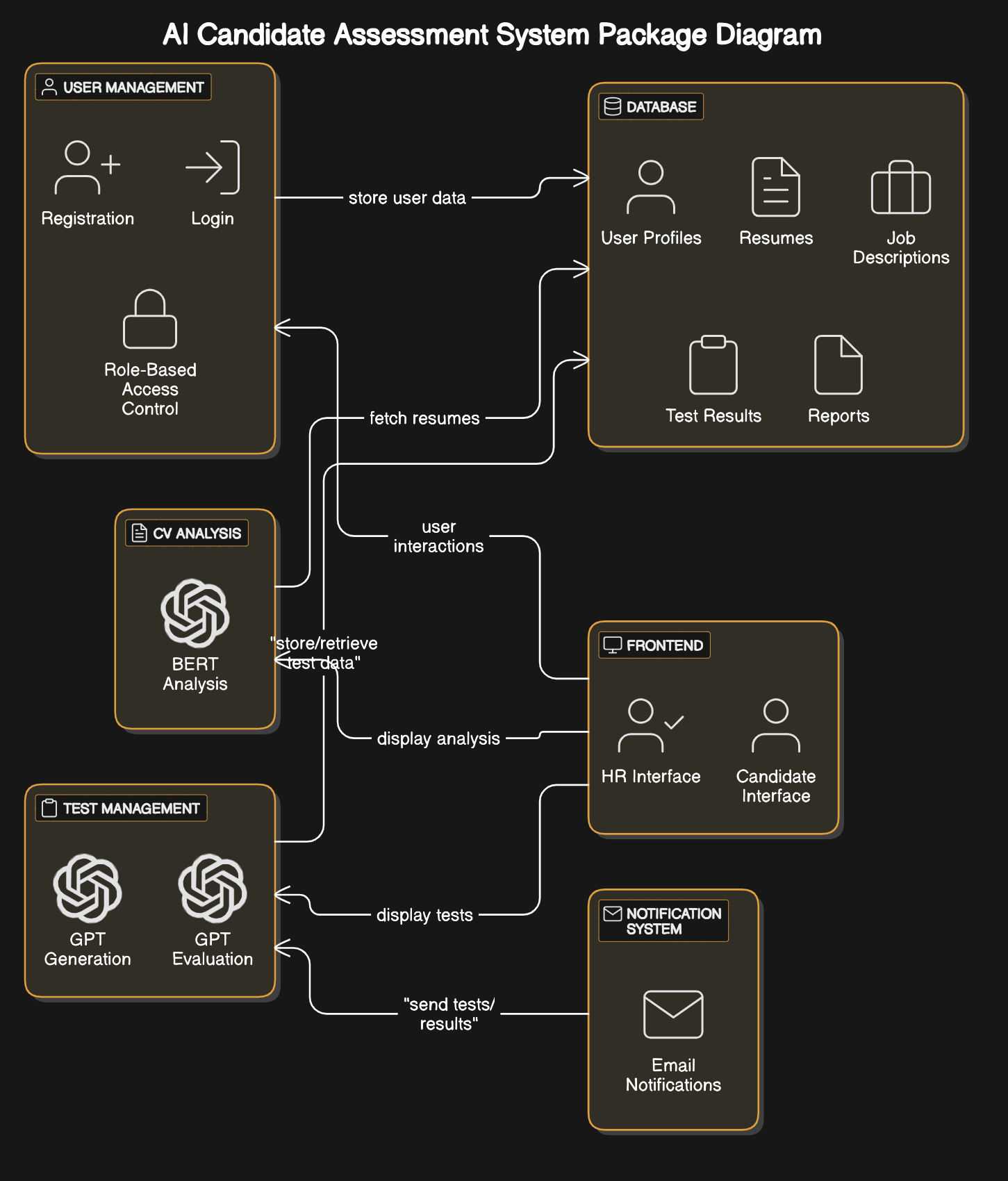
# System Architecture

## System Level Architecture

*The AI Candidate Assessment System is designed using a modular architecture to ensure scalability, maintainability, and clear separation of concerns. The system comprises the following core packages, as depicted in the Diagram:*

1. ***User Management****:*
   * *Handles user registration and authentication.*
   * *Implements role-based access control (RBAC) to differentiate between HR and candidate users.*
2. ***CV Analysis****:*
   * *Integrates with SBERT for extracting and analyzing key information from resumes.*
   * *Communicates with the database to store and retrieve resume data.*
3. ***Test Management****:*
   * ***Test Generation****: Uses GPT to generate skill-based tests tailored to job descriptions.*
   * ***Test Evaluation****: Also integrates with GPT for automated assessment of test submissions, generating scores and performance reports.*
4. ***Notification System****:*
   * *Sends automated email notifications to candidates and HR for test links, results, and other updates.*
5. ***Frontend****:*
   * *Provides distinct interfaces for HR professionals and candidates.*
   * *Interacts with backend services for data display and updates.*
6. ***Database****:*
   * *Manages persistent storage for user profiles, resumes, job descriptions, test results, and reports.*

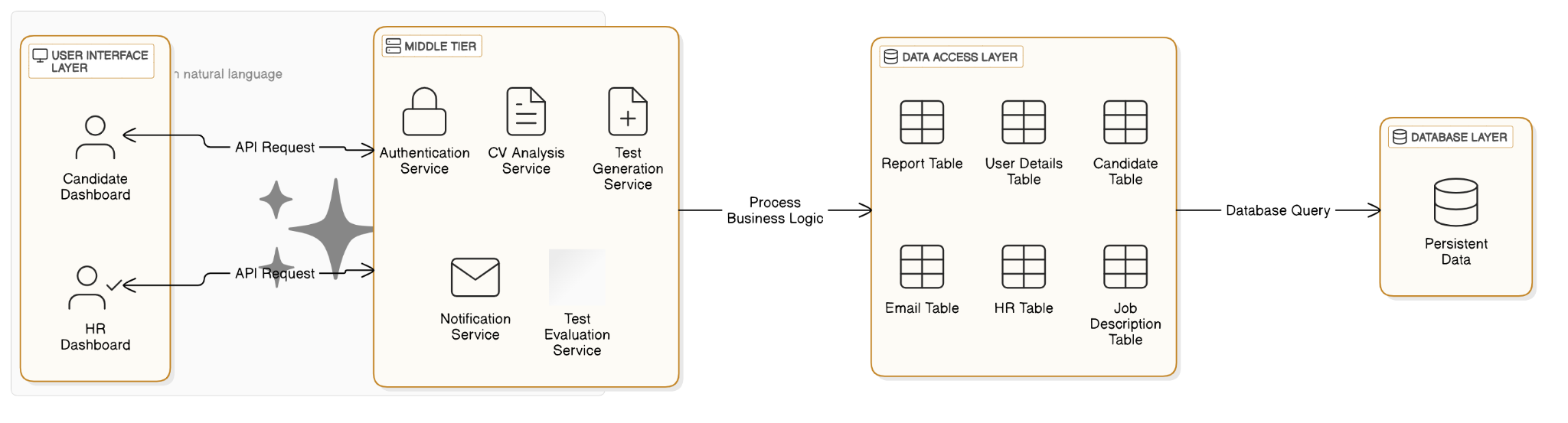
*The* ***Package Diagram*** *above illustrates the relationships and dependencies between these components, emphasizing the flow of data and interactions.*

**

## Software Architecture

*The system is structured using a three-tier architecture model, ensuring separation of concerns:*

1. ***Presentation Layer****:*
   * *Includes HR and Candidate Dashboards.*
   * *Built using React.js for dynamic and responsive user interfaces.*
2. ***Middle Tier (Business Logic)****:*
   * *Implements the core logic and services:*
     + ***Authentication Service****: Manages login, registration, and role verification.*
     + ***CV Analysis Service****: Processes resumes using SBERT for suitability scoring.*
     + ***Test Generation Service****: Generates skill-specific tests using GPT.*
     + ***Test Evaluation Service****: Evaluates test submissions and provides detailed performance insights using GPT.*
     + ***Notification Service****: Handles email-based communication.*
3. ***Data Access Layer****:*
   * *Interacts with MongoDB to manage system data such as profiles, test results, and reports.*

**

# Design Strategy

### *4.1 Design Goals*

*The AI Candidate Assessment System is designed with the following goals in mind:*

1. ***Scalability****:*
   * *The system should accommodate increasing users, roles, and job descriptions without significant redesign.*
2. ***Modularity****:*
   * *Each component should have a clearly defined responsibility to ensure easy maintenance and updates.*
3. ***Reusability****:*
   * *Core functionalities, such as CV analysis and test generation, should be reusable across different modules.*
4. ***Security****:*
   * *Ensure sensitive data such as resumes, test results, and user credentials are stored and transmitted securely.*

### *4.2 Key Strategies*

#### *4.2.1 Future System Extension or Enhancement*

* ***AI Model Updates****: The architecture supports replacing or retraining AI models (e.g., SBERT and GPT) without significant codebase changes.*
* ***Role Expansion****: New roles or industries can be added by defining their specific CV and test evaluation criteria.*
* ***Feature Addition****: The modular architecture allows for adding features like advanced analytics or job posting management.*

#### *4.2.2 System Reuse*

* *The CV Analysis and Test Management modules are designed as independent services that can be reused in other systems requiring similar AI-based functionalities.*
* *Notification and Authentication services can also be repurposed for systems with email-based communication needs.*

#### *4.2.3 User Interface Paradigms*

* *The system uses a* ***responsive design approach*** *to ensure accessibility across devices (desktop, tablet, and mobile).*
* *HR and Candidate dashboards follow a minimalist design for intuitive navigation, leveraging React.js for dynamic interactions.*

#### *4.2.4 Data Management*

* ***Storage****: MongoDB is used to store structured data like user profiles, test results, and reports.*
* ***Persistence****: Regular database backups are scheduled to ensure data integrity.*
* ***Distribution****: The system supports distributed database deployment for scalability and fault tolerance.*

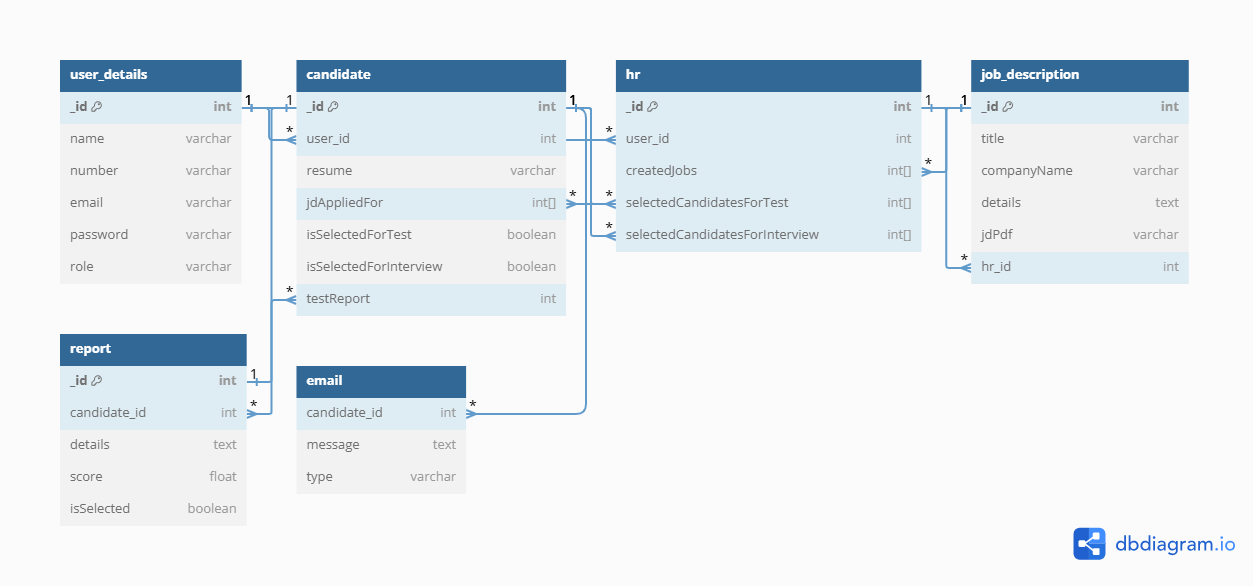
#### *4.2.5 Concurrency and Synchronization*

* ***Concurrent Requests****: The system handles multiple requests simultaneously using Node.js's non-blocking architecture.*
* ***Synchronization****: Critical operations like test evaluation and CV scoring are synchronized to avoid conflicts.*

# Detailed System Design

## Database Design

### ER Diagram





### Data Dictionary

#### Data 1

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **User** | | | | | | | |
| **Name** | | user\_details | | | | | |
| **Alias** | | User | | | | | |
| **Where-used/how-used** | | Manages all user entities, both candidates and HR. | | | | | |
| **Content description** | | Stores essential user information for authentication and role management. | | | | | |
|  | | | | | | | |
| **Column Name** | **Description** | | **Type** | **Length** | **Null able** | **Default Value** | **Key Type** |
| *\_id* | *Unique identifier* | | *int* |  | *NO* | *None* | *PK* |
| *name* | *Full name of the user* | | *varchar* | *255* | *NO* | *None* |  |
| *number* | *Contact number* | | *varchar* | *255* | *Yes* | *NULL* |  |
| *email* | *Email address* | | *varchar* | *255* | *NO* | *None* |  |
| *password* | *Password for login* | | *Varchar* | *255* | *NO* | *None* |  |
| *role* | *Role of the user* | | *Varchar* | *50* | *NO* | *None* |  |

#### Data 2

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Candidate** | | | | | | | |
| **Name** | | candidate | | | | | |
| **Alias** | | Candidate | | | | | |
| **Where-used/how-used** | | Details specific to candidates. | | | | | |
| **Content description** | | Stores data related to job applications, including resumes and application status. | | | | | |
|  | | | | | | | |
| **Column Name** | **Description** | | **Type** | **Length** | **Null able** | **Default Value** | **Key Type** |
| *hr\_id* | *unique identifier* | | *int* |  | *NO* | *None* | *PK* |
| *user\_id* | *References user\_details table* | | *Int* |  | *NO* | *None* | *FK* |
| *resume* | *Path to resume document* | | *varchar* | *255* | *NO* | *None* |  |
| *jd\_applied\_for* | *IDs of job descriptions applied for* | | *int[ ]* |  | *Yes* | *None* | *FK* |
| *isSelectedForTest* | *Candidate selected for test flag* | | *boolean* |  | *NO* | *False* |  |
| *isSelectedForInterview* | *Interview selection flag* | | *boolean* |  | *No* | *False* |  |
| *testReport* | *Reference to test results* | | *int* |  | *YES* | *NULL* | *FK* |

.

#### Data 3

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **HR** | | | | | | | |
| **Name** | | hr | | | | | |
| **Alias** | | HR | | | | | |
| **Where-used/how-used** | | Used by HR for managing candidates and vacancies. | | | | | |
| **Content description** | | Contains data specific to HR personnel activities. | | | | | |
|  | | | | | | | |
| **Column Name** | **Description** | | **Type** | **Length** | **Null able** | **Default Value** | **Key Type** |
| *\_id* | *unique identifier* | | *Int* |  | *NO* | *None* | *PK* |
| *user\_id* | *Link to user\_details* | | *int* |  | *NO* | *None* | *FK* |
| *createdJobs* | *List of job descriptions created* | | *int[ ]* |  | *YES* | *NULL* | *FK* |
| *selectedCandidatesForTest* | *IDs of candidates selected for tests* | | *int[ ]* |  | *Yes* | *NULL* |  |
| *selectedCandidatesForInterview* | *IDs of candidates selected for interviews* | | *int[ ]* |  | *Yes* | *NULL* |  |

#### Data 4

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Job Description** | | | | | | | |
| **Name** | | job\_description | | | | | |
| **Alias** | | Job | | | | | |
| **Where-used/how-used** | | Used to store job details that candidates apply for. | | | | | |
| **Content description** | | Contains descriptions of jobs including requirements and details. | | | | | |
|  | | | | | | | |
| **Column Name** | **Description** | | **Type** | **Length** | **Null able** | **Default Value** | **Key Type** |
| *\_id* | *unique identifier* | | *Int* |  | *NO* | *None* | *PK* |
| *title* | *Title of the job position* | | *Varchar* | *255* | *NO* | *None* |  |
| *companyName* | *Name of the company* | | *Varchar* | *255* | *NO* | *None* |  |
| *details* | *Description of job responsibilities* | | *text* |  | *NO* | *None* |  |
| *jdPdf* | *Link to the job description PDF* | | *varchar* | *255* | *NO* | *None* |  |
| *hr\_id* | *HR who created the job description* | | *int* |  | *NO* | *None* | *FK* |

#### Data 5

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Report** | | | | | | | |
| **Name** | | report | | | | | |
| **Alias** | | Report | | | | | |
| **Where-used/how-used** | | Used to store assessment reports for each candidate after tests. | | | | | |
| **Content description** | | Stores detailed evaluation reports and scores for candidates. | | | | | |
|  | | | | | | | |
| **Column Name** | **Description** | | **Type** | **Length** | **Null able** | **Default Value** | **Key Type** |
| *\_id* | *unique identifier* | | *Int* |  | *NO* | *None* | *PK* |
| *candidate\_id* | *Link to the candidate table* | | *int* | *15* | *NO* | *None* | *FK* |
| *details* | *Textual details of the report* | | *text* |  | *NO* | *None* |  |
| *score* | *Numeric score from the assessment* | | *float* |  | *NO* | *None* |  |
| *isSelected* | *Whether candidate is selected or not* | | *boolean* |  | *NO* | *False* |  |

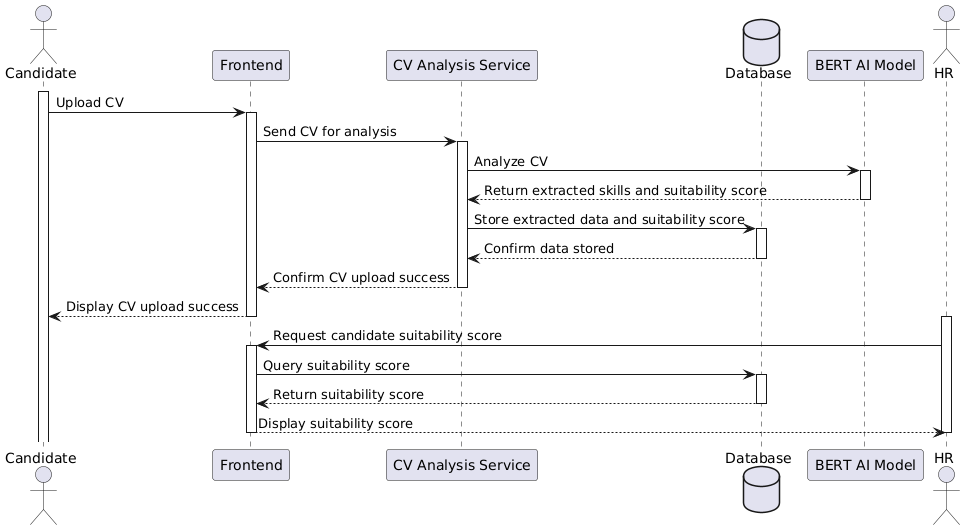
#### Data 6

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Email** | | | | | | | |
| **Name** | | email | | | | | |
| **Alias** | | Email | | | | | |
| **Where-used/how-used** | | Manages emails sent to candidates and HR staff. | | | | | |
| **Content description** | | Stores messages sent as part of the recruitment process. | | | | | |
|  | | | | | | | |
| **Column Name** | **Description** | | **Type** | **Length** | **Null able** | **Default Value** | **Key Type** |
| *\_id* | *unique identifier* | | *Int* |  | *NO* | *None* | *PK* |
| *candidate\_id* | *Reference to the candidate receiving the email* | | *int* |  | *NO* | *None* | *FK* |
| *message* | *Content of the email message* | | *text* |  | *NO* | *None* |  |
| *type* | *Type of email (e.g, “test\_Schedule”, “hr\_Decision”, “general”)* | | *Varchar* | *255* | *NO* | *None* |  |

## Application Design

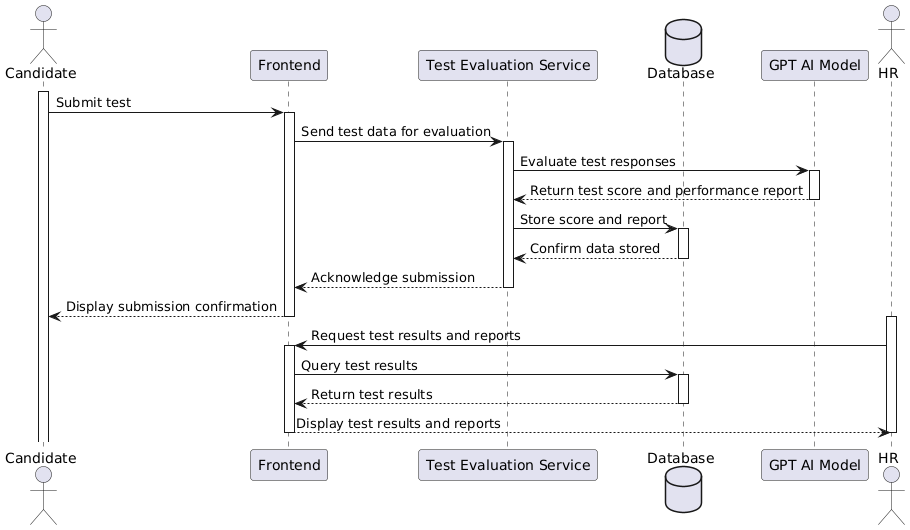
### Sequence Diagram

#### CV Upload and Analysis

**

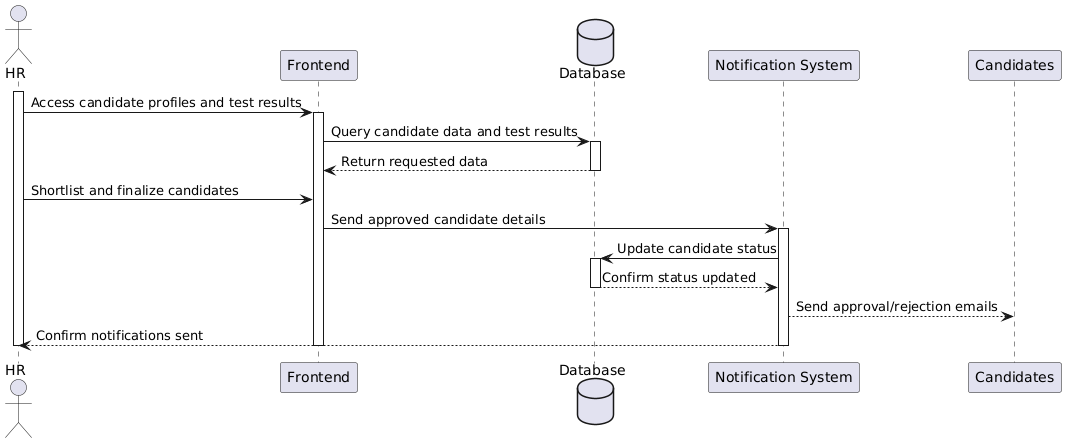
*The candidate starts by uploading the CV through the frontend, which forwards the file to the CV Analysis Service. The service processes the CV using the SBERT AI model to extract relevant skills, qualifications, and experiences. It then calculates a suitability score for the candidate. The extracted data and score are stored in the database for future use. HR professionals can later request and view the suitability score and extracted information via the frontend, enabling them to assess the candidate’s relevance to the job description.*

#### Test Submission and Evaluation

**

*The candidate submits their completed test through the frontend, which forwards the test data to the Test Evaluation Service. The service sends the test data to the GPT AI model for evaluation, which calculates the candidate’s score and generates a short performance report. The results and report are stored in the database for future access by HR. The system acknowledges the submission by displaying a confirmation message to the candidate, and the test results are made available to HR for decision-making.*

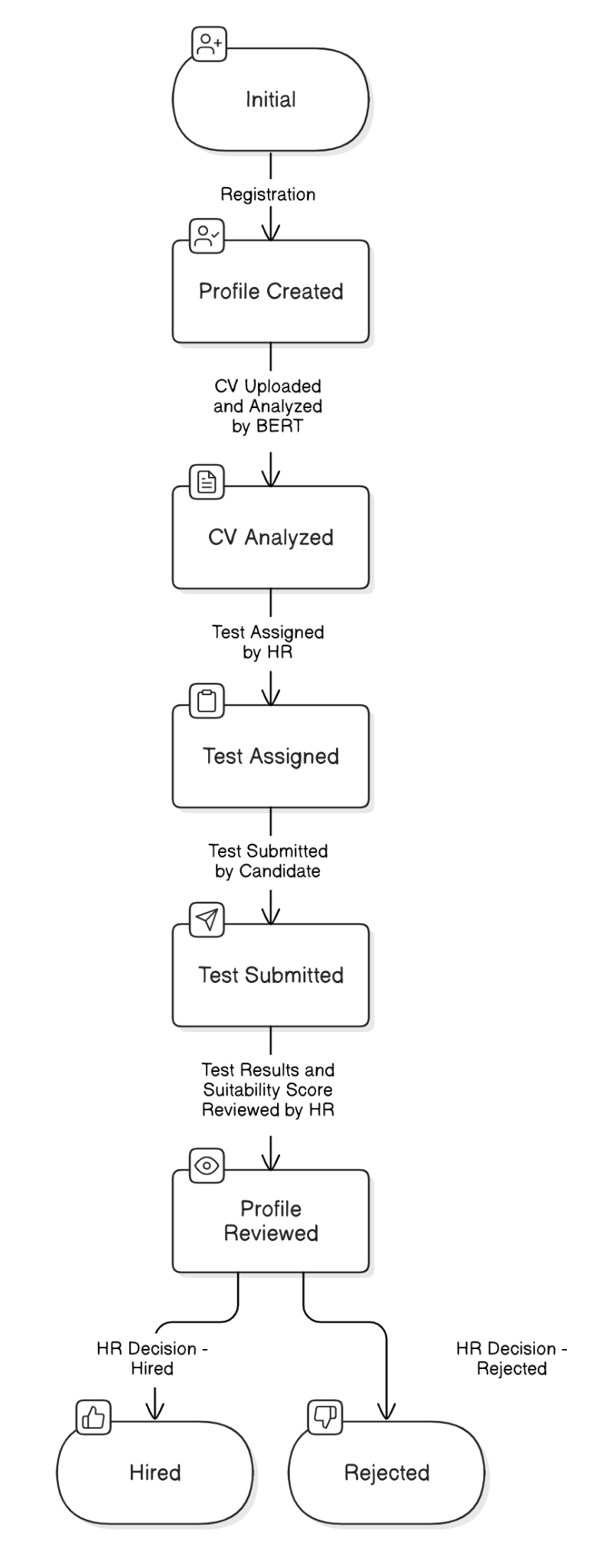
#### Final Candidate Selection

**

*HR professionals review the suitability scores, test results, and performance reports via the frontend. They shortlist and finalize the candidates for a job, sending the details to the notification service. The notification service updates the database with the candidates’ statuses (hired or rejected) and sends emails to notify the candidates. The system confirms email delivery to HR, ensuring the process is complete. This sequence reflects the core purpose of the system: enabling HR to efficiently manage and finalize candidate selection.*

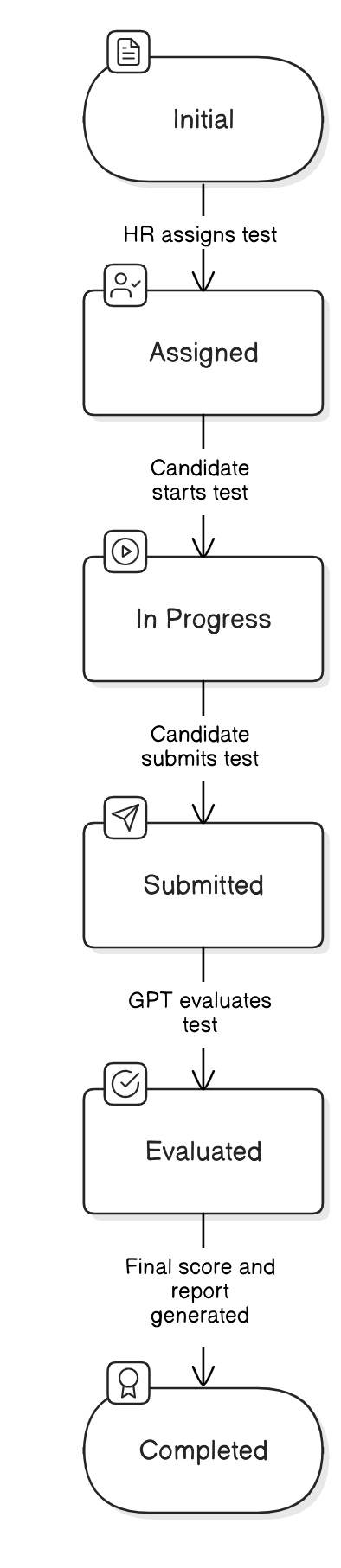
### State Diagram

#### Candidate Profile State Diagram

**

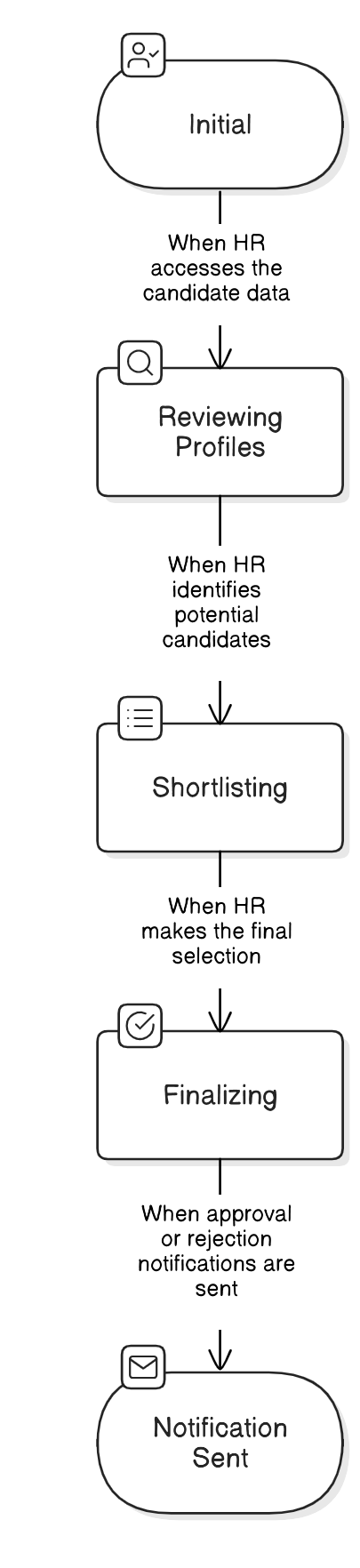
*the lifecycle of a candidate within the system, from the initial registration to the final decision of being hired or rejected. Initially, the candidate registers and creates a profile in the system. Once the profile is created, the candidate uploads their CV, transitioning to a state where the CV is analyzed using SBERT to generate a suitability score. After the analysis, HR assigns a test to the candidate, and the candidate subsequently completes and submits the test. The system evaluates the test results, and HR reviews the profile along with the test performance. Finally, based on HR’s decision, the candidate profile transitions to either a "Hired" or "Rejected" state.*

#### Test State Diagram



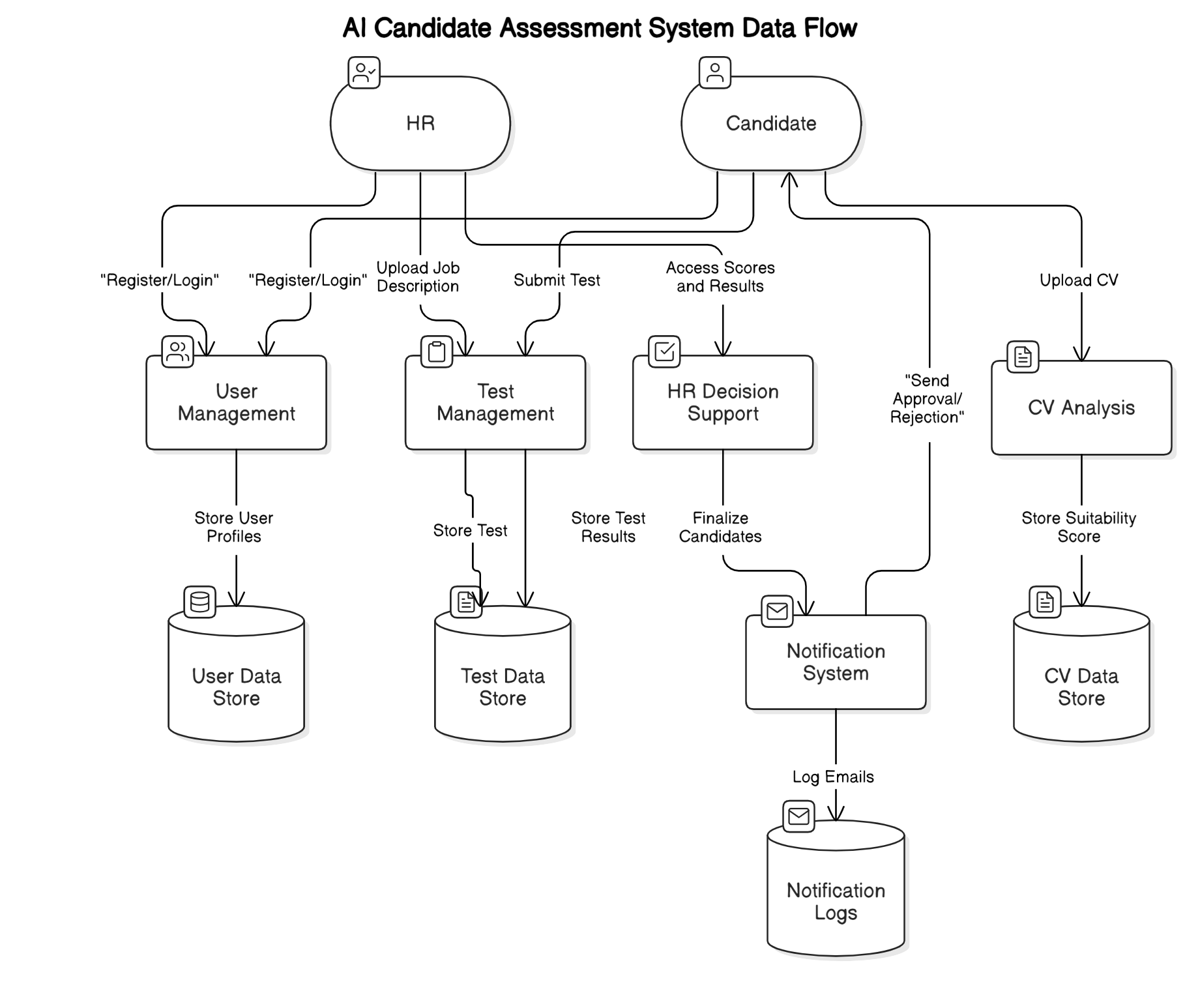
The test begins in an "Initial" state when created based on a specific job description by HR. It then transitions to an "Assigned" state once it is linked to a particular candidate. As the candidate begins the test, it moves to the "In Progress" state, reflecting active participation. Upon completion, the test enters the "Submitted" state, where the system processes it for evaluation using GPT. After evaluation, the test transitions to an "Evaluated" state, where the system generates a score and a performance report. The final state, "Completed," indicates that the test lifecycle is finished, with all results stored and accessible for review.

#### HR Decision Process State Diagram

**

*The process starts when HR logs into the system, transitioning to a "Reviewing Profiles" state where they assess candidates’ CV analysis scores and test results. Based on these assessments, HR shortlists potential candidates, moving to the "Shortlisting" state. From the shortlist, HR proceeds to finalize the selection of the top candidates, transitioning to the "Finalizing" state. The process concludes in the "Notification Sent" state, where the system sends emails to the selected candidates, notifying them of their approval or rejection.*

### 5.1.5 Data Flow Diagram



# References

**BERT Model Documentation**

* **Source**: Official AI Model Documentation from Hugging Face
* **URL**: https://huggingface.co/docs/
* **Publisher**: Hugging Face
* **Description**: Provided guidance BERT

**SBERT Model Documentation**

* **Source**: Official AI Model Documentation from Hugging Face
* **URL**: https://huggingface.co/sentence-transformers
* **Publisher**: Hugging Face
* **Description**: Provided guidance on integrating SBERT working

**Applying BERT-Based NLP for Automated Resume Screening and Candidate Ranking**

* **Source**: Applying BERT-Based NLP for Automated Resume Screening and Candidate Ranking
* **URL**: [https://www.researchgate.net/publication/378829570\_Applying\_SBERT-Based\_NLP\_for\_Automated\_Resume\_Screening\_and\_Candidate\_Ranking](https://www.researchgate.net/publication/378829570_Applying_BERT-Based_NLP_for_Automated_Resume_Screening_and_Candidate_Ranking)
* **DOI**: 10.1007/s40745-024-00524-5
* **Publisher**: Springer
* **Year**: 2024

**Augmented SBERT: Data Augmentation Method for Improving Bi-Encoders for Pairwise Sentence Scoring Tasks**

* **Source**: NAACL 2021
* **URL**: <https://arxiv.org/abs/2010.08240>
* **Publisher**: Accepted at NAACL 2021
* **Year**: 2021

**Resume Parsing using Natural Language Processing**

* **Source**: Grenze scientific Society
* **URL**: https://thegrenze.com/pages/servej.php?fn=587.pdf&name=Resume%20Parsing%20using%20Natural%20Language%20Processing&id=1418&association=GRENZE&journal=GIJET&year=2023&volume=9&issue=1
* **Publisher**: Grenze International Journal of Engineering and Technology
* **Year**: 2021

# Appendices

### *7.1 Supporting Materials*

* *Sequence diagrams: Representing workflows like registration, CV analysis, test submission, and HR decision-making.*
* *State diagrams: Depicting object life cycles for core functionalities.*
* *Class diagram: Overview of system entities and their relationships.*