**A purple square with a pen and a notebook

AI-generated content may be incorrect.**

**Software Design Descriptions**

Prepared By:

|  |  |  |  |
| --- | --- | --- | --- |
| **Member** | **ID#** | **Program** | **Signature** |
| Coral S. Schmidt Montilla | #148830 | Computer Science |  |
| Jaime Cuebas | #121703 | Computer Engineering |  |
| Christopher Ortiz | #130607 | Computer Science |  |
| Christian Brito | #9734 | Computer Science |  |
| Taishali N. Jimenez Quinones | #148017 | Computer Engineering |  |

**REVISION HISTORY**

The purpose of the table is to record version number, version date, name of the person making the change and a short revision description.

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Dates** | **Version** | **Task** |
| Taishali N. Jimenez Quinones | 10/09/2024 | 1 | Fill out the information |
| Taishali N. Jimenez Quinones | 10/11/2024 | 1.1 | Fill out the information |
| Taishali N. Jimenez Quinones | 10/16/2024 | 1.2 | Fill out the information |
| Coral S. Schmidt Montilla | 4/22/2025 | 1.3 | Review and update information |
| Coral S. Schmidt Montilla | 4/29/2025 | 1.4 | Review and update information |
| Coral S. Schmidt Montilla | 5/4/2025 | 1.5 | Updated our logo. |
| Coral S. Schmidt Montilla | 5/6/2025 | 1.6 | Finish updating document |

**Table of Contents**

[1. Introduction 5](#_Toc197435359)

[1.1 Purpose 5](#_Toc197435360)

[1.2 Scope 5](#_Toc197435361)

[1.3 Definitions, Acronyms and Abbreviations 5](#_Toc197435362)

[2. References 6](#_Toc197435363)

[3. Decomposition Description 7](#_Toc197435364)

[3.1 Module Decomposition 7](#_Toc197435365)

[3.1.1 Home Page Module Decomposition 7](#_Toc197435366)

[3.1.2 Registration Screen Decomposition 8](#_Toc197435367)

[3.2 Concurrent Process Decomposition 9](#_Toc197435368)

[3.2.1 User Registration and Log In 10](#_Toc197435369)

[3.3 Data Decomposition 10](#_Toc197435370)

[3.3.1 User Entity 10](#_Toc197435371)

[3.3.2 Profile Entity 11](#_Toc197435372)

[3.3.3 ResumeBuilder Entity 11](#_Toc197435373)

[3.3.4 Education Entity 11](#_Toc197435374)

[3.3.5 Experience Entity 11](#_Toc197435375)

[3.3.6 Responsibilities Entity 11](#_Toc197435376)

[3.3.7 Skills Entity 12](#_Toc197435377)

[3.3.8 Certification Entity 12](#_Toc197435378)

[3.3.9 Projects Entity 12](#_Toc197435379)

[3.3.10 ProjectDescription Entity 12](#_Toc197435380)

[3.3.11 ProjectSkills Entity 12](#_Toc197435381)

[3.3.12 Type Entity 12](#_Toc197435382)

[3.3.13 TypeRef Entity 13](#_Toc197435383)

[4. Dependencies Description 13](#_Toc197435384)

[4.1 Intermodule Dependencies 13](#_Toc197435385)

[4.2 Interprocess Dependencies 14](#_Toc197435386)

[4.3 Data Dependencies 22](#_Toc197435387)

[5. Interface Description 23](#_Toc197435388)

[5.1 User Interfaces 23](#_Toc197435389)

[5.1.1 State Diagram for User Interfaces 29](#_Toc197435390)

[5.2 Administrator Interfaces 30](#_Toc197435391)

[5.2.1 State Diagram for Administrator Interfaces 31](#_Toc197435392)

[6. Detailed Design 32](#_Toc197435393)

[6.2 Data Detailed Design 33](#_Toc197435394)

**List of Tables**

[Table 1: Definitions 5](#_Toc197435395)

[Table 2: Acronyms and Abbreviations 6](#_Toc197435396)

[Table 3: Intermodule Dependencies 14](#_Toc197435397)

**List of Figures**

[Figure 1: Home Page Module Decomposition 7](#_Toc197435418)

[Figure 2: Registration Screen Module Decomposition 8](#_Toc197435419)

[Figure 3: Log in Screen Module Decomposition 9](#_Toc197435420)

[Figure 4: User Registration and Log In 10](#_Toc197435421)

[Figure 5: Intermodule Dependencies 13](#_Toc197435422)

[Figure 6: Home Page State Diagram 24](#_Toc197435423)

[Figure 7: Main Page State Diagram 24](#_Toc197435424)

[Figure 8: Resume Builder State diagram 25](#_Toc197435425)

[Figure 9: Resume Builder Page 29](#_Toc197435426)

[Figure 10: User Interface 30](#_Toc197435427)

[Figure 11: Administrator Interface 31](#_Toc197435428)

[Figure 12: State diagram of Administrator Interface 31](#_Toc197435429)

[Figure 13: Sequence Diagram Log in User and Administrator 32](#_Toc197435430)

[Figure 14: Sequence Diagram Main Page 33](#_Toc197435431)

[Figure 15: Detailed Design 34](#_Toc197435432)

# 1. Introduction

## 1.1 Purpose

The purpose of this document is to create an outline of the design of our Resume Builder. This application will allow the user to navigate through the process of creating a resume step by step.

## 1.2 Scope

This document will cover the design of the Resume Builder website starting by discussing the module decomposition, the dependencies, and the user interface.

## 1.3 Definitions, Acronyms and Abbreviations

The following table illustrates the series of definitions, acronyms and abbreviations that are necessary to be aware of for proper understanding of this document.

|  |  |
| --- | --- |
| **Term** | **Definition** |
| Resume | Formal document that displays a person’s professional background and skills. |
| Entity | An entity is a thing, person, place or object that has an independent existence and can be uniquely identified in a database |
| Attribute | Each entity has specific attributes that provide more information about it. |
| Module | A module is an extension to a main program dedicated to a specific function. |

Table 1: Definitions

|  |  |
| --- | --- |
| **Acronym or Abbreviation** | **Term** |
| UI | User Interface |
| DB | Database |

Table 2: Acronyms and Abbreviations

# 2. References

The following references are cited using the IEEE format. These references are for technical words in this document.

[1] Pressman, R. (2014). *Software Engineering: A Practitioner’s Approach*. New York: McGraw Hill.s Design--Software Design Descri

[2] Gustafson, D. (2002). *Schaum's Outline of Software Engineering.* McGraw Hill Professional

# 3. Decomposition Description

The website will be divided into several modules to allow the creation of a resume easier while dividing into a step-by-step process. Each module will handle a specific task, like user registration, form submission, and allow to generate a PDF.

## 3.1 Module Decomposition

This section will describe and break down the duties and responsibilities of each module.

### 3.1.1 Home Page Module Decomposition

A diagram of a website

AI-generated content may be incorrect.The home page module serves as the entry point for users to begin their resume building process. With just a click of their button, the user will be able to navigate towards the module which will begin the step-by-step process of building a resume.

Figure 1: Home Page Module Decomposition

### 3.1.2 Registration Screen Decomposition

The registration screen module allows users to create an account by providing basic information like their username, email, and password. This will make sure the user has safe experience within our website. A diagram of a sign up button

AI-generated content may be incorrect.

Figure 2: Registration Screen Module Decomposition

## 3.2 Concurrent Process Decomposition

A diagram of a computer program

AI-generated content may be incorrect.This section will outline the concurrent process that runs with this application like the user login information.

Figure 3: Log in Screen Module Decomposition

### 3.2.1 User Registration and Log In

A screenshot of a login screen

AI-generated content may be incorrect.A screenshot of a sign up form

AI-generated content may be incorrect.The user registration and login module will be accessible within one module. Once the user clicks on the “Sign up here” link, the user will then go over to the registration module mentioned in figure 2.

Figure 4: User Registration and Log In

## 3.3 Data Decomposition

Data decomposition is essential for breaking down the data entities used in the resume builder application. These entities allow for organizing the user's information and storing resume-related content in a structured way.

### 3.3.1 User Entity

The User entity will store general information about the user such as their name, username, password, email, and contact details. It also keeps track of the user’s creation date, last login, and status. The User entity is connected to both the Profile and ResumeBuilder entities using foreign keys.

### 3.3.2 Profile Entity

The Profile entity stores personal information related to the user including their first and last name, email address, birth date, and age. It is directly linked to the User entity through a foreign key (UserId).

### 3.3.3 ResumeBuilder Entity

The ResumeBuilder entity serves as the central point for organizing all resume-related data. It is linked to both the User and Profile entities and will serve as the reference for other related entities like Education, Experience, Skills, and Projects.

### 3.3.4 Education Entity

The Education entity holds attributes related to the user's academic background including institution name, degree, field of study, GPA, years attended, and honors. This entity uses ResumeBuilderId as a foreign key.

### 3.3.5 Experience Entity

The Experience entity stores information related to the user's work experience such as company name, job title, location, and dates. It includes a foreign key that links to the ResumeBuilder entity.

### 3.3.6 Responsibilities Entity

The Responsibilities entity stores detailed descriptions of tasks and roles performed in a specific job. It includes a foreign key that links to the Experience entity.

### 3.3.7 Skills Entity

The Skills entity allows the user to input specific skills, including their proficiency level and category. It includes a foreign key that links it to the ResumeBuilder entity. This entity is also referenced by ProjectSkills to associate skills with projects.

### 3.3.8 Certification Entity

The Certification entity stores information regarding any certifications the user has, including issuing organization, issue date, and expiration date. This entity is currently not directly linked to ResumeBuilder.

### 3.3.9 Projects Entity

The Projects entity stores information about the user's projects, such as project name, description, tools used, start and end dates, and location. It is connected to the ResumeBuilder entity via a foreign key.

### 3.3.10 ProjectDescription Entity

The ProjectDescription entity allows for multiple written descriptions for a single project. It includes a foreign key that connects it to the Projects entity.

### 3.3.11 ProjectSkills Entity

The ProjectSkills entity serves as a bridge between the Projects and Skills entities. It connects specific skills to specific projects, allowing the user to associate relevant skills to each project.

### 3.3.12 Type Entity

The Type entity is used to group and classify information within the application. It includes information such as name, description, creation date, and status.

### 3.3.13 TypeRef Entity

The TypeRef entity provides values that are related to the Type entity. It contains reference codes and descriptions, and it includes a foreign key linking it to the Type entity.

# 4.Dependencies Description

The dependencies descriptions will include all the dependencies used for the entities and how they are interrelated.

## 4.1 Intermodule Dependencies

A diagram of a website

AI-generated content may be incorrect.The following table outlines the intermodule dependencies for the entities required in the process of building a resume. These relationships are based on the foreign key structure defined in the database schema and illustrate how the different entities interact with each other.

Figure 5: Intermodule Dependencies

| **Intermodule** | **Dependency** |
| --- | --- |
| User ↔ Profile | One-to-one relationship through UserId |
| User ↔ ResumeBuilder | One-to-many relationship via UserId |
| Profile ↔ ResumeBuilder | One-to-one relationship via ProfileId |
| ResumeBuilder ↔ Education | One-to-many relationship via ResumeBuilderId |
| ResumeBuilder ↔ Experience | One-to-many relationship via ResumeBuilderId |
| ResumeBuilder ↔ Skills | One-to-many relationship via ResumeBuilderId |
| ResumeBuilder ↔ Projects | One-to-many relationship via ResumeBuilderId |
| Experience ↔ Responsibilities | One-to-many relationship via ExperienceId |
| Projects ↔ ProjectDescription | One-to-many relationship via ProjectId |
| Projects ↔ ProjectSkills | One-to-many relationship via ProjectId |
| ProjectSkills ↔ Skills | Many-to-one relationship via SkillId |
| Type ↔ TypeRef | One-to-many relationship via TypeId |

Table 3: Intermodule Dependencies

## 4.2 Interprocess Dependencies

Interprocess dependencies refer to the internal relationships and interactions between various components of the resume builder application, including the processes involved in managing and structuring the application's database.

As part of the database setup process, the following script is used to initialize the application's data model. It ensures that all necessary tables are created in the correct order and any pre-existing tables are dropped to avoid conflicts during the initial deployment.

This setup script performs the following operations:

1. Drops all existing tables related to the resume builder if they already exist, to prevent duplication or conflicts.
2. Creates each required table with its respective fields, constraints, and foreign key relationships.
3. Establishes dependencies among entities based on foreign key constraints to enforce referential integrity.

This process ensures the database starts from a clean state and reflects the structure required for the application’s functionality.

The script includes the creation of the following entities:

* Users – stores login and account data.
* Profile – stores personal user details.
* ResumeBuilder – central resume hub linked to Profile and User.
* Education, Experience, Skills, Projects – sections of the resume linked via ResumeBuilderId.
* Responsibilities, ProjectDescription, ProjectSkills – support tables that further break down Experience and Projects.
* Certification – standalone entity for listing credentials.
* Type and TypeRef – classification entities to support system-wide categorization.

This process is essential for development and testing environments to ensure consistency and to help align the application logic with the database schema.

|  |
| --- |
| -- Drop tables if they already exist (optional in dev)  DROP TABLE IF EXISTS ProjectSkills;  DROP TABLE IF EXISTS ProjectDescription;  DROP TABLE IF EXISTS Project;  DROP TABLE IF EXISTS Responsibilities;  DROP TABLE IF EXISTS Experience;  DROP TABLE IF EXISTS Education;  DROP TABLE IF EXISTS Download;  DROP TABLE IF EXISTS Skill;  DROP TABLE IF EXISTS Template;  DROP TABLE IF EXISTS TypeRef;  DROP TABLE IF EXISTS Type;  DROP TABLE IF EXISTS Certification;  DROP TABLE IF EXISTS Profile;  DROP TABLE IF EXISTS ResumeBuilder;  DROP TABLE IF EXISTS [User];  -- User Table  CREATE TABLE [User] (  Id INT PRIMARY KEY,  Name VARCHAR(100),  Username VARCHAR(50) UNIQUE,  Password VARCHAR(255),  Email VARCHAR(100) UNIQUE,  MobilePhone VARCHAR(20),  External\_Id VARCHAR(50),  Creation\_Date DATE,  Last\_Login DATE,  Is\_Active BOOLEAN  );  -- ResumeBuilder Table  CREATE TABLE ResumeBuilder (  Id INT PRIMARY KEY,  UserId INT,  ResumeName VARCHAR(255),  FOREIGN KEY (UserId) REFERENCES [User](Id)  );  -- Profile Table  CREATE TABLE Profile (  Id INT PRIMARY KEY,  FirstName VARCHAR(100),  LastName VARCHAR(100),  MiddleName VARCHAR(100),  Email VARCHAR(100),  BirthDate DATE,  Address VARCHAR(255),  LinkedInProfile VARCHAR(255),  HighSchool VARCHAR(255),  College VARCHAR(255),  FinishedEducation VARCHAR(255),  HighestEducation VARCHAR(255),  ResumeBuilderId INT,  FOREIGN KEY (ResumeBuilderId) REFERENCES ResumeBuilder(Id)  );  -- Education Table  CREATE TABLE Education (  Id INT PRIMARY KEY,  ResumeBuilderId INT,  InstitutionName VARCHAR(255),  Degree VARCHAR(255),  FieldOfStudy VARCHAR(255),  StartDate DATE,  EndDate DATE,  YearName VARCHAR(10),  GPA DECIMAL(3,2),  Honors VARCHAR(255),  Location VARCHAR(255),  IsHighSchoolOrGED BOOLEAN,  IsAssociatesDegree BOOLEAN,  IsBachelorsDegree BOOLEAN,  IsMastersDegree BOOLEAN,  IsDoctoratesDegree BOOLEAN,  ProfileId INT,  FOREIGN KEY (ResumeBuilderId) REFERENCES ResumeBuilder(Id),  FOREIGN KEY (ProfileId) REFERENCES Profile(Id)  );  -- Experience Table  CREATE TABLE Experience (  Id INT PRIMARY KEY,  ResumeBuilderId INT,  CompanyName VARCHAR(255),  JobTitle VARCHAR(255),  StartDate DATE,  EndDate DATE,  Responsibilities TEXT,  Location VARCHAR(255),  ResponsibilityId INT,  FOREIGN KEY (ResumeBuilderId) REFERENCES ResumeBuilder(Id)  );  -- Responsibilities Table  CREATE TABLE Responsibilities (  Id INT PRIMARY KEY,  Description TEXT,  ExperienceId INT,  FOREIGN KEY (ExperienceId) REFERENCES Experience(Id)  );  -- Skill Table  CREATE TABLE Skill (  Id INT PRIMARY KEY,  ResumeBuilderId INT,  SkillName VARCHAR(100),  ProficiencyLevel VARCHAR(50),  SkillsCategory VARCHAR(50),  FOREIGN KEY (ResumeBuilderId) REFERENCES ResumeBuilder(Id)  );  -- Project Table  CREATE TABLE Project (  Id INT PRIMARY KEY,  ResumeBuilderId INT,  ProjectName VARCHAR(255),  Skills TEXT,  IDEUsed VARCHAR(100),  Description TEXT,  FOREIGN KEY (ResumeBuilderId) REFERENCES ResumeBuilder(Id)  );  -- ProjectDescription Table  CREATE TABLE ProjectDescription (  Id INT PRIMARY KEY,  ProjectId INT,  Description TEXT,  FOREIGN KEY (ProjectId) REFERENCES Project(Id)  );  -- ProjectSkills Table  CREATE TABLE ProjectSkills (  Id INT PRIMARY KEY,  ProjectId INT,  SkillId INT,  FOREIGN KEY (ProjectId) REFERENCES Project(Id),  FOREIGN KEY (SkillId) REFERENCES Skill(Id)  );  -- Certification Table  CREATE TABLE Certification (  Id INT PRIMARY KEY,  CertificationName VARCHAR(255),  IssuingOrganization VARCHAR(255),  IssueDate DATE,  ExpirationDate DATE  );  -- Download Table  CREATE TABLE Download (  Id INT PRIMARY KEY,  ResumeBuilderId INT,  VersionId INT,  TemplateId INT,  UserId INT,  IsFinished BOOLEAN,  UpdatedBy VARCHAR(100),  UpdatedOn DATE,  CreatedBy VARCHAR(100),  CreatedOn DATE,  FOREIGN KEY (ResumeBuilderId) REFERENCES ResumeBuilder(Id),  FOREIGN KEY (TemplateId) REFERENCES Template(Id),  FOREIGN KEY (UserId) REFERENCES [User](Id)  );  -- Type Table  CREATE TABLE Type (  Id INT PRIMARY KEY,  Name VARCHAR(100),  Description VARCHAR(255),  CreatedBy VARCHAR(100),  CreatedOn DATE,  UpdatedBy VARCHAR(100),  UpdatedOn DATE,  IsActive BOOLEAN  );  -- TypeRef Table  CREATE TABLE TypeRef (  Id INT PRIMARY KEY,  TypeId INT,  Name VARCHAR(100),  Description VARCHAR(255),  Code VARCHAR(50),  CreatedBy VARCHAR(100),  CreatedOn DATE,  UpdatedBy VARCHAR(100),  UpdatedOn DATE,  Sorting INT,  IsActive BOOLEAN,  FOREIGN KEY (TypeId) REFERENCES Type(Id)  );  -- Template Table  CREATE TABLE Template (  Id INT PRIMARY KEY,  Name VARCHAR(100),  PreviewImage VARCHAR(255),  HtmlContent TEXT,  CreatedOn DATE,  IsActive BOOLEAN  ); |

Note:  
This script is not intended to be executed in a production environment without proper migration procedures, as it deletes existing data. Future changes to the schema should be managed through version-controlled migration scripts.

## 4.3 Data Dependencies

The Resume Builder application is composed of multiple entities that are linked together through foreign key relationships to ensure consistency and accurate data representation across all resume sections.

The User entity stores basic account information and is directly connected to the ResumeBuilder entity through the UserId foreign key. The ResumeBuilder entity acts as the central point of reference for the resume and is also linked to the Profile entity, which contains detailed personal and contact information.

The Profile entity is further referenced by the Education entity through the ProfileId foreign key, allowing education data to be associated with specific profile details. Additionally, education records also reference the ResumeBuilder entity using the ResumeBuilderId.

The Experience, Skill, Project, and Download entities are all linked directly to the ResumeBuilder entity using the ResumeBuilderId foreign key, enabling each resume to maintain its own set of experience entries, skills, projects, and download records.

The Experience entity connects to the Responsibilities entity through the ExperienceId foreign key, allowing each job entry to have multiple responsibility descriptions.

The Project entity is extended by the ProjectDescription entity, which supports storing multiple description segments for a single project. It is also connected to the ProjectSkills entity through the ProjectId, which serves as a bridge to associate specific Skill entries with each project.

The Download entity includes references to the User, ResumeBuilder, and Template entities to keep track of who generated the resume, what content was used, and which version it corresponds to.

Lastly, the Type and TypeRef entities represent classification types and reference values. TypeRef is directly linked to Type through the TypeId foreign key and is used to provide extensible categorization options throughout the application.

This entity relationship structure ensures that all resume-related data is properly organized, allowing users to build, customize, and download resumes with consistent and relational data across the system.

# 5. Interface Description

This section will describe the user and administrator interfaces within the Resume Builder application. These interfaces guide both types of users through the resume creation and management process. Diagrams illustrate how the application transitions between key states and how users interact with the system components.

## 5.1 User Interfaces

The user interface is a central part of the Resume Builder application. It enables users to create an account, log in, input personal information, and build their resume step by step using pre-defined templates.

The homepage allows users to get started quickly and navigate to the login or registration screens. Once logged in, users are guided through entering their profile details, education history, work experience, skills, certifications, and project data. They can preview their resume in real-time using available templates and download the final version as a PDF.

A diagram of a website

Description automatically generatedThese modules are designed to be clear, responsive, and user-friendly, ensuring an intuitive experience across devices.

Figure 6: Home Page State Diagram

A diagram of a computer program

Description automatically generated

Figure 7: Main Page State Diagram

A diagram of a button

AI-generated content may be incorrect.

Figure 8: Resume Builder State diagram

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.A screenshot of a computer

AI-generated content may be incorrect.

**A screenshot of a computer

AI-generated content may be incorrect.**Figure 9: Resume Builder Page

### 5.1.1 State Diagram for User Interfaces

The state diagram for user interfaces demonstrates how a user navigates through the application, beginning with account creation or login, and moving through each module required to build and download a resume. Each state reflects a part of the form-filling and editing process, concluding with the resume download feature.

A diagram of a website

Description automatically generated

Figure 10: User Interface

## 5.2 Administrator Interfaces

The administrator interface is designed to allow system administrators to monitor and manage user activity, resume entries, and the templates available for resume generation. This section of the application includes interfaces to review submitted data, activate or deactivate templates, and access audit information regarding resume generation history (from the Download entity).

Administrators can also manage classification types using the **Type** and **TypeRef** entities to control categories used in the app (e.g., skill categories, education levels, etc.).

A screenshot of a computer

Description automatically generated

Figure 11: Administrator Interface

### 5.2.1 State Diagram for Administrator Interfaces

The state diagram for the administrator interface illustrates the navigation available to administrators. This includes access to the dashboard, template management, and user activity logs. It also shows how administrators move between these screens to perform maintenance and configuration tasks.

A diagram of a button

Description automatically generatedFigure 12: State diagram of Administrator Interface

# 6. Detailed Design

This section will include the detailed design of the Resume Builder application. It describes the key modules, system behavior, and entity relationships that support resume creation and management.

**6.1 Module Detailed Design**

This section contains sequence diagrams for major functionalities within the Resume Builder application. These diagrams illustrate the interaction between the user and the system throughout various tasks.

* Login and Registration Module: This module allows users to create an account or log in using their credentials. Once authenticated, users are redirected to the main dashboard.
* Resume Builder Module: This module guides users through entering data into each section of the resume such as profile, education, experience, and skills.
* Template Selection Module: Users can preview available resume templates and choose one before downloading.
* *A screenshot of a computer

  Description automatically generated*PDF Generation Module: Once a resume is completed, this module handles the logic for converting the HTML-based resume into a downloadable PDF using the selected template and stored user data.

Figure 13: Sequence Diagram Log in User and Administrator

A diagram of a website

Description automatically generatedFigure 14: Sequence Diagram Main Page

## 6.2 Data Detailed Design

This section outlines the database structure supporting the Resume Builder application. The system is composed of interrelated entities including:

* User – stores login and contact info for authentication.
* ResumeBuilder – serves as the core link for each resume, tying together all related information.
* Profile, Education, Experience, Skill, Project, and Download – these entities capture structured information relevant to a resume.
* Responsibilities, ProjectDescription, and ProjectSkills – support detailed content under experience and projects.
* Template – stores available resume formats in HTML form for rendering and download.
* Type and TypeRef – used to organize data across the app by defining categories and selectable values.

A computer screen shot of a computer

AI-generated content may be incorrect.These entities follow a normalized relational structure that uses foreign keys to maintain integrity and allow precise filtering and association of user-specific data.

Figure 15: Detailed Design