****

**Software Design Description (SDD)**

Project: RECRUITMENT SOLUTION AGENCY (RSA)

Yazan Mohammed Amin

|  |  |  |
| --- | --- | --- |
| NAME | CONTACTNO | EMAIL |
| Yazan Mohammad Amin | 0790939430 | [yznbrhm77@gmail.com](mailto:yznbrhm77@gmail.com) |

**Cover Page**

**Revisions Page**

**Table of Contents**

**Table of Contents**

**1 INTRODUCTION………………………………………………3**

1.1 Design Overview………………………………………………………….3

1. 2 Requirements Traceability Matrix…………………………………………3

**2 SYSTEM ARCHITECTURAL DESIGN…………………………4**

2.1 Chosen System Architecture…………………………………………4

2.2 Discussion of Alternative Designs…………………………………………4

2.3 System Interface Description…………………………………………5

**3 DETAILED DESCRIPTION OF COMPONENTS………………5**

3.n Component-n…………………………………………6

**4 USER INTERFACE DESIGN………………………………….6**

4.1 Description of the User Interface…………………………………………6

4.1.1 Screen Images………………………………………………………………………………7

4.1.2 Objects and Actions………………………………………………………………………………7

**5 ADDITIONAL MATERIAL………………………………………..7**

**1 INTRODUCTION**

The **"Introduction"** section provides a foundational overview of the Recruitment Solution Agency project. It outlines the purpose and scope of the system, highlighting the key goals and objectives. This section offers readers a clear understanding of the context in which the agency operates and the challenges it aims to address. Additionally, it may touch upon the significance of efficient recruitment processes in today's competitive job market.

**1.1 Design Overview**

The **"Design Overview"** segment offers a high-level summary of the project's design approach. It outlines the design principles, methodologies, and philosophies that guide the creation of the Recruitment Solution Agency. This section provides insight into how design decisions align with the project's goals and intended user experience. It might also briefly mention the chosen architectural style and design considerations that contribute to the system's overall effectiveness.

**1.2 Requirements Traceability Matrix**

The **"Requirements Traceability Matrix"** establishes a clear link between the initial project requirements and the design decisions made throughout the development process. This matrix serves as a roadmap that ensures each requirement is addressed by the corresponding design component. By maintaining this traceability, the project team can verify that the final system fulfills all the defined objectives and meets the expectations of stakeholders.

**2 SYSTEM ARCHITECTURAL DESIGN**

In the **"System Architectural Design"** phase, the chosen architecture serves as the backbone of the Recruitment Solution Agency. This architecture is a carefully crafted structure that encompasses different modules and their interactions. It outlines how the user interface, the database, and the backend processes harmonize to create a seamless experience. By opting for this architecture, we are optimizing efficiency, scalability, and maintainability, crucial factors in managing a large volume of recruitment data effectively.

**2.1 Chosen System Architecture**

After evaluating various architecture options, the decision was made to adopt a multi-tier architecture. This architecture segregates the system into three main tiers: presentation, application logic, and data storage. This separation enhances flexibility and scalability by allowing updates to be made to individual tiers without affecting others. The presentation tier includes the user interface, ensuring a user-friendly experience. The application logic tier handles business processes, while the data storage tier manages the recruitment data efficiently.

**2.2 Discussion of Alternative Designs**

While considering alternative designs, several architectures were evaluated, including monolithic and micro services architectures. The monolithic approach seemed straightforward but could potentially hinder future scalability. On the other hand, the micro services architecture offered flexibility but might introduce complexity in the initial setup. The multi-tier architecture was deemed optimal due to its balance between manageability, scalability, and user experience.

**2.3 System Interface Description**

The system interface description outlines how the different architectural tiers communicate. The user interface layer interacts with the application logic layer via RESTful APIs, enabling seamless data exchange. This separation also permits front-end updates without affecting the backend processes. Moreover, the application logic layer interacts with the data storage layer through a well-defined data access layer, ensuring data integrity and security.

**3 DETAILED DESCRIPTION OF COMPONENTS**

**1. User interface**

The design of the user interface and the interactions the user will experience while browsing jobs and managing candidates will be described.

**2. Database**

We will provide details on how job and candidate information is organized and stored in the database.

**3. Back Component**

The logic and functions performed by the back-end component to support processes and procedures will be explained.

**4. Performance improvement**

We will briefly discuss how to achieve excellent system performance by implementing effective optimizations.

**5. Integration with External Services**

We will explain how the system integrates with external services to enhance and extend the functionality offered.

**3.n Component-n**

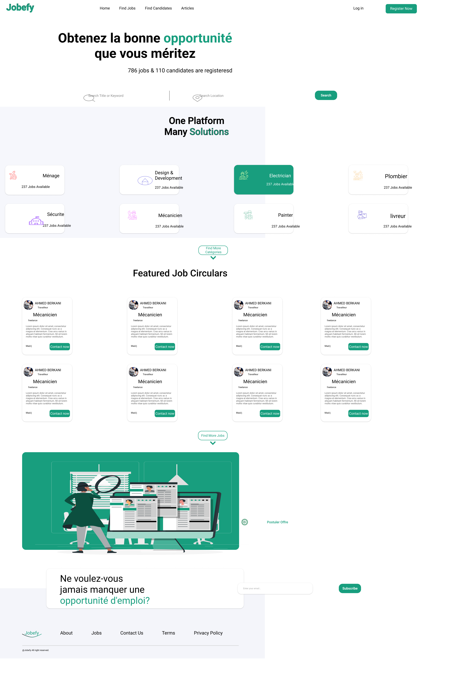
Is part of the Employment Solutions Agency system and performs a function Determine what function the component performs, eg managing jobs or processing candidates Interacts with components Decide how to interact such as using APIs or exchanging data Contributes to defining the benefit the component provides, eg improving user experience or facilitation of performance.

**4 USER INTERFACE DESIGN**

The User Interface (UI) Design component focuses on crafting an intuitive and visually appealing user experience. The UI is designed with user-centered principles, prioritizing easy navigation and clarity. A consistent design language is employed throughout the platform, enhancing brand recognition. The UI's responsive design ensures a seamless experience across different devices, catering to a diverse user base.

**4.1 Description of the User Interface**

The User Interface (UI) serves as the bridge between users and the system. It offers a dashboard that provides quick access to essential features. The navigation bar at the top ensures easy movement between different sections. Users can input search criteria, view job listings, and manage candidate profiles effortlessly. The UI employs a clean and modern aesthetic, enhancing user engagement and interaction.

**4.1.1 Screen Images**

**4.1.2 Objects and Actions**

**1. Objects:** These are the different items or entities that users interact with within the user interface. Examples of objects could include "Job Listings," "Candidate Profiles," "User Accounts," and "Application Forms."

**2. Actions:** These are the operations or activities that users can perform on the objects. Examples of actions could include "Viewing a Job Listing," "Editing Candidate Details," "Creating a User Account," and "Submitting an Application."

**5 ADDITIONAL MATERIAL**

**1. Flowcharts and Diagrams:** Provide visual representations of system processes, data flows, and interactions between components. Flowcharts, sequence diagrams, and entity-relationship diagrams can help clarify system behavior.

**2. Use Case Scenarios:** Describe specific scenarios or use cases to illustrate how different user roles interact with the system. This helps demonstrate how the system supports various tasks and workflows.

**3. Sample Screenshots:** Include screenshots or wireframes of key user interface screens to provide a visual representation of how the interface will look and the information it will display.

**4. Data Samples:** Provide example data entries for different components, such as sample job listings, candidate profiles, and user accounts. This helps illustrate how the system handles and presents data.

**5. External Services Documentation:** If the system integrates with external services or APIs, include relevant documentation or references to the APIs used for integration.

**6. References:** List any sources or references that were used to inform the design decisions in the document, such as best practices, industry standards, or research materials.

**7. Assumptions and Constraints:** If there are any assumptions or constraints that influenced the design decisions, document them here to provide context for readers.

**8. Glossary:** Include a glossary of technical terms or project-specific terminology to ensure a consistent understanding of terms throughout the document.