SOFTWARE ENGINEERING PROGRAM

SWENG 894 – CAPSTONE EXPERIENCE

Team 2

Software Requirement Specification

11 August 2024

Jack Taylor
Javier A. Rodriguez
Julio Aira IV

I - Concept of Operations (CONOPS)

1. CONOPS

The Penn State World Campus MSE program includes a culminating Capstone project. This document introduces the candidate concepts for further discussion and exploration with the professor by providing a brief description of our candidate project for the SWENG 894 Capstone. Sections in this document include approach, assumptions, mission statement, solution description, and the required estimated details: domain model, programming language and tool chain.

a. Approach

For this effort, we will be adopting an overarching strategy and approach that supports agile discovery and the ability to react to discoveries (both blockers and enablers). We term this approach Minimum Viable Product (MVP). This mindset complements the agile-like approach desired. In reviewing the capstone intention, the unique challenges of working as a completely independent team of one, and considering how to best complement our current career, the need to enable rapid innovation is paramount. The goal of each decision and effort will be to rapidly achieve the minimum viable outcomes. This means that we will be assuming and declaring different technical debt and adding new efforts to the project backlog for future prioritization.

Why? Some of the tasks of recruiters and other employees from the Human Resources department involve identifying potential candidates for open organizational positions based on their submitted resumes. The volume of resumes that require scanning by Human Resources employees can be overwhelming, especially in Small to Medium Enterprises (SMEs).

Our focus is enhancing the process of identifying candidates based on resumes submitted to job postings by developing an Automated Resume Screening Tool that will increase efficiency and accuracy.

b. Foundational Assumptions

- As students that actively apply to job postings, we are familiar with the problem domain from the candidate's perspective.
- However, our team does not have experience from the human resources team's perspective, therefore the associated business objectives and subsequent decomposition of scenarios will evolve.
- All languages and tools identified in this document are subject to change based on insight received by the professor.
- An agile-like approach will be leveraged. Given this is a group effort, we will be managing our effort and time following the sprints defined for the Capstone program.
    a) This supports the imperative for innovation and reacting to discoveries.
    b) This supports the CAPSTONE mandates for traditional/historic artifacts on a timed basis.

2. Introduction

The capstone project will be code-named *Automated Resume Screening Tool Development*.

a. Mission

Our capstone project's mission is to enhance the efficiency and accuracy of human resources departments by developing an Automated Resume Screening Tool. This tool will leverage advanced natural language processing technologies to intelligently analyze, categorize, and rank job applicant resumes based on relevant skills and experiences. By automating the initial stages of the resume review process, our solution aims to significantly reduce the manual effort involved in candidate screening, allowing HR professionals to focus more on engaging with potential hires rather than sifting through vast quantities of applications. This system will be designed to integrate seamlessly with existing HR workflows and will be scalable to accommodate different organizational needs. Our project will deliver a reliable, efficient, and user-friendly platform that empowers HR teams to make more informed and timely hiring decisions.

b. Background and Problem Statement

In today's rapidly evolving job market, human resources departments face significant challenges in managing large volumes of job applications efficiently. Traditional methods of resume screening are time-consuming and often lead to bottlenecks in the hiring process, affecting the overall speed and effectiveness of talent acquisition. The advent of digital technology has paved the way for automating many manual tasks, yet many organizations still rely on outdated methods for initial candidate evaluations. There is a growing need for innovative solutions that can integrate seamlessly into existing Human Resources systems and enhance the recruitment process by automating the screening of resumes, which would also benefit applicants since responses can be provided expeditiously.

Human resources departments across various industries struggle with the manual task of sifting through hundreds, sometimes thousands, of resumes for each job opening. This process is not only labor-intensive but also prone to human error and bias, which can lead to inconsistent and unfair candidate selection. Additionally, the delay in processing applications can result in losing top talent to competitors and increased operational costs. There is a crucial requirement for a tool that can automatically analyze and categorize resumes based on predefined criteria such as skills, experience, and educational background. Such a tool should leverage modern natural language processing techniques to ensure high accuracy and relevance in candidate selection, thereby streamlining the recruitment process, reducing time-to-hire, and enhancing the overall efficiency of HR operations.

c. Target audience

The target audience for our Automated Resume Screening Tool includes:

1. Human Resources Professionals: HR managers and recruiters who are directly involved in the hiring process will benefit from this tool by reducing the time and effort required to screen large volumes of resumes, allowing them to focus more on engaging with qualified candidates.
2. Small to Medium Enterprises (SMEs): SMEs often lack the extensive resources of larger corporations for managing extensive recruitment processes. This tool will enable them to handle recruitment more efficiently without needing to significantly expand their HR departments.

3. Large Corporations: Large companies that handle high volumes of applications, especially those with frequent hiring rounds, will find this tool beneficial for managing and streamlining their recruitment processes.
4. Staffing Agencies: These organizations can utilize the tool to improve their efficiency in matching candidates to job postings, thereby enhancing service delivery to their clients.
5. Educational Institutions' Career Services: Universities and colleges can use this tool to assist their graduates in finding suitable job placements by matching students' resumes with potential employers.
6. Technology Startups: Startups in rapid growth phases that need to scale their workforce quickly will benefit from an automated tool that helps them identify the best talents efficiently.

This diverse target audience will benefit from an enhanced, automated resume screening process that ensures faster and more accurate candidate assessment, thereby improving the overall recruitment workflow.

d. Key Features

The Automated Resume Screening Tool will have several key features designed to optimize and streamline the recruitment process for HR departments. Here are some of the key features:

1. Automated Skill Extraction: The tool will automatically identify and extract key skills and qualifications from resumes ensuring that resumes are analyzed thoroughly and accurately.
2. Customizable Screening Criteria: Users will be able to set specific criteria based on job requirements, such as certain skills, years of experience, or educational qualifications, to help the system categorize and prioritize candidates effectively.
3. Resume Ranking and Scoring: Each resume will be scored and ranked based on how well it matches the job criteria set by the recruiter. This feature helps in quickly identifying the most suitable candidates.
4. Integration with Existing HR Systems: The tool will be designed to seamlessly integrate with existing HR management systems to provide a unified experience and reduce the need for multiple platforms.
5. Dashboard for Recruitment Analytics: An interactive dashboard will provide HR professionals with insights into the recruitment process, such as the number of resumes processed, breakdown of skills across the candidate pool, and tracking of recruitment stages.
6. Data Privacy and Security: Ensuring the confidentiality and integrity of candidate data will be a priority, with features adhering to industry-standard data protection regulations.
7. Multi-Format Support: The tool will be capable of parsing resumes from various formats, including PDF, DOCX, and text files, to accommodate different submission types.
8. Feedback Loop for Continuous Improvement: The system will include a mechanism for HR users to provide feedback on the tool's recommendations, which will be used to continually refine and improve the machine learning models behind the scenes.
9. Cloud-Based Accessibility: Being cloud-based, the tool will allow HR teams to access the system from anywhere, facilitating remote recruitment processes and collaboration among team members.

10. Automatic Updates and Maintenance: With cloud deployment, the system will automatically receive updates and maintenance, ensuring that it remains efficient and secure with minimal downtime.

These features are designed to make the recruitment process more efficient, reduce administrative burden, and improve the quality of hiring decisions.

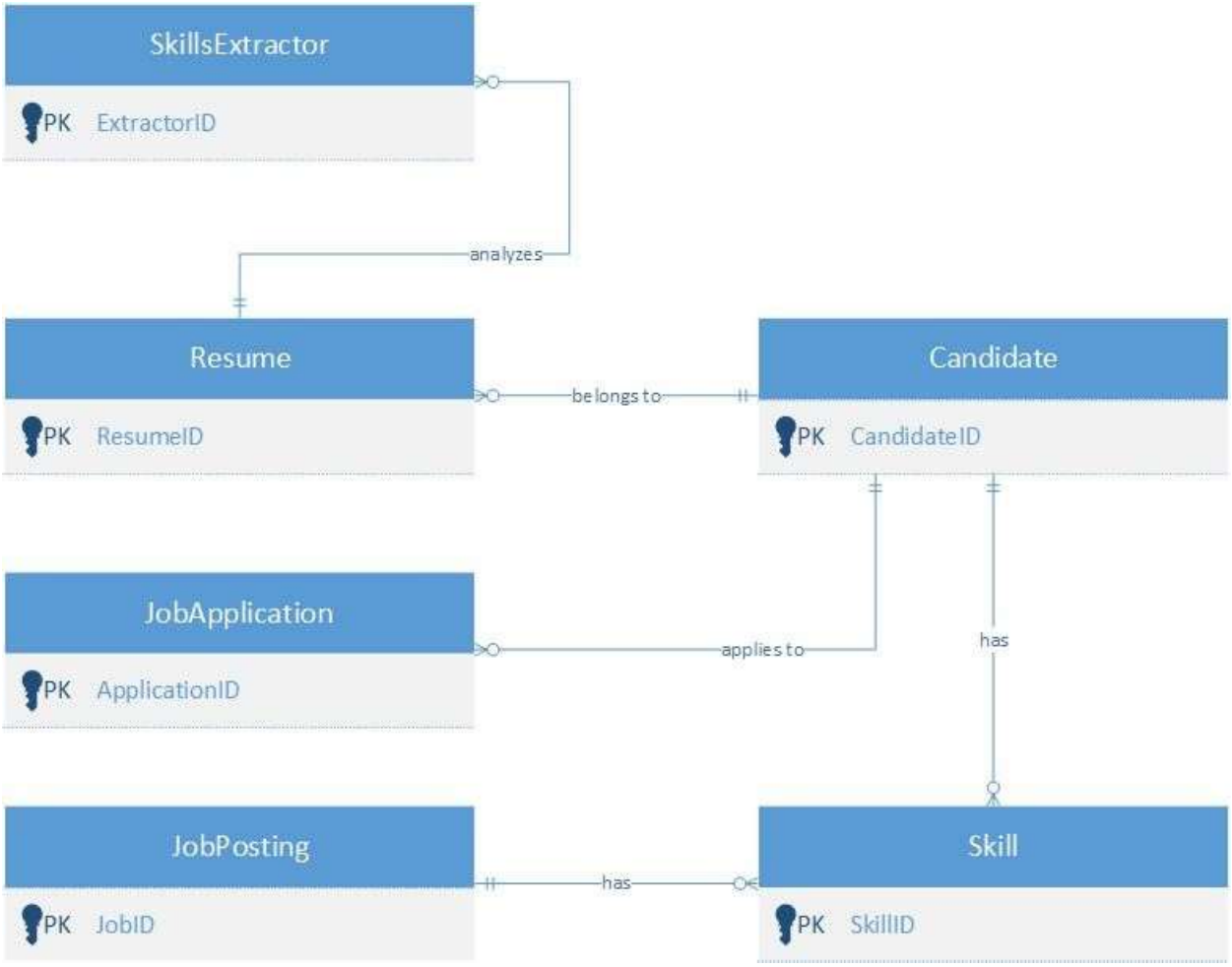
e. Additional Considerations

This effort should leverage commercial-off-the-shelf (COTS) technology, cloud-native capabilities, and open source to support rapid delivery. To the extent possible, managed cloud capabilities will be leveraged to reduce operational drag and accelerate the delivery of features and subsequent value to the target audience.

3. Domain Model

This domain model is intentionally high-level and conceptual and will evolve throughout this capstone. Full relationship modeling has not been applied in the essence of time. This will evolve during the coming weeks

- For simplicity, attributes have been limited
- Noun/Noun-phrases represent objects in the solution domain
- Vern/Verb-Verb phrases, relationships, and actions
- Candidate is linked to Resume through a one-to-many relationship, as one candidate can have multiple resumes.
- JobPosting is connected to JobApplication, which in turn is connected to Candidate, representing the applications made by candidates to specific job openings.
- Skill is a central entity linking to both Candidate and JobPosting through many-to-many relationships, indicating the skills of candidates and the skills required by job postings.
- SkillsExtractor works directly on Resume, analyzing the text and populating the Skill entities associated with a Candidate or required by a JobPosting.



a. Programming Language(s)

The following list of languages is representational and may change based on discoveries during early prototyping and innovation. Supporting libraries are not selected in their totality.

Purpose	Language	Rationale
Front End	- TypeScript with React	- TypeScript offers strong typing which helps in building more robust and error-free code. It enhances code quality and understandability, which is vital when working in team environments. - React is widely used for building user interfaces, particularly due to its component-based architecture which allows for reusable UI

		components. This makes it a great choice for the dynamic elements needed in your dashboard for displaying analytics from the resume screening. -The combination of React and TypeScript ensures that the application is scalable, maintainable, and can handle complex states and user interactions efficiently.
Back End	- Python with Flask	- Simplicity and readability, making it ideal for rapid development. It has a strong ecosystem for machine learning and natural language processing, which will be beneficial for implementing features like resume parsing and analysis. - Flask is a lightweight web framework that is easy to use and extends with numerous plugins available. It allows for easy setup of RESTful APIs which your front-end can communicate with.
Infrastructure As Code	Terraform	- Terraform is an open-source infrastructure as a code software tool that allows you to define both cloud and on-premises resources using a declarative configuration language. - Terraform facilitates automation and helps keep consistency of environments through its state management capabilities, which is essential for deploying and scaling cloud resources efficiently and reliably.

b. Toolset

A list of tools to be used in project development and management is provided below, with the accompanying rationale for the decisions. It is important to highlight that a stretch goal is to include automated DevOps capabilities into a CI/CD-like pipeline.

Type	Tool	Rationale
Integrated Development Environment (IDE)	Visual Code	- VS Code is widely favored for its robust features such as IntelliSense (code completion), debugging support, and native support for a wide range of programming languages including TypeScript and Python. - It offers extensive plugin support for both front-end and back-end development, enhancing productivity.
Project Management (PM)/ Agile Management	Jira	- Jira is specifically designed for agile project management. It supports Scrum and Kanban and integrates well with tools for continuous integration and deployment. - It offers powerful tools for tracking user stories, tasks, sprints, and reporting, making it ideal for managing complex software projects.
Source Code Repository	GitHub	- GitHub is the industry standard for version control and offers excellent support for collaboration features such as pull requests and code reviews. - It integrates seamlessly with most CI/CD pipelines and project management tools.
Defect Tracking	GitHub - Issues	- Integrated directly within GitHub, it allows for tracking bugs and issues alongside the

		codebase, linking commits to issues for traceability. - Simplifies managing project tasks and bugs without the need for an external tool.
Compiler	Babel (for TypeScript and JavaScript)	- Babel is a JavaScript compiler that allows you to use next-generation JavaScript, today. It can compile TypeScript into JavaScript ensuring browser compatibility. - Helps in using the latest features of JavaScript while ensuring compatibility with older browsers.
Unit Test Framework	Jest	- Jest is a delightful JavaScript Testing Framework with a focus on simplicity and support for large web applications. - Works out of the box for any React project and has great integrated support for TypeScript through Babel.
Functional/Integration Test Frameworks	Cypress	- Cypress provides end-to-end testing that's easy to set up and write. It's robust enough for complex testing scenarios. - It offers real-time reloads for test development and excellent debugging capabilities.
CI/CD	GitHub Actions	- GitHub Actions allows you to automate workflows directly from your GitHub repository, from simple CI tasks to complex CD pipelines. - It can be configured to run tests, deployments, and releases in response to GitHub events like push or pull requests.
Data Storage	IndexDB	- DynamoDB is a fast and flexible NoSQL database service for all applications that need

		consistent, single-digit millisecond latency at any scale. - It's a fully managed cloud database and supports both document and key-value store models.

II - Functional Requirements

1. Functional Requirements

The Automated Resume Screening Tool intends to meet requirements set by recruiters, and other employees from the Human Resources department, which identify need to simplify the process of identifying candidates for open positions. High-level goals are increasing efficiency and accuracy of hiring process by automating the screening of resumes submitted by candidates to job postings. Below are features identified through interviews with recruiters and other employees from the Human Resources department which were translated to requirements included in upcoming sections.

- Candidates are allowed to submit their resumes in various formats (e.g. .pdf, .doc…), rather than having to convert files to a specific format, users prefer the solution accepts upload of various file formats.

- Candidates' skills, educational background, and professional experience drive decision to extend and interview, therefore, users need the solution to clearly identify and capture these key performance indicators in a structured fashion.

- Ranking of resumes based candidates' skills, educational background, and professional experience is desired in order to select high ranking resumes for further review, avoiding spending too much time on resumes that are low ranking.

- Notification of high ranking resumes is requested to further expedite hiring process and engagement with potential candidates.

- Overriding resume's automated ranking would help special cases (e.g. internal referrals or candidates with military experience).

- HR personnel need the ability to tag or add notes to individual resumes allowing further identification (e.g. top-talent, internal referral….).

- User that require access to the application must have the ability to establish usernames/password and be leveraged for authentication purposes.

- Desire for rejection letters to be emailed over to candidates once upon rejection action is entered in the application.

ID #	Description
RUT-1	The system shall provide a user interface for HR users to upload resumes in multiple formats including PDF, DOCX, and TXT
RUT-2	The system shall utilize python script to extract skills, educational background, and professional experience from uploaded resumes.
RUT-3	The system shall store resume extracted data in a structured format in IndexDB.
RUT-4	The system shall allow HR users to define job-specific criteria, including required skills, years of experience, and educational qualifications.
RUT-5	The system shall rank resumes based on defined job-specific criteria
RUT-6	The system must support user account creation capturing username and password.
RUT-7	The system shall allow the user to authenticate to the account based on account information
RUT-8	The system must provide a dashboard that displays analytics of the resume screening process.
RUT-9	The system must provide a dashboard that displays summaries of the resume screening process.
RUT-10	The system shall automatically categorize resumes into predefined job categories based on the extracted skills and experiences using natural language processing.
RUT-11	The system shall generate a match percentage for each resume, indicating how closely the resume meets the job-specific criteria set by the HR user.
RUT-12	The system shall provide a mechanism for HR users to manually override automated rankings.
RUT-13	The system shall provide a mechanism for HR users to add notes or tags to individual resumes.
RUT-14	The system shall allow HR users to schedule and send automated email communications to candidates.
RUT-15	The system shall support real-time notifications to HR users when a resume matches high-priority job criteria.

2. User Stories

The product shall backlog contains a list of all requirements, functional and architectural, required to build our MVP. It is important to note that our team's Product Owner oversees the maintenance and development of the backlog.

The product backlog consists of both functional and architectural requirements mapped to the following 15 User Stories.

Title: US-1	**Priority:** High	**Estimation:** 3 days
User Story: As an HR staff member, I want to be able to upload resumes in multiple formats including PDF, DOCX, and TXT through the system's user interface, so that I can efficiently manage and process applications from various sources without having to manually convert them into a single format, ensuring no candidate is overlooked due to technical constraints.		
Acceptance Criteria: **Given** I am logged into the system as an HR staff member, **When** I upload a resume file in applicable format, **Then** the system should successfully save the file and confirm that the upload was successful.		

Title: US-2	**Priority:** High	**Estimation:** 4 days
User Story: As an HR staff member, I want the system to automatically utilize python script to extract skills, educational background, and professional experience from uploaded resumes.		
Acceptance Criteria: **Given** the extracted data is available in IndexDB **When** an HR staff member accesses a candidate's profile **Then** the profile should display the structured data on skills, education, and professional experience as extracted from the resume.		

Title: US-3	**Priority:** High	**Estimation:** 3 days
User Story: As an HR staff member, I want to store extracted resume data to IndexDB so that this information can be stored in a structured format enabling quick access and analysis of candidate qualifications.		
Acceptance Criteria: **Given** qualification data is extracted **When** resume is uploaded **Then** data is stored in IndexDB		

Title: US-4	**Priority:** Medium	**Estimation:** 4 days
User Story: As an HR staff member, I want to define job-specific criteria, including required skills, years of experience, and educational qualifications so that related data can be stored and queried.		
Acceptance Criteria: **Given** position is created **When** defining job specific criteria **Then** data must be stored to support queries		

Title: US-5	**Priority:** Medium	**Estimation:** 4 days
User Story: As an HR staff member, I want to rank resumes based on defined job-specific criteria so that top candidates can be identified.		
Acceptance Criteria: **Given** resume is uploaded **When** job specific criteria are identified **Then** resumes are ranked based on defined ranking criteria		

Title: US-6	**Priority:** High	**Estimation:** 2 days
User Story: As an HR staff member, I want to have a user account so that I can maintain my username and password.		
Acceptance Criteria: **Given** I am an HR staff member **When** I need a user account **Then** I can maintain my username and password.		

Title: US-7	**Priority:** High	**Estimation:** 2 days
User Story: As an HR staff member, I want to use my account username and password so that I login into the system.		
Acceptance Criteria: **Given** I have a username and password **When** I want to access the system **Then** the username and password can be used to login and authenticate into the system.		

Title: US-8	**Priority:** Low	**Estimation:** 4 days
User Story: As an HR staff member, I want a dashboard that displays analytics so that captured data can be analyzed thoroughly.		
Acceptance Criteria: **Given** resumes are being uploaded and data is stored in IndexDB **When** there is a need to analyze structured data **Then** a dashboard is available that provides analytics of the resume screening process		

Title: US-9	**Priority:** Low	**Estimation:** 4 days
User Story: As an HR staff member, I want a dashboard that displays summaries of the resume screening process so that an overview of the overall process, rather than detailed data, is provided.		
Acceptance Criteria: **Given** resumes are being uploaded and data is stored in IndexDB **When** there is a need to view a summary of screening process **Then** a dashboard is available that leverages uploaded resumes and stored data to provide a process overview		

Title: US-10	**Priority:** Medium	**Estimation:** 4 days
User Story: As an HR staff member, I want to categorize resumes into predefined job categories based on the extracted skills and experiences using natural language processing so that categorizing resumes can be simplified.		
Acceptance Criteria: **Given** job categories are identified, resumes are being uploaded, and data is stored in IndexDB **When** there is a need to categorize resumes based on predefined job categories **Then** natural language processing facilitates categorization based on the extracted skills and experiences		

Title: US-11	**Priority:** Medium	**Estimation:** 3 days
User Story: As an HR staff member, I want to generate a match percentage for each resume so that it is simple to see how closely the resume meets the predefined job-specific criteria.		
Acceptance Criteria: **Given** job categories are identified, resumes are being uploaded, and data is stored in IndexDB **When** best candidate must be identified from various resumes **Then** system generated match percentage will facilitate identification of candidates accordingly		

Title: US-12	**Priority:** Medium	**Estimation:** 3 days
User Story: As an HR staff member, I want to override system generated resume match percentage so that exceptional circumstances (e.g., candidate has skills not previously identified) can be accommodated.		
Acceptance Criteria: **Given** system generated a resume match percentage **When** there is a need to update or edit generated percentage **Then** the ability to manually update the percentage shall be available to rank resume accordingly		

Title: US-13	**Priority:** Medium	**Estimation:** 3 days
User Story: As an HR staff member, I want to add notes or tags to individual resumes so that I can include additional notes (e.g., referred by employee, top candidate...)		
Acceptance Criteria: **Given** resumes are being uploaded, and data is stored in IndexDB **When** there is a need to add comments or updates **Then** the system allows including notes or tags to individual resume to support additional information		

Title: US-14	**Priority:** Medium	**Estimation:** 2 days
User Story: As an HR staff member, I want to schedule and send automated email communications to candidates so that I can ensure notifications are sent and allow time to concentrate on other efforts (e.g., rejection letter).		
Acceptance Criteria: **Given** applicant applies to position **When** an action takes place that requires communication with candidate (e.g., candidate is not selected) **Then** an email should be automatically sent to the candidate notifying them of said action		

Title: US-15	**Priority:** Medium	**Estimation:** 2 days
User Story: As an HR staff member, I want to receive real-time notifications when a resume matches high-priority job criteria so that application can be processed immediately.		
Acceptance Criteria: **Given** high-priority criteria is identified and resume is uploaded **When** uploaded resume matches high-priority criteria **Then** a notification is provided alerting user that uploaded resume contains identified high priority for expedited processing		

Additional Estimated Effort

Estimating effort in terms of hours or days may more often lead to failures in project planning. Moreover, effort estimation considers factors such as technical complexity, potential risks, dependencies, and the need for research or learning new skills. Below are two additional methods that will allow us to measure the complexity but also consider the effort required to complete each user story.

User Story Points

The product backlog outlines 15 user stories, each prioritized and estimated using story points based on the Fibonacci sequence (1, 2, 3, 5, 8, 13, 21) to measure complexity and effort, facilitating efficient planning and execution by the team.

US-1: 5 points	US-6: 3 points	US-11: 5 points
US-2: 8 points	US-7: 3 points	US-12: 5 points
US-3: 5 points	US-8: 8 points	US-13: 5 points
US-4: 8 points	US-9: 8 points	US-14: 3 points
US-5: 8 points	US-10: 8 points	US-15: 3 points

T-shirt Sizes

By categorizing tasks into sizes (e.g., XS, S, M, L, XL), similar to T-shirt sizes, teams can quickly assess and communicate the relative scale of work, making it easier to plan, prioritize, and allocate resources effectively.

T-Shirt Size	User Story	Estimation
Small (S)	US-6	S
	US-7	S
	US-14	S
	US-15	S
Medium (M)	US-1	M
	US-3	M
	US-11	M
	US-12	M
	US-13	M
Large (L)	US-2	L
	US-4	L
	US-5	L
	US-8	L
	US-9	L
	US-10	L

3. Agile Board

The Product and Sprint Backlogs are both maintained within our Atlassian Jira project. To access the board, please first create an account. After you have registered, you can access it at this link.

Due to the limitations of the free version of Jira, which only allows for the creation of tasks and epics but not user stories and sprints, we will use this document to guide our project. This approach will ensure we maintain organization and clear direction throughout our project's development.

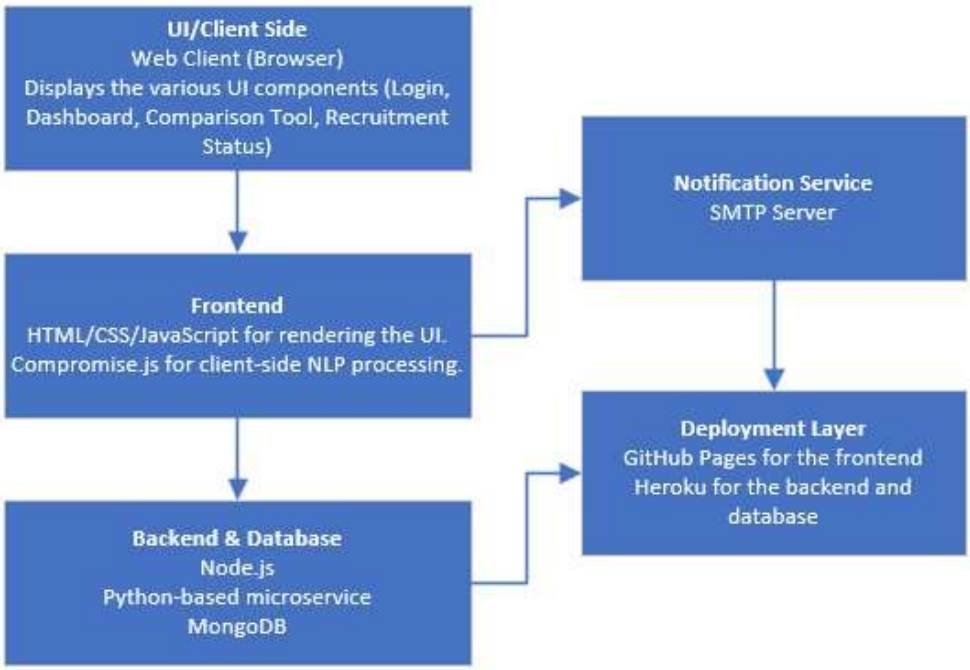
4. Architecture Representation

The solution provides an architecture representation and includes components description, explanation of design decisions, and allocation of responsibilities. The architecture representation is composed of nine components.

- **User Interface**
    - Web Client (user's interface for interacting with the system)
- **Front End Layer**
    - HTML/CSS/JavaScript (assets served to the browser to render the user interface)

    - Compromise.js (JavaScript library used for client-side NLP)
- **Backend Layer**
    - Node.js (routes for authentication, uploading resumes, processing data)
    - Python (NLP processing)
- **Database Layer**
    - MongoDB (store user information, resumes, job descriptions, and analysis results)
- **Notification Service**
    - SMTP Server (Notification and Email)
- **Deployment Layer**
    - GitHub Pages (frontend)
    - Heroku (back-end and database)

Each component was selected not only for its individual capabilities but also for how well it integrates with other services, ensuring a cohesive and efficient system architecture.



III - Non-Functional Requirements

The following are three software quality attributes considered in the architecture. Clearly displaying the rationale on how the architecture incorporates the requirements, and how responsibilities are assigned to architectural components.

The resume upload tool is designed to streamline the process of analyzing and managing resumes by HR professionals. This tool will support the extraction, categorization, and analysis of text from uploaded resumes using advanced algorithms. The primary users are HR staff members who require efficient and accurate processing of candidate information.

ID #	Description
NFR-1	The system must be designed to scale horizontally to accommodate growth in user numbers and data volume without degradation of performance.
NFR-2	The system shall provide a user-friendly interface, requiring no more than 30 minutes of training for new HR users to become proficient.
NFR-3	The system shall process and categorize each uploaded resume in under 10 seconds.
NFR-4	The system should be available with a minimum uptime of 90% during peak usage hours.

IV - Algorithmic Component: Resume Ranking and Matching

The proposed algorithmic component is a resume ranking and matching algorithm that leverages Natural Language Processing (NLP). This component will analyze the content of resumes and rank them according to their relevance to specific job criteria provided by HR professionals.

The resume ranking algorithm will consist of several stages:

1. Text Preprocessing: Clean and normalize the extracted text from resumes.
2. Feature Extraction: Extract relevant features from the text, such as keywords, phrases, and semantic similarities.
3. Similarity Scoring: Compute similarity scores between resumes and job criteria using vector space models and similarity metrics (e.g., cosine similarity).
4. Ranking and Sorting: Generate a ranked list of resumes for each job posting based on the computed scores.

This algorithmic component adds significant value to the resume upload tool by automating the complex task of resume evaluation. It ensures that HR professionals can quickly identify the most relevant candidates, improving the efficiency and accuracy of the hiring process. By leveraging advanced NLP and

ML techniques, the system can adapt to various job criteria and continually improve its ranking accuracy over time.

Rationale for the Solution

The resume ranking and matching algorithm is designed to address the critical need for efficient and accurate resume evaluation. It integrates advanced logic and programming techniques to automate and enhance the hiring process.

1. Preprocessing: Tokenize, lemmatize, and clean the text to prepare it for analysis.
2. Feature Extraction: Use techniques like TF-IDF, word embeddings, and named entity recognition to extract meaningful features from the text.
3. Similarity Scoring: Implement vector space models and similarity metrics to compute how closely a resume matches the job criteria.
4. Ranking: Sort resumes based on their similarity scores.

Flow of the Resume Ranking Algorithm:

1. Input Layer:
    a. Upload Resume
    b. Extract Text using Optical Character Recognition
2. Preprocessing Layer:
    a. Text Cleaning
    b. Tokenization
    c. Lemmatization
3. Feature Extraction Layer:
    a. TF-IDF Vectorization
    b. Word Embeddings
    c. Named Entity Recognition
4. Similarity Scoring Layer:
    a. Compute Cosine Similarity
    b. Calculate Relevance Scores
5. Output Layer:
    a. Generate Ranked List of Resumes
    b. Display Results to HR Professionals

By implementing this algorithmic component, the resume upload tool will significantly enhance its functionality, providing a robust solution for automated resume analysis and ranking. This will not only improve the hiring process but also ensure that the system meets the advanced algorithmic requirements of the course.