PUCIT
Punjab University College of Information Technology

Revaluator - The Ai Recruiter

First Deliverable

Version 1.0

TABLE OF CONTENTS

1. Introduction

Revaluator - Our Platform is a three-tier architecture web-based system designed to revolutionize how organizations hire. By automating CV screening, generating tailored tests using a Large Language Model (LLM), and prioritizing data security, AI-Recruiter streamlines the recruitment journey.

It offers a solution that enhances efficiency, reduces bias, and enables recruiters to make more informed decisions, ultimately redefining how businesses identify and select top-tier candidates in an increasingly competitive job market.

1.1 Project Feasibility Report

1.1.1 Technical Feasibility

Revaluator, will be developed using well-established technologies within the framework of a three-tier architecture. This approach offers several advantages, including cost-effectiveness, scalability, and ease of maintenance. The selected technology stack is based on widely accessible open-source tools, ensuring that Revaluator is technically feasible for development.

1.1.2 Operational Feasibility

Operational feasibility plays a pivotal role in our project. Revaluator addresses a real need in the recruitment industry, providing a cutting-edge solution for recruiters. The platform caters to those seeking to streamline the process of turning candidate selection into a seamless and resource-efficient operation. This critical problem necessitates a solution, and Revaluator is poised to provide a means to raise the bar in candidate evaluation.

1.1.3 Economic Feasibility

Economically, Revaluator is a feasible project. Hosting expenses are minimal, and future revenue can be generated by deducting a percentage from funds contributed to each recruitment campaign. This approach ensures that development, acquisition, maintenance, and operational costs are kept to a minimum. No significant initial investment is required, and cost estimates do not deter our project's viability.

1.1.4 Schedule Feasibility

Regarding schedule feasibility, Revaluator is expected to be completed within an estimated timeframe of eight to nine months. The project will be structured into distinct modules, each with its own schedule. Accountability will be maintained through assigned tasks for each team member. Meeting deadlines and achieving milestones for project proposals, deliverables, prototypes, and other development stages will be closely monitored to ensure the project's timely completion.

1.1.5 Specification Feasibility

Our project will adhere to UML (Unified Modeling Language) methods to identify and model various business processes. This will involve defining requirements, use cases, actors, and more to fully grasp user needs. Revaluator is designed to run on personal computers with modest specifications and compatibility with standard web browsers such as Chrome, Firefox, Opera, and others.

1.1.6 Information Feasibility

All information provided by Revaluator, including data about candidates, recruiters, job positions, and application criteria, is dependable and meaningful. The project is built to ensure information accuracy and reliability.

1.1.7 Motivational Feasibility

Evaluating user motivation to engage with Revaluator, whether it involves creating job listings or applying for positions, will be a core part of our project's ongoing development. User feedback and response will guide us in enhancing the platform's motivational aspects.

1.1.8 Legal & Ethical Feasibility

Revaluator is designed with legal and ethical considerations in mind. We will adhere to all applicable laws and regulations in the development and operation of the platform. Proper privacy policies and rules and regulations will be in place to ensure ethical usage of Revaluator, providing a secure and responsible environment for all users.

1.2 Project / Product Scope

"Revaluator - The AI Recruiter," aims to address these challenges by introducing an advanced CV filtering mechanism powered by a Large Language Model (LLM). This technology-driven approach provides a more efficient and objective means of candidate screening. By analyzing the content of CVs in a holistic and context-aware manner, our system identifies and ranks candidates based on their qualifications, skills, and potential, reducing the burden of manual screening and increasing the likelihood of identifying the best candidates for a job opening.

Our core mission is to equip recruiters with an effective tool that redefines the way candidates are selected and assessed, striving for greater efficiency and precision in the hiring process.

1.3 Project/Product Costing

Cost estimation is done by Function Point Analysis(FPA).

1.3.1 Project Cost Estimation By Function Point Analysis

External Inputs:

1. Recruiter Registration Screen
2. Recruiter Login Screen
3. Resume/CV UploadRecruiter Screen
4. Skill Entry Screen
5. Test Editor Screen
6. Emails Adding Screen
7. Set Time and Complexity Level Screen
8. Test Platform Screens

External Outputs:

1. Skills Extraction Screen
2. Test Instruction Screen
3. Test Viewing Screen
4. Test Result Screen
5. CV Rating Screen
6. Candidates List Screen
7. Confirmation and Notification Messages

User Inquiries:

1. User Validation(Authentication)
2. Payment Verification (Banking App)

Internal Files:

1. Recruiter Database
2. Candidate Database
3. Resume/CV Database
4. Test Database
5. Evaluation Database

External Interfaces:

1. Large Language Model (LLM) - for generating tests
2. API Calls for CV Rating
3. Database Communication

4. Email Services - to send test links
5. Third-Party APIs - for specific features or functionality.

Information domain values:

1. External Inputs: 8
2. External Outputs: 7
3. User Inquiries: 2
4. Internal Files: 5
5. External Interfaces: 5

Information Domain Value	Count	Simple	Average	Complex	Individual Attributes
External Inputs	8	4x5	4x2	6x1	34
External Outputs	7	4x4	5x2	7x1	33
Internal Files	5	8x3	10x1	15x1	49
External Interfaces	5	5x3	7x1	10x1	32
User Inquiries	2	4x1	5x1	6x0	9
Total Count					157

Sr#	Complexity Adjustment Factor	Scale					
		1	2	3	4	5	Value
1	CV Data Processing				√		4
2	LLM Integration					√	5
3	Real-time Data Handling				√		4
4	User Interaction Complexity			√			3
5	Recruiter Authentication			√			3
6	Custom Test Generation				√		4
7	Scalability and Performance					√	5
8	Data Security Measures				√		4
9	Automated Evaluation Process				√		4
10	Skill Extraction Complexity			√			3
11	Database Schema Complexity		√				2
12	Third-party API Integration			√			3
13	Cross-Browser Compatibility		√				2
14	AI Model Training Complexity				√		4
Value Adjustment Factor ∑ Fi							**50**

Functional Point Calculation:

FP est. = 157 * [0.65 + 0.01 * (50)]
= 180.55 **pm**

Project Labour Rate and Productivity:

1 month = 22 working days,
Labour Rate per Person = 80,000 Rs/pm (per day: 3636.36 Rs)
Average productivity per Person = 36 FP/month (per day: 1.63 FP)

Cost per FP =Labor Rate/Productivity

Cost/FP = 3636.36/36
= 101.0101 Rs/FP

Total Project Cost = FP estimation *(Cost/FP)

Total Project Cost = 180.55 * 101.0101
= 18237.3735 Rs

Total Estimated Effort per Month :

Total estimation effort = FP estimation/Average Productivity
Total estimation effort = 180.55/36
= 5.01 persons-month

1.4 CPM - Critical Path Method

1.4.1 Critical Path Method Activity Sequence and Completion Time:

Information Gathering and Planning

- Project scope
- Requirement Gathering
- Risk List

Analysis Phase

- Feasibility Report
- Cost Estimation
- System Specification

Designing Phase

- Identifying Entities
- Draw Context Level Diagram
- Draw Data Flow Diagram
- Capturing Shall Statement
- Draw Use Case Diagram

Development Phase

- CV Uploading and Skill Extraction Module
- CV Rating System Module
- Language Model (LLM) Testing Module
- Custom Test Creation Module
- Candidate Test Distribution Module
- Back-end Development

Testing Phase

- Testing
- Correction of Errors.

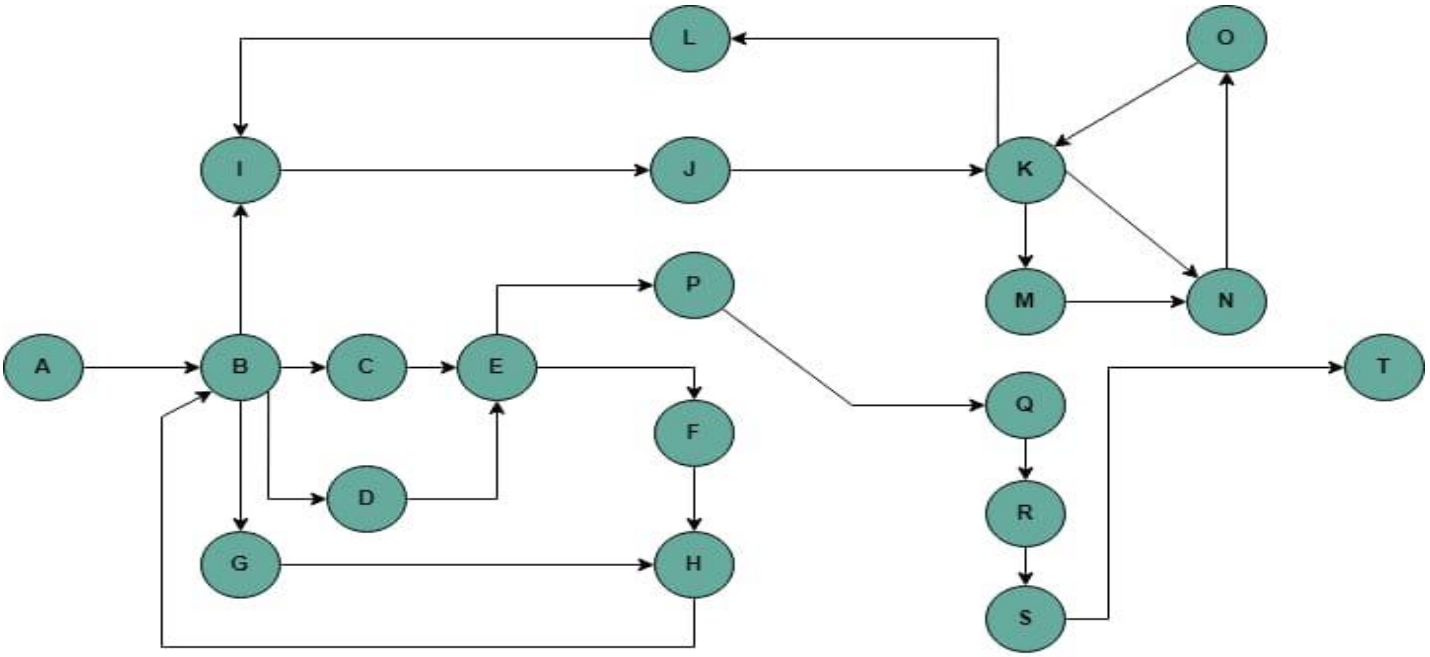
Deployment Phase

- Deployed and Delivered

Activity	Activity Name	Immediate Predecessor	Duration(Days)
A	Project Scope	None	1 day
B	Requirement Gathering	A	5 days
C	Risk List	B	2 days
D	Feasibility Report	A, B	5 days
E	System Specifications	D,C	5 days
F	Identifying Entities	E	3 days
G	Draw Context Level Diagram	B	2 days
H	Draw Data Flow Diagram	G,F	2 days

I	Capturing Shall Statement	B,H	2 days
J	Draw Use Case Diagram	I	8 days
K	CV Uploading and Skill Extraction Module	J	10 days
L	CV Rating System Module	K,I	10 days
M	Language Model (LLM) Testing Module	K	40 days
N	Custom Test Creation Module	K,M	30 days
O	Candidate Test Distribution Module	K,N	20 days
P	Back-end Development	E	90 days
Q	Integration	P	10 days
R	Testing	Q	10 days
S	Deployment	R	4 days
T	Deliver	S	1 day

1.4.2 Network Diagram:



1.4.3 Identify the Critical Path:

s

Activity	Duration	ES	EF	LS	LF	TS	FS
A	1	0	1	0	1	0	0
B	5	1	6	1	6	0	0
C	2	6	8	6	8	0	0
D	5	1	6	31	36	30	0
E	2	8	10	36	38	28	0
F	5	10	15	38	43	28	0
G	3	15	18	43	46	28	0
H	2	18	20	46	48	28	0
I	2	20	22	48	50	28	0
J	8	22	30	50	58	28	0
K	10	30	40	58	68	28	0
L	10	40	50	68	78	28	0
M	40	50	90	78	118	28	0
N	30	90	120	118	148	28	0
L	10	40	50	68	78	28	0
M	40	50	90	78	118	28	0
N	30	90	120	118	148	28	0
O	20	120	140	148	168	28	0
P	90	140	230	168	258	28	0
Q	10	230	240	258	268	28	0

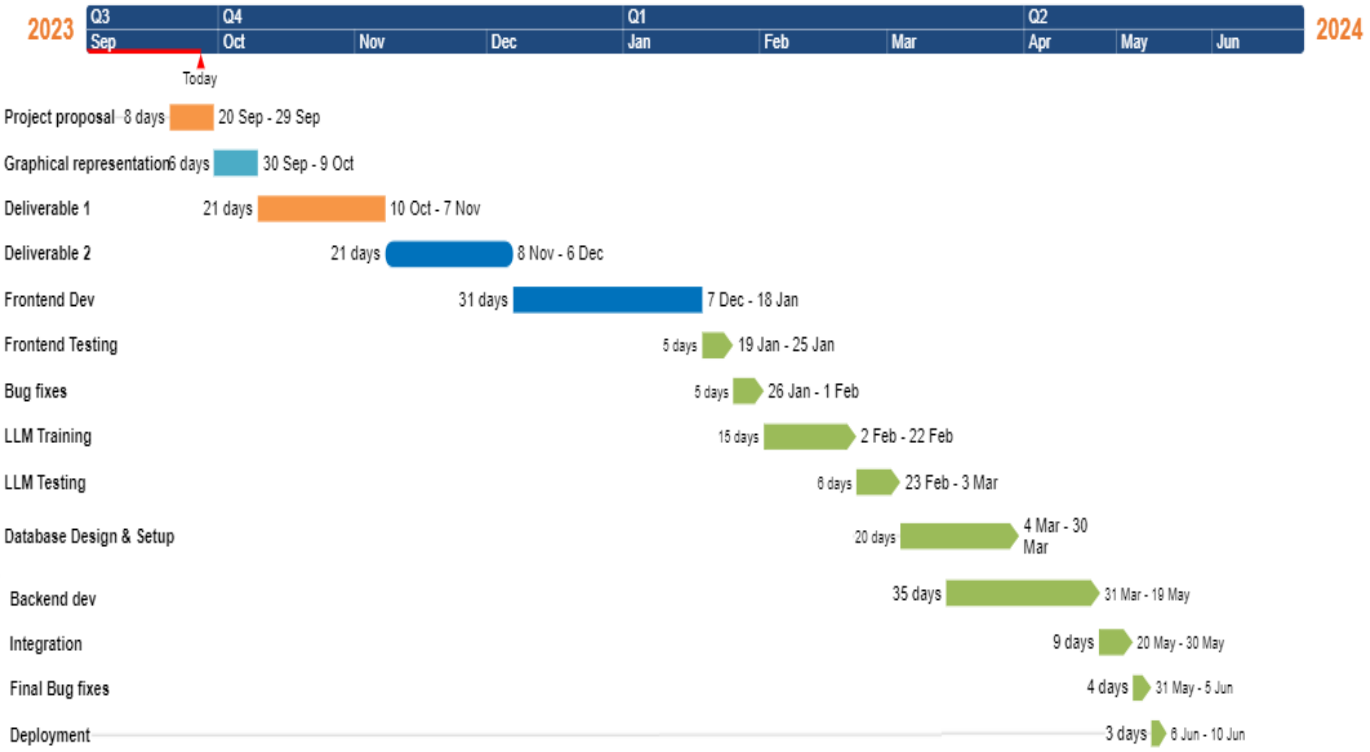
R	10	240	250	268	278	28	0
S	5	250	255	278	283	28	0
T	1	255	256	283	284	28	0

Parameters and slacks are calculated as follows:

The critical path is:

A → B → C → E → P→ Q →R →S →T

1.5 Gantt chart



1.6 Introduction to Team members and their skill set.

➢ Hashir (Team Lead)

Hashir's skill set is a formidable asset to our project, encompassing:

- Profound expertise in the MERN (MongoDB, Express.js, ReactJs, NodeJs) stack
- Python, C++, C#, MVC Framework along with strong Problem Solving Skills
- Strong Capabilities in ML, AI , Database Design and Project Management

Hashir is the steadfast leader who anchors our project's journey to success. His passion for AI and Machine Learning is the driving force that propels our team forward.Hashir thrives in fast-paced environments, excels in team collaboration, and is known for his organization and strong interpersonal skills. With a compelling communication style, he orchestrates our project's success with grace and influence.

➢ Shazil (Team Member)

Shazil's technical skills related to this project include

- React Js, Node.Js, MongoDB , PostgreSql .
- Python, Problem Solving, Database Modeling.
- Analytical Skills

Apart from these skills, he is good at problem-solving and communication with the team. He will be working on Backend Development, as well as Front End Development , Documentation of the requirement phase, and Database Design. He will be managing project resources and timelines

➢ Asim Ali (Team Member)

Asim skill set includes the following skills related to this project:

- C++, C#, Python, Mvc Rest Api Framework
- Database Design, Machine Learning, Project Management

Ambitious and enthusiastic seventh semester undergraduate student having passion about programming who can work in a fast-paced environment and can collaborate effectively within teams. He is organized and has strong interpersonal skills. He has convincing powers.

- **Muhammad Qadeer (Team Member)**

Qadeer skill set includes the following skills related to this project:

- C++, C#, Python, LLM fine tuning, Django along with good Problem solving skills.
- Strong Capabilities in Database Design, Machine Learning, Project Management and Artificial Intelligence.

Introducing Qadeer, a dynamic team member with a powerful skill set tailored to our project's needs. With expertise in C++, C#, Python, and LLM fine tuning, His programming prowess is complemented by strong problem-solving skills. Furthermore, his capabilities span Database Design, Machine Learning, Project Management, and Artificial Intelligence, making him a valuable asset for our team's success.

- **Huzaifa Khawar (Team Member)**

Huzaifa skill set includes the following skills related to this project:

- C++, C#, Mvc Rest API Framework
- Database Design, Project Management
- Front end development (React.js)

Meet Huzaifa Khawar, an ambitious and passionate seventh-semester undergraduate with strong technical skills and teamwork capabilities. He's an ideal fit for the Employee Attendance System project, well-prepared to ensure its success and deeply interested to learn new skills and technologies.

1.7 Tools and Technologies with reasoning

Tools	Reason for Using
VS Code	Used as IDE for backend and frontend development, it is lightweight, extensible, free, open source, and cross-platform.
Jupyter Notebook	Jupyter Notebook provides an interactive environment for data analysis and experimentation, aiding in the development of our machine learning models.

PyCharm	Specialized Python and Django IDE with code analysis, debugging, and testing support.
GIT and GitHub	Use for the collaboration of the team. It is an open-source and cloud-based hosting service that lets you manage git repositories and changes.
MS Word	Use for documentation, It is accessible and user-friendly, highlights spelling or grammar mistakes, and helps in fixing them.
Online Gantt	Used for Gantt charts, It is a user-friendly online tool that helps you organize projects, improve overall project visibility, and keeps everyone on track.
Figma/Canva	Use for Prototypes, It is best for collaboration, you can edit designs as a team, can see who changed what, and can share it with everyone for free.
PgAdmin and DBeaver	These database management tools, PgAdmin for PostgreSQL and DBeaver for various databases, assist in database administration and schema design.
LucidChart	LucidChart is the best tool for diagramming. It consists of a number of shapes and diagram options.
ClickUp	Used for project management, offering a user-friendly online tool to organize projects, improve visibility, and facilitate team collaboration.

Technologies:

Technologies	Reasons for Using
Next.js	We employ Next.js to build a high-performance, server-rendered React application, delivering a responsive and engaging front-end for our CV filtering system.
Language Model (LLM)	LLM plays a central role in our project by generating customized tests based on candidate skills and qualifications extracted from CVs. It enables personalized assessments and enhances the evaluation process.
Django	Python serves as the primary programming language for our project,facilitating machine learning model development, data processing, and scripting tasks.
MongoDB/PostgreSQL	Depending on the project requirements, we use MongoDB for flexible, NoSQL data storage or PostgreSQL for structured, relational data storage, ensuring data integrity and retrieval efficiency
Prisma ORM	Prisma ORM streamlines database access and management, allowing us to work with databases using a type-safe and intuitive API, improving the overall robustness of our system.

Languages:

Languages	Reasons for Using
JavaScript	React, Node, and Express are frameworks of JS.
Python	Django , LLM , PyPdf2 , Transformers .

1.8 Vision Document for Project Revaluator

1.8.1 Introduction

Revaluator is a groundbreaking project aimed at revolutionizing the recruitment process. It presents a three-tier architecture web-based platform designed to streamline candidate selection and evaluation. At its heart, Revaluator employs a CV filtering system, powered by a cutting-edge Large Language Model (LLM). Our core mission is to equip recruiters with an effective tool that transforms how candidates are chosen and assessed.

1.8.2 Existing Recruitment Processes

In the modern recruitment landscape, technological advancements have given rise to platforms like LinkedIn, Indeed, and Monster, as well as Applicant Tracking Systems (ATS) and online job portals. However, these platforms frequently struggle to comprehensively evaluate candidates' qualifications and potential, highlighting the need for more efficient solutions in candidate assessment.

1.8.2.1 Inefficient Candidate Screening

Traditional recruitment processes are often plagued by inefficient candidate screening. Managing a large influx of job applications manually is time-consuming and can lead to the oversight of potentially excellent candidates. Our technology-driven solution seeks to streamline and enhance the assessment of candidate qualifications, alleviating the manual screening workload and boosting recruitment precision.

1.8.2.2 The Power of the Large Language Model (LLM)

Revaluator leverages the capabilities of a Large Language Model to understand, interpret, and evaluate candidate CVs. This technology will provide a more accurate and efficient means of identifying the most promising candidates for a given job role.

Key Objectives of Revaluator

Our primary objectives for Revaluator include:

- ➔ Enhancing the efficiency of the candidate selection process, reducing manual effort, and improving the accuracy of matching candidates to job roles.
- ➔ Creating a dynamic platform that simplifies the job of recruiters by utilizing advanced AI and NLP techniques to assess candidate qualifications, skills, and experience.

- ➔ Providing a comprehensive solution that streamlines the entire recruitment process, from initial test to making final selections.
- ➔ Reducing the burden of administrative tasks for recruiters and HR professionals, enabling them to focus on the most critical aspects of candidate evaluation.
- ➔ Offering a cost-effective and user-friendly solution that can be integrated seamlessly into existing recruitment processes.

Conclusion

Revaluator aims to set a new standard in the recruitment industry by leveraging cutting-edge technology to enhance the way candidates are evaluated and selected. Our innovative approach will empower recruiters with the tools they need to make more informed and efficient hiring decisions.

1.8.3 Project Overview

"Revaluator - The AI Recruiter" is designed to simplify and enhance the recruitment process. Using advanced technology, it automatically evaluates CVs by extracting skills and ranking candidates based on qualifications. This eliminates manual screening, making it easier for recruiters to find the best candidates efficiently and with precision. Our mission is to provide a powerful tool that transforms candidate selection, improving the overall hiring process.

1.8.4) Identifying External Entities or Actors:

The Identification of External Entities or Actors is done in two phases

1.8.5) Over Specify Entities from Abstract:

Based on the project's abstract and objectives, we can identify the following entities and actors:

- ➢ User
- ➢ Recruiters
- ➢ Candidates
- ➢ CVs (Candidate Resumes)
- ➢ Skills
- ➢ Test Questions
- ➢ Test Results
- ➢ Notifications
- ➢ User Interface (UI)
- ➢ Language Model (LLM)
- ➢ Database
- ➢ Test Editor
- ➢ Communication System
- ➢ Reporting and Analytics
- ➢ Payment (if applicable)

1.8.6) Perform Refinement:

Upon further refinement and alignment with the project's core objectives, we can narrow down the key entities and actors that are central to our system:

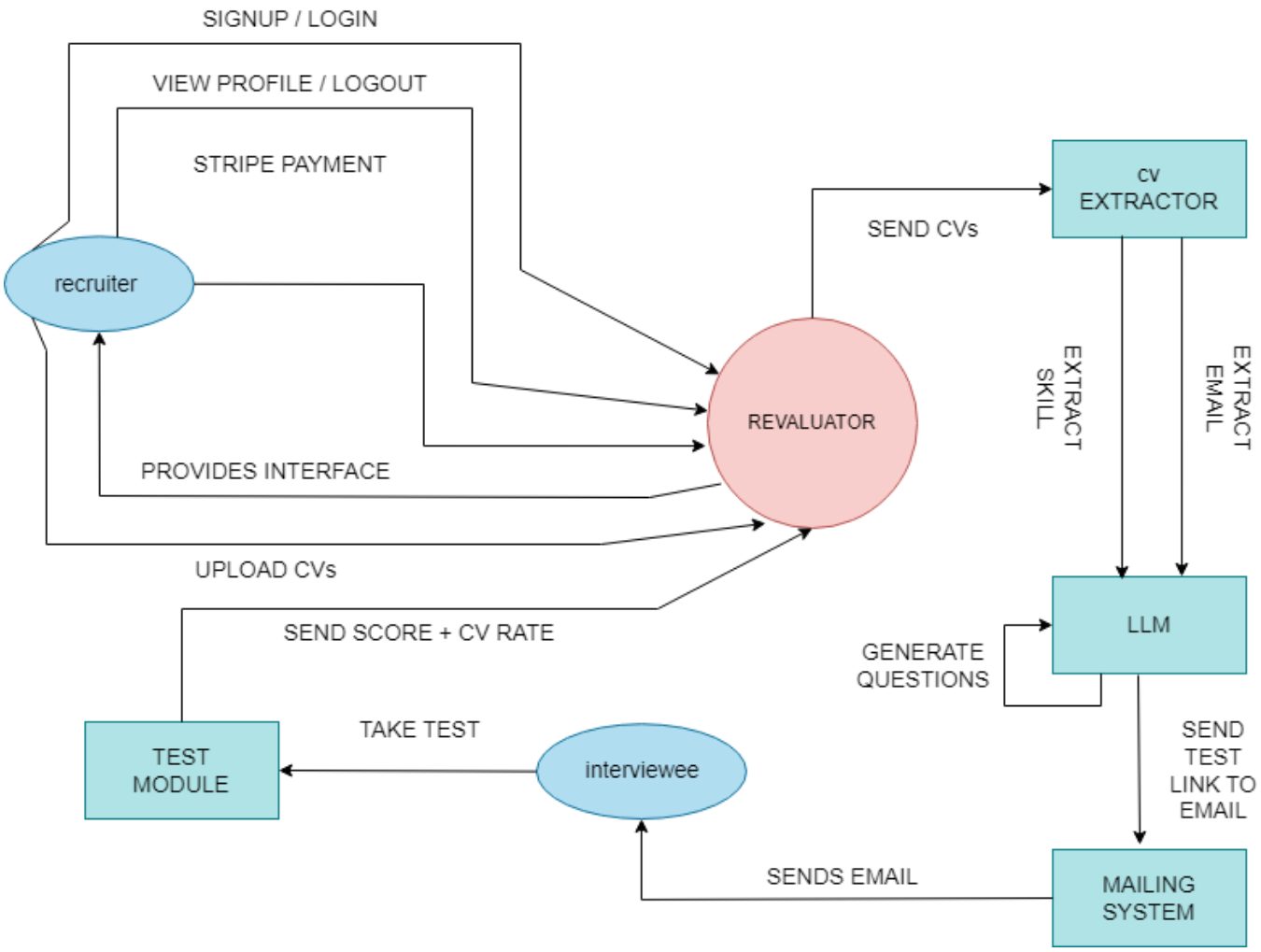
- User
  - Recruiters
  - Candidates
- CVs (Candidate Resumes)
- Skills
- Skill Extraction System
- Custom Tests
  - Test Questions
  - Test Results
- Notifications
  - Automated Notifications
  - Communication System
- Language Model (LLM)
  - LLM Training
- Database
  - Data Storage
- User Interface (UI)
  - User Interaction

1.8.4 Summary of Requirements

Sr#	Requirement Entity	Category
1	User Registration	Functional
2	User Login	Functional
3	CV Upload	Functional
4	CV Rating	Functional
5	Skills Extraction	Functional
6	Email Extraction	Functional

7	Custom Skill Configuration	Functional
8	LLM Integration	Non-Functional
9	Test Generation	Functional
10	Test Link Generation	Functional
11	Test Taking	Functional
12	Test Security	Non-Functional
13	Test Evaluation	Functional
14	Results Presentation	Functional
15	CV Rating Display	Functional
16	Data Security	Non-Functional
17	Data Integrity	Non-Functional
18	Performance	Non-Functional
19	Scalability	Non-Functional
20	Usability	Non-Functional
21	Compatibility	Non-Functional
22	Payment Processing	Functional
23	Reporting and Analytics	Functional
24	Mobile Responsiveness	Non-Functional
25	User Account Management	Functional
26	Data Backup and Recovery	Non-Functional
27	User Data Privacy Protection	Non-Functional
28	User Notifications	Functional
29	Test Result Recording	Functional
30	Evaluation Criteria	Functional

1.8.7 Context Level Data Flow Diagram



1.8.8 Capture "shall" Statements and the external entities (Actors)

Sr#	Entities	Shall Statements
01	User	The system shall allow the User to register using Google Accounts.
02	User	The system shall enable the User (Recruiter) to sign in to the application.

03	User	The system shall enable the user to purchase the software using Stripe.
04	System	The system shall enable the management of profile information.
05	System	The system shall provide an intuitive and user-friendly user interface, ensuring the User can easily navigate the platform.
06	System	The system shall offer a straightforward and user-friendly CV upload interface.
07	User	The system shall allow the User to easily upload candidate CVs.
08	System	The system shall support various PDF file formats for CV uploads.
09	System	The system shall automatically extract skills and qualifications from the CV.
10	System	The system shall extract the user's email from the CV.
11	User	The system shall provide and allow the User with an option to generate custom tests for candidates based on their skills or provide a role.
12	System	The system shall generate clear and understandable tests for candidates.
13	User	The system shall allow the User to customize test questions and format.
14	System	The system shall provide an easy-to-use and intuitive test editor.
15	System	The system shall send secure test links to candidates' email addresses.
16	System	The system shall take the test on the candidate's extracted skills and qualifications or provided by the User.
17	System	The system shall provide secure test links that prevent candidates from switching tabs or accessing external resources during the test.
18	System	The system shall automatically evaluate the tests and provide objective results.
19	System	The system shall ensure the integrity and accuracy of the evaluation process.
20	System	The system shall present results in an easily understandable format for the User.

21	System	The system shall provide the User with clear and understandable ratings for candidate CVs.
22	System	The system shall securely store candidate data.
23	System	The system shall provide 24/7 system availability, ensuring continuous system accessibility for Recruiters.

1.8.9 : Allocate Requirements

Sr#	Entities	Shall Statements	Use Case Name
01	User	The system shall allow the User to register using Google Accounts.	UC_Register_Google_Account
02	User	The system shall enable the User (Recruiter) to sign in to the application.	UC_Login_Recruiter
03	User	The system shall enable the user to purchase the software using Stripe.	UC_Purchase_with_Stripe
04	System	The system shall enable the management of profile information.	UC_Manage_Profile
05	System	The system shall provide an intuitive and user-friendly user interface, ensuring the User can easily navigate the platform.	UC_User_Interface
06	System	The system shall offer a straightforward and user-friendly CV upload interface.	UC_CV_Upload_Interface
07	User	The system shall allow the User to easily upload candidate CVs.	UC_Upload_Candidate_CVs
08	System	The system shall support various PDF file formats for CV uploads.	UC_Support_PDF_Formats
09	System	The system shall automatically extract skills and qualifications from the CV.	UC_Extract_Skills
10	System	The system shall extract the user's email from the CV.	UC_Extract_Email
11	User	The system shall provide and allow the User with an option to generate custom tests for candidates based on their skills or provide a role.	UC_Generate_Custom_Tests
12	System	The system shall generate clear and understandable tests for candidates.	UC_Generate_Test
13	User	The system shall allow the User to customize test questions and format.	UC_Customize_Test

14	System	The system shall provide an easy-to-use and intuitive test editor.	UC_Test_Editor
15	System	The system shall send secure test links to candidates' email addresses.	UC_Send_Secure_Test_Links
16	System	The system shall take the test on the candidate's extracted skills and qualifications or provided by the User.	UC_Take_Test
17	System	The system shall provide secure test links that prevent candidates from switching tabs or accessing external resources during the test.	UC_Secure_Test_Links
18	System	The system shall automatically evaluate the tests and provide objective results.	UC_Auto_Evaluate_Tests
19	System	The system shall ensure the integrity and accuracy of the evaluation process.	UC_Evaluation_Integrity
20	System	The system shall present results in an easily understandable format for the User.	UC_Present_Results
21	System	The system shall provide the User with clear and understandable ratings for candidate CVs.	UC_CV_Ratings
22	System	The system shall securely store candidate data.	UC_Secure_Candidate_Data
23	System	The system shall provide 24/7 system availability, ensuring continuous system accessibility for Recruiters.	UC_System_Availability

1.8.10: Priorities Requirements

Sr#	Rank	Shall Statements	Use Case ID	Use Case Name
01	High	The system shall allow the User to register using Google Accounts.	UC-01	UC_Register_Google_Account
02	High	The system shall enable the User (Recruiter) to sign in to the application.	UC-02	UC_Login_Recruiter
03	Medium	The system shall enable the user to purchase the software using Stripe.	UC-03	UC_Purchase_with_Stripe
04	Low	The system shall enable the management of profile information.	UC-04	UC_Manage_Profile
05	High	The system shall provide an intuitive and user-friendly user interface, ensuring the User can easily navigate the platform.	UC-05	UC_User_Interface

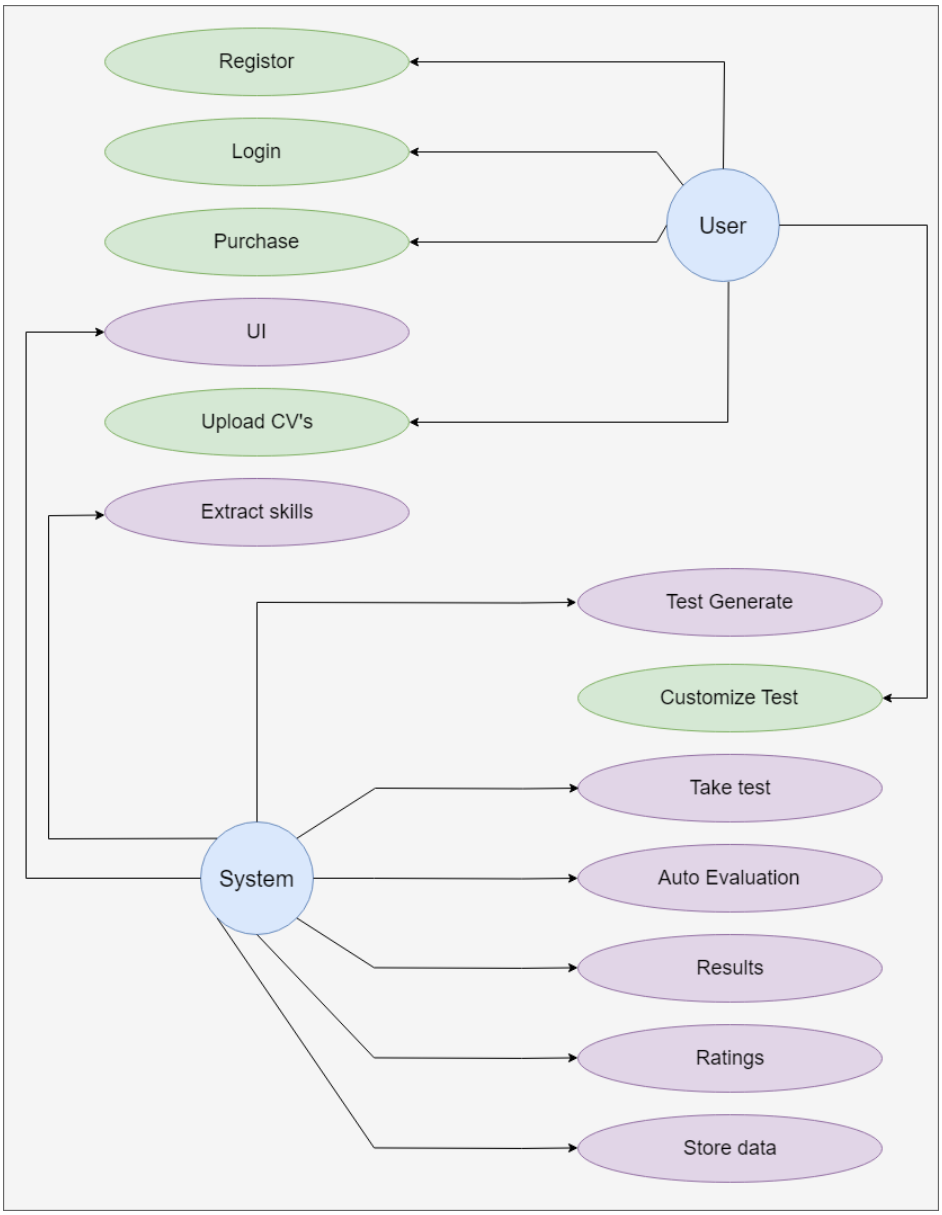
06	High	The system shall offer a straightforward and user-friendly CV upload interface.	UC-06	UC_CV_Upload_Interface
07	High	The system shall allow the User to easily upload candidate CVs.	UC-07	UC_Upload_Candidate_CVs
08	Medium	The system shall support various PDF file formats for CV uploads.	UC-08	UC_Support_PDF_Formats
09	High	The system shall automatically extract skills and qualifications from the CV.	UC-09	UC_Extract_Skills
10	Medium	The system shall extract the user's email from the CV.	UC-10	UC_Extract_Email
11	Medium	The system shall provide and allow the User with an option to generate custom tests for candidates based on their skills or provide a role.	UC-11	UC_Generate_Custom_Tests
12	Medium	The system shall generate clear and understandable tests for candidates.	UC-12	UC_Generate_Test
13	Low	The system shall allow the User to customize test questions and format.	UC-13	UC_Customize_Test
14	Medium	The system shall provide an easy-to-use and intuitive test editor.	UC-14	UC_Test_Editor
15	High	The system shall send secure test links to candidates' email addresses.	UC-15	UC_Send_Secure_Test_Links
16	High	The system shall take the test on the candidate's extracted skills and qualifications or provided by the User.	UC-16	UC_Take_Test
17	High	The system shall provide secure test links that prevent candidates from switching tabs or accessing external resources during the test.	UC-17	UC_Secure_Test_Links
18	High	The system shall automatically evaluate the tests and provide objective results.	UC-18	UC_Auto_Evaluate_Tests
19	Medium	The system shall ensure the integrity and accuracy of the evaluation process.	UC-19	UC_Evaluation_Integrity
20	Medium	The system shall present results in an easily understandable format for the User.	UC-20	UC_Present_Results
21	High	The system shall provide the User with clear and understandable ratings for candidate CVs.	UC-21	UC_CV_Ratings

22	Medium	The system shall securely store candidate data.	UC-22	UC_Secure_Candidate_ Data
23	High	The system shall provide 24/7 system availability, ensuring continuous system accessibility for Recruiters.	UC-23	UC_System_Availability

1.8.11 : Requirements Traceability Matrix

Sr#	Build	Initial Requirements	Category	Use Case Name
1	B1	The system shall allow the User to register using Google Accounts.	Functional	UC_Register_Google_ Account
2	B1	The system shall enable the User (Recruiter) to sign in to the application.	Functional	UC_Login_Recruiter
3	B2	The system shall enable the user to purchase the software using Stripe.	Functional	UC_Purchase_with_Str ipe
4	B3	The system shall enable the management of profile information.	Functional	UC_Manage_Profile
5	B3	The system shall provide an intuitive and user-friendly user interface, ensuring the User can easily navigate the platform.	Functional	UC_User_Interface
6	B3	The system shall offer a straightforward and user-friendly CV upload interface.	Functional	UC_CV_Upload_Interf ace
7	B3	The system shall allow the User to easily upload candidate CVs.	Functional	UC_Upload_Candidate _CVs
8	B3	The system shall support various PDF file formats for CV uploads.	Functional	UC_Support_PDF_For mats
9	B4	The system shall automatically extract skills and qualifications from the CV.	Functional	UC_Extract_Skills
10	B4	The system shall extract the user's email from the CV.	Functional	UC_Extract_Email
11	B5	The system shall provide and allow the User with an option to generate custom tests for candidates based on their skills or provide a role.	Functional	UC_Generate_Custom _Tests
12	B5	The system shall generate clear and understandable tests for candidates.	Functional	UC_Generate_Test

13	B5	The system shall allow the User to customize test questions and format.	Functional	UC_Customize_Test
14	B5	The system shall provide an easy-to-use and intuitive test editor.	Functional	UC_Test_Editor
15	B6	The system shall send secure test links to candidates' email addresses.	Functional	UC_Send_Secure_Test_Links
16	B6	The system shall take the test on the candidate's extracted skills and qualifications or provided by the User.	Functional	UC_Take_Test
17	B6	The system shall provide secure test links that prevent candidates from switching tabs or accessing external resources during the test.	Functional	UC_Secure_Test_Links
18	B7	The system shall automatically evaluate the tests and provide objective results.	Functional	UC_Auto_Evaluate_Tests
19	B7	The system shall ensure the integrity and accuracy of the evaluation process.	Functional	UC_Evaluation_Integrity
20	B7	The system shall present results in an easily understandable format for the User.	Functional	UC_Present_Results
21	B7	The system shall provide the User with clear and understandable ratings for candidate CVs.	Functional	UC_CV_Ratings
22	B8	The system shall securely store candidate data.	Functional	UC_Secure_Candidate_Data
23	B8	The system shall provide 24/7 system availability, ensuring continuous system accessibility for Recruiters.	Functional	UC_System_Availability



1.9 Risk List

Below are the risks involved in developing the proposed application.

1.9.1 Risk Related to Requirement Evolution

Given the dynamic nature of our project, the constant evolution of requirements is expected. Changes in requirements have the potential to impact the project's scope, and it's highly likely that requirements may undergo modifications at any stage of development.

1.9.2 Risk Related to Market Impact

Failure to achieve the intended project outcomes can expose the project to market risks. This could create opportunities for competitors to exploit weaknesses and potentially impact the project's market presence.

1.9.3 Risk Related to Time Management

Learning and implementing new technologies can be time-consuming. The process of understanding and applying these technologies may take longer than initially anticipated.

1.9.4 Risk Related to Model Training

The integration of the Machine Learning Model into our project and the training process may be time-intensive, potentially resulting in delays in adhering to our project schedule.

1.9.5 Risk Related to Data Collection

Gathering a substantial dataset tailored to the unique requirements of company hiring question set is a challenge. This challenge may lead to potential delays in completing scheduled tasks, primarily linked to dataset acquisition for training purposes.