BUSINESS REQUIREMENTS DOCUMENT

Project Name: IRAP [Smart Contract]

Document Author: Kashif Rizwan / Usman Naqi

Date of Issue: DD MMM YYYY

Document Version: 0.0.1

Phase: 2.0

Table of Contents

Description	Page Number
1 Required Approvers	3
2 Executive Summary	3
3 Introduction	3
3.1 Objectives	3
4 Project Scope	4
4.1 In Scope	4
4.2 Out of Scope	6
5 Strategic Solution	6
5.1 Application 1 – Credit Scoring App	6
5.2 Application 2: IRAP [Smart Contract]	7
6 Data Requirements	7
7 KPIs to Measure	8
8 Information Architecture	9
9 Business Process and Procedures	
9.1 Current State	
9.2 Future State	
9.3 Process Flow	
10 Requirements	
10.1 Functional Requirements	
10.2 Non-Functional Requirements	

1 Required Approvers

Name, Title	Line of Business/Team

2 Executive Summary

Our platform will completely revolutionize the way how business owners look at the software projects budgeting. This tool will empower the businesses to estimate the time and cost of a software project with multiple team groups on just a click of a button and a few lines of input about what features are required in application. This will help businesses make informed decisions on which resources to deploy for each project with effective budgeting and decreased time to market. Our platform is solely focused on delivering effective solutions to software projects with a prediction about the X % percentage of the reusable code from organizations centralized code base that can be leveraged in new software projects saving a cost, time and other resources. This platform can also track the performance of the Engineering workforce on the basis of the defined KPIs. It provides a dashbaord to the leadership team with the visibility of KPI results for each individual in the engineering team.

3 Introduction

The purpose of this document is to outline the requirements of a business system that facilitates efficient software project time and cost estimation, predicting reusable code, resource tracking, and performance monitoring.

3.1 Objectives

Developing an AI-based platform that has the capability of estimating the cost and time required to complete a software project and generate smart contracts.

- The platform has the capability to track the performance of the Engineering workforce on the basis of the defined KPIs.
- The platform has the capability to do AI-based KPIs calculation considering the reasoning behind the values and the context along with the input data. The context defines the reasoning behind the unexpected input values.
- The platform will seamlessly integrate with multiple software tools including Jira, Confluence, GitHub, SonarQube, and Bamboo HR to collect the data required for measuring KPIs, analyzing the human resource cost.
- The platform owns the capability to understand, analyze, and predict the software project requirements from a given project statement either consisting of a few lines or a detailed SRD *(Software Requirement Document)*.

- The platform has the capability to take decisions on required technology stack and creates a detailed WBS (work breakdown sheet) with high level information about technical tasks required to be done.
- The platform owns the capability to suggest different teams to do the job based on the prior results and recommend the most effective team in terms of time and cost.
- The platform has the capability to make predictions about the reusable code from the organizations centralized code base.
- Empower resources by providing dashboards to monitor their individual KPIs.

4 Project Scope

4.1 In Scope:

1. Intelligent Resource Allocation Platform (IRAP)
    - Build a platform which allows technical leads to Input a Business Requirement Document at the start of a new project and in return it Outputs different Team Options that can execute the project.
    - Use an AI Model to extract relevant details from Business Requirement Document and run analysis considering four main factors: 1) Available Resource Profiles, 2) Project Cost 3) Forecasted Timelines 4) Code Reusability
    - After running analysis, generate Team options by incorporating a Tier Based System: Fastest Delivery + High Cost, Slowest Delivery + Low Cost and Optimized Resourcing Mix.
    - Build functionality to select a team from the given options and immediately generate resource contracts, Project IP Contracts and Product Requirement Document / Phase – Gate Material as a result.
2. Resource Scoring Model
    - Create a Resource Scoring Mechanism that is driven by Key Performance Indicators extracted from specific Performance Measurement Tools that are widely accepted by Leading Companies around the world.
    - These Performance Measurement Tools, from which relevant Data will be extracted, have been chosen to be Jira – Project Management, GitHub – Code Repositories and SonarQube – Code Quality as their adoption is around or more than 50% for the industry they operate in
    - Use an AI Model to analyze the extracted data from the performance management tools and based on logic recommend a score for the Resource that will be at the center of each Resource Profile
    - The Resource Score is to be derived in essence from the Credit Score Model which itself is built on an overtime building and depleting mechanism.

    - Continuously auto update score by putting in place machine to machine transaction system entirely removing the element of human reporting
    - Integrate platform with Bamboo HR to retrieve Academic Background, Professional Experiences, Certification as well as current Tech Stack and Hourly Rates of employees.
3. Code Reusability:
    - Develop a feature of Code Reusability that after receiving Input from Business Requirement Document tells the Technical Lead how much percentage of code requirement can already be satisfied from code generated during previous projects.
    - Use an AI Model to analyze and drive code requirements from a Business Requirement Document and run them against entire company's code repository with result being a code profile that can be reused by the Team once the project starts.
4. Project Timelines
    - When the Business Requirement Document is Input into the platform, generate an estimated project timeline.
    - Use an AI Model to analyze the Business Requirement Document and based on data available from previous documents and their eventual time to completion, suggest a project timeline for the new project.
5. Project Costing
    - Develop a feature of Project Costing which considers the Resource Profile that has been suggested for the execution of the project, picks up the Pay Rate for that resource along with the estimated project timeline and gives an estimated cost.
    - Repeat this costing exercise for each resource of the team to come up with a total combined labor cost for the project.
    - Use an AI Model to analyze the Business Requirement Document and determine based on previous data, what type of cloud, infrastructure, support costs will be associated with the project to come up with final costing of the project.
6. General
    - Develop the portal's dashboard, along with Resource + Manager Profiles and implement additional features such as notifications, profiles, and settings.
    - Display the results arising from KPIs, Scoring Mechanism, Employee Rankings, Resource Mix Suggestions, Project Estimations in UI components.
    - Develop a system for user authentication and user roles and authorities clearly defined.

4.2 Out Of Scope:

1. The timesheets of resources involved in a specific project can be recorded and consolidated under the portal for convenience.
2. The portal can made be capable of generating invoices based on the consolidated timesheets and other value-added services provided.
3. Processing any kind of payments according to agreed resource contracts
4. Monitoring and evaluation Resource or Team Member once the onboarding is completed and project begins.

5 Strategic Solution

To effectively achieve the features defined in the scope of the product, the idea is to develop two totally independent sub-applications with their own USPs. Both the applications can be used independently of each other under the umbrella of the parent product.

5.1 Application 1 – Credit Scoring App

- Credit Scoring application consists of a plugin's dev development. There will be a total of three plugins, one each for JIRA, GitHub, and SonarQube.
- This application will also have a Web Dashboard where the user's profiles can be viewed along with their credit scores and some history about the recent projects the individual has worked on.
- This application will also have an ***AI based*** server-side application that holds an ML model which runs the analyses on the data extracted by Plugins and generates a credit scores for the each individual in the global work force.
- When an organization installs these Plugins, the Plugins will extract the required information from the tools like JIRA, Github and SonarQube and pass the extracted information to the server-side application.
- Once the ML Model in the server-side application runs it’s analysis on the extracted data and update the credit scores against the individual profiles, those profiles can be searched on the web dashboard where users can see search those profiles and see the credit scores.
- The credit scoring application can serve as the data provider for the IRAP [Smart Contract] application. It provides the credit scores and the availability of the global work force that IRAP application can use when making suggestions on the team groups to do a project.
- ***Key USPs:***
    - Credit Scoring application can generate the certificate to an individual. The certificate mentions his/her credit score achieved during his career in the industry.
    - This certificate can also be presented at any platform to show case the performance he had during the projects he worked on.

- Or it can be asked by organization to showcase this certificate when hiring anyone to judge how the candidate was performing in his recent projects.

5.2 Application 2: IRAP [Smart Contract]

- IRAP [Smart Contract] applications consist of a web application and an AI based server-side application
- Web application provides the UI to upload a Business Requirement Document [BRD] and show the results of project timelines, the team groups with the predicted cost analysis and the suggestions about the reusable code to achieve the required project.
- An ***AI based server-side application*** that hold the three ML models
  - One for analysing a BRD, creating the Work Breakdown Sheet from the BRD, and estimating and planning the project timelines
  - 2nd one for suggesting the team groups based on the work force credit scores and availability it will get from the application 1 [Credit Scoring app]. Along with suggested team groups it will effectively calculate and show the cost involved in delivering the project with each group.
  - 3rd ML model will be responsible for analyzing the project key features and do the recommendations about reusability of the code to acheive those features instead of re-inventing the wheel.
- The web dashboard will also have the capability to view the user's detailed profile, a link to users LinkedIn profile, and his credit scores.

6 Data Requirements

This platform integrates with tools like JIRA, Github and SonarQube that will serve as a main source of input data. Below is the data required from each tool to train ML model that can smartly calculate the KPIs for the Engineering work force and generate the estimations for the required software projects.

6.1 JIRA

Epics	***Ticket [Story, Task, Bug, Improvement Ticket]***
All Epics under all projects in a JIRA Organization	Assignee
All tickets under all Epics of a project	Story Points
	Priority
	Start Timestamp
	QA Timestamp
	Closed Timestamp
	Count of Bugs Linked to Ticket
	Comments

	Change Logs

6.2 Github

Author
Creation Timestamp
Merged Timestamp
Closed Timestamp
Comments Count
Comments Content
PR Title
Source Branch Name

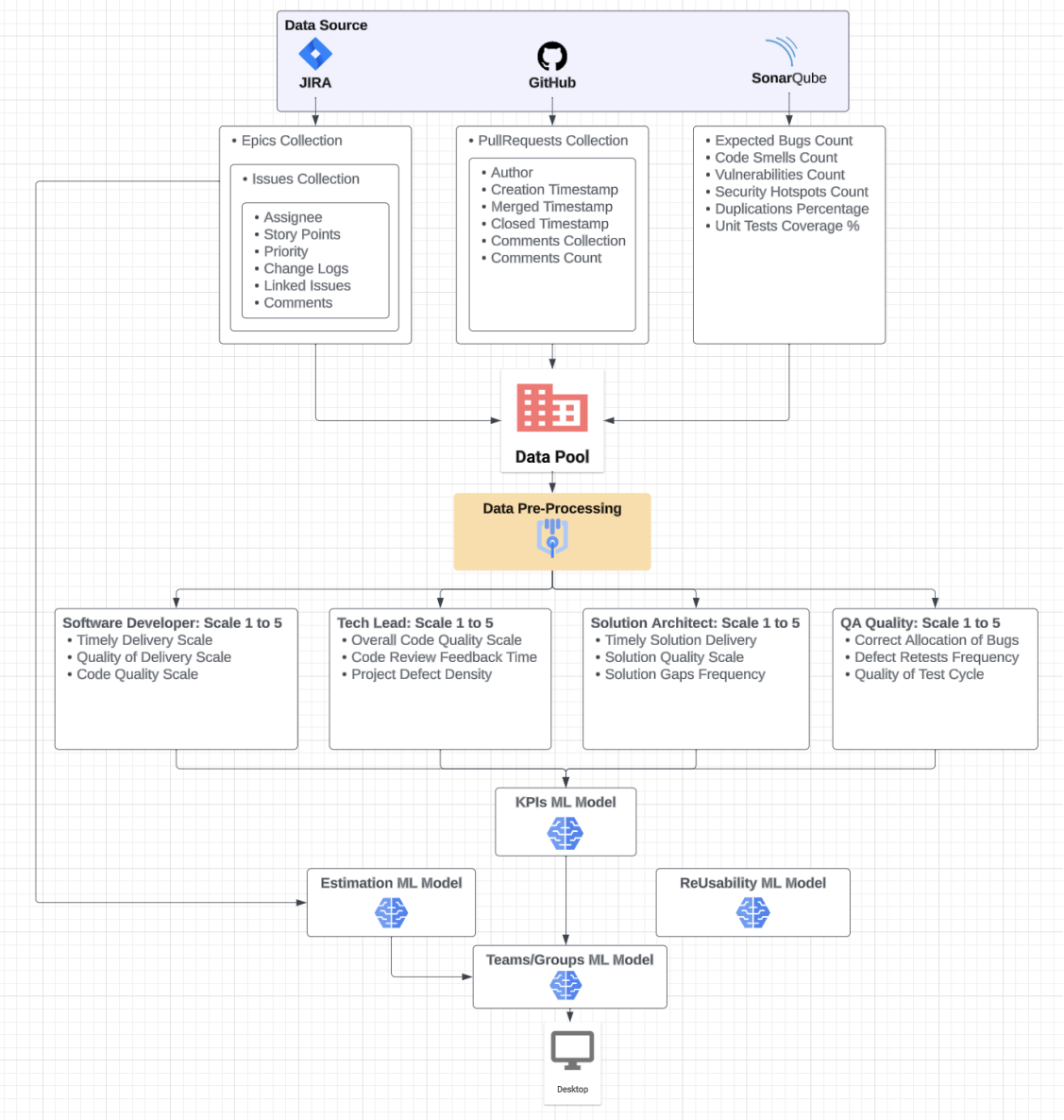
6.3 SonarQube

Code Smells Count
Duplicate Code %
Unit Tests Coverage %
Vulnerabilities Count
Security Hotspots Count
Probable Bugs Count

7 KPIs to Measure

Software Developer	**Tech Lead**	**Solution Architect**	**QA**
Timely Delivery	Defect Density	Timely Solution Delivery	Correct allocation of bugs
Quality of Delivery	Code Review Feedback Time	Impactful Solution Gaps	Defect Retests
Code Quality	Code Scalability		
Defect Resolution Time			

8 Information Architecture



Editable link here.

9 Business Process and Procedures

9.1 Current State

In the current state, mobileLIVE and other professional consulting service providers face challenges in dealing with Contractors and their timesheets. When Clients request validation of the work done by Contractors, it becomes a cumbersome process that involves manual checking of timesheets against the actual work delivered. This manual validation process is time-consuming and prone to errors, leading to delays in providing accurate reports to Clients.

The generation of a consolidated report that combines productivity and performance data of Contractors after project completion takes approximately four weeks to complete. Subsequently, Clients require an additional four weeks to review and approve the report, which then leads to the release of payments. Due to these inefficiencies, businesses like mobileLIVE often experience significant delays of 8 to 10 weeks in receiving payments.

Furthermore, mobileLIVE is responsible for delivering the entire scope of work (SOW) and meeting the relevant Key Performance Indicators (KPIs) as per the Master Services Agreements (MSAs) signed with Clients. However, a challenge arises as the Client can easily dictate their requirements to mobileLIVE, while mobileLIVE cannot easily enforce these requirements onto Contractors. The current timesheet model widely accepted in the industry lacks the flexibility to accommodate all the specific requirements outlined by the Clients.

9.2 Future State

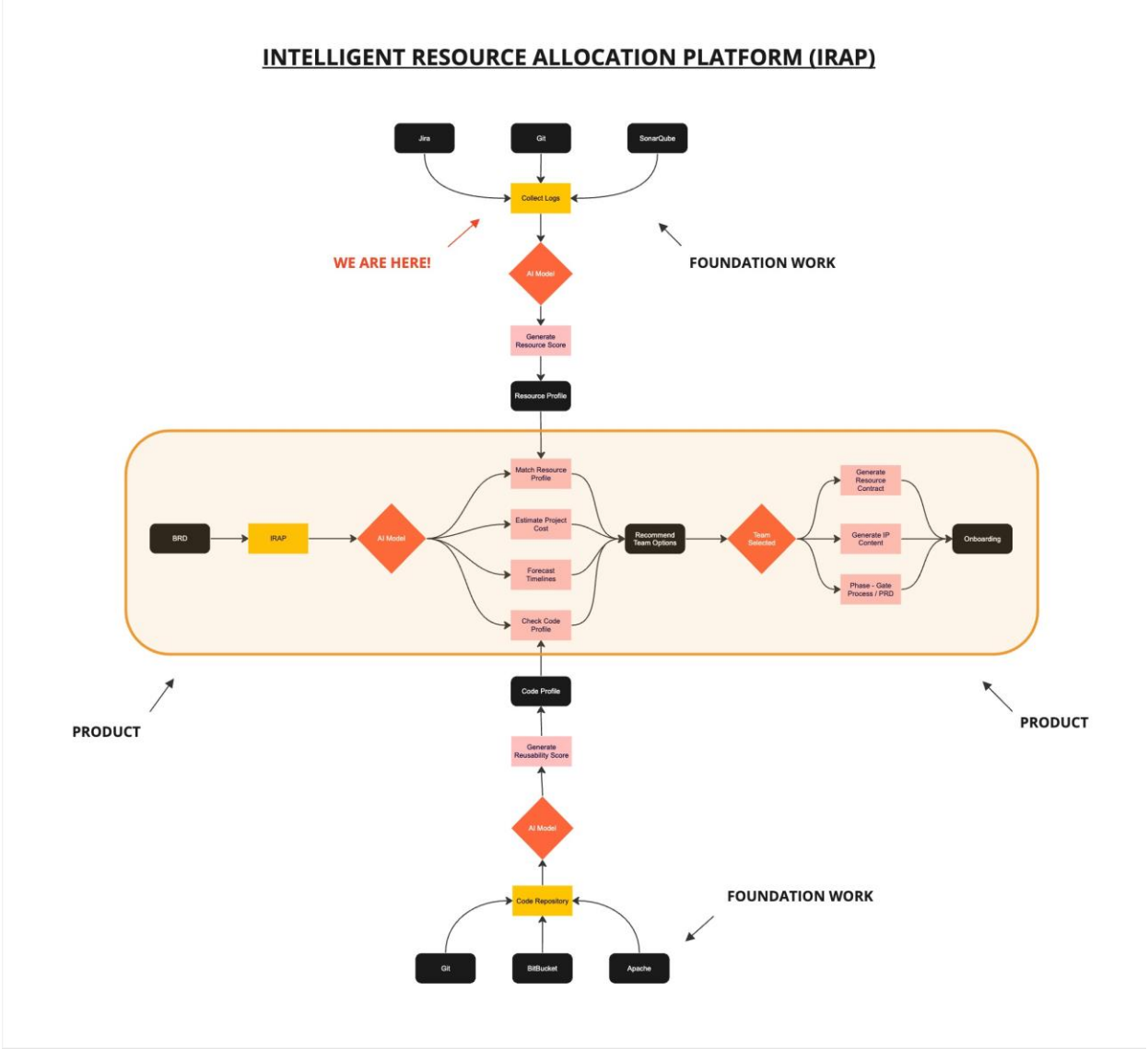
The future state of the system aims to address the challenges faced in the current state and introduce improvements to enable efficient project management, resource tracking, performance monitoring, collaboration, and invoicing processes. The key features and enhancements in the future state are:

- The system will provide functionalities for generating smart contracts that define project scope, deliverables, timelines, and associated KPIs.
- Resource KPIs will be tracked automatically, providing real-time visibility into resource performance and project progress.
- KPIs extracted from smart contracts will be utilized to track resource performance and measure their adherence to project objectives.
- A performance management tool will be implemented to monitor resource productivity, identify areas of improvement, and provide actionable insights.

- Seamless data exchange and collaboration between the project management system and external tools like Jira, Confluence, Github, Sonarqube, and Bamboo HR will streamline workflows and improve efficiency.
- Resources will have better visibility into their contributions, fostering a sense of ownership and accountability.
- Managers will have access to consolidated reports which will provide valuable insights into resource utilization, project milestones, financials, and overall project health.
- The system will prioritize data privacy and security measures, ensuring compliance with relevant regulations.

  These future state enhancements will address the current pain points and provide a more effective and productive environment for mobileLIVE and its stakeholders.

9.3 Process Flow: Only processes enclosed in the box to be captured within the umbrella of POC



10 Requirements

10.1 Functional Requirements

Req#	Requirement	Requested By	Priority
1	Systems should be analyzed to extract relevant KPIs for	mobileLIVE	H

	each resource involved in the project.		
2	A performance management tool should be provided to track and update the extracted KPIs for each resource.	mobileLIVE	H
3	Integration with external tools like Jira and Git should be established to link tasks/commits to resource KPIs and update them accordingly.	mobileLIVE	H
4	The system should display a dashboard for resources to monitor their individual KPIs, track performance, view progress, and identify areas for improvement.	mobileLIVE	H
5	Resources should have the ability to generate timesheets based on their recorded performance data, with the option to include additional details and submit them for approval.	mobileLIVE	L
6	Resources should be able to generate invoices based on approved timesheets	mobileLIVE	L

	and associated rates, with the capability to review and finalize invoice details before submission.		
7	The system should provide managers with an overall project-wide dashboard to monitor project status, key metrics, and performance indicators.	mobileLIVE	H
8	Managers should be able to generate consolidated performance reports for clients, selecting desired parameters, reviewing the reports, and sharing them with clients.	mobileLIVE	L
9	Managers should have the ability to generate consolidated invoices for clients based on approved timesheets and billing rates, with the capability to review and finalize invoices before submission.	mobileLIVE	L
10	The system should provide a form or interface to add/update employee profiles with fields such as name, contact information,	mobileLIVE	H

	job title, and department.		
11	Each employee profile should have a unique identifier stored on the blockchain.	mobileLIVE	H
12	Authorized users should be able to view employee profiles stored on the blockchain.	mobileLIVE	H
13	The system should provide a user-friendly interface to access and retrieve employee information securely.	mobileLIVE	H
14	Privacy controls should be implemented to ensure that only authorized users can access specific employee profiles.	mobileLIVE	H
15	The system should support the addition of education, skills, experience, and certificates to employee profiles. These details should be securely associated with the respective employee's profile and stored on the blockchain.	mobileLIVE	H

16	The system should allow authorized users to upload contract files related to employee agreements, such as employment contracts or non-disclosure agreements.	mobileLIVE	H
17	Access controls should be implemented to restrict viewing and editing rights to authorized parties only.	mobileLIVE	H
18	The system should provide a streamlined process for onboarding new employees.	mobileLIVE	H
19	The system should automate and track the onboarding process, recording each step on the blockchain for transparency and auditability.	mobileLIVE	H
20	The system should enable efficient management of employees' data. It should allow authorized users to search, filter, and sort employee profiles based on various criteria.	mobileLIVE	H

21	The system should provide role-based access controls to manage user permissions and restrict access to sensitive functionality and data.	mobileLIVE	H
22	The system should enforce the assigned roles and permissions, ensuring that users can only perform actions aligned with their designated roles.	mobileLIVE	H
23	Data Retrieval from Jira	mobileLIVE	H
23.1	Retrieve all Epics associated with the project.	mobileLIVE	H
23.2	Extract all stories authored by a specific Author under each Epic.	mobileLIVE	H
23.3	Fetch all relevant ticket types (Stories, Bugs, Tasks, Improvement) with their details.	mobileLIVE	H
23.4	For each Ticket (Story, Task, Improvement): • Retrieve the Assignee. • Collect Story Points. • Capture Start Timestamp.	mobileLIVE	H

	• Record QA Timestamp. • Note Closed Timestamp. • Count and list Bugs linked to the Ticket. • Capture Comments associated with the Ticket.		
23.5	For each Bug: • Retrieve the Assignee. • Categorize Severity. • Record Start Timestamp. • Note QA Timestamp. • Capture Closed Timestamp. • Collect Comments associated with the Bug.	mobileLIVE	H
24	Data Retrieval from Github: • Extract details related to Pull Requests (PRs). • Gather PR Author, Creation Timestamp, Merged Timestamp,	mobileLIVE	H

	and Closed Timestamp. • Count and list Comments associated with each PR. • Capture Comments' content.		
25	Data Retrieval from SonarQube: • Code Smells. • Duplicate Code Blocks. • Unit Tests Coverage Percentage. • Vulnerabilities • Code Debt.	mobileLIVE	H
26			

10.2 Non-Functional Requirements

Req#	Requirement	Requested By	Priority
1	The system should ensure the privacy and security of data.	mobileLIVE	H
2	The system should be accessible from various devices and web browsers.	mobileLIVE	H
3	Appropriate user roles and permissions	mobileLIVE	H

	should be provided by the system.		
4	The system should have a user-friendly interface that is easy to navigate.	mobileLIVE	H
5	The system should be scalable to accommodate future growth and increasing project volumes.	mobileLIVE	H
6	The deployment architecture should support easy deployment across different environments, such as cloud platforms, on-premises servers, or hybrid setups.	mobileLIVE	H
7	The system should be designed to ensure high availability, minimizing downtime and service interruptions.	mobileLIVE	H
8	The system should have recovery mechanisms in place, including regular backups, data replication, and failover capabilities.	mobileLIVE	H
9	The infrastructure should provide sufficient computational resources, such as processing power,	mobileLIVE	H

	memory, and storage, to support the AI and blockchain components effectively.		
10	The infrastructure should adhere to applicable regulations and compliance requirements, such as data protection laws and industry-specific regulations.	mobileLIVE	H
11	The infrastructure should support interoperability with other systems and technologies.	mobileLIVE	H