EMGT 5220: Engineering Project Management

Summer-1 (2024)

Perq’s with Zelle: Seamless Financial Transactions with Rewards

Project Proposal Draft-Final

Team 2

Team Point of Contact: Nayana Magadi Nagaraj
Email: magadinagaraj.n@northeastern.edu

Team Members List:
Kumar, Rohith
Narendra Kumar, Vinay
Magadi Nagaraj, Nayana
Ravishankar, Abhijith Raj Urs
Yogendrakumar, Pavan Thaloor
Bangalore Ramachandra, Harshitha

Table of Contents

Letter of Transmittal i

Executive Summary ii

1.0 Introduction 1

1.1 Problem 1

1.2 Solution 1

2.0 Purpose & Objectives 2

2.1 Purpose 2

2.2 Objectives 2

3.0 Technical Overview 3

4.0 Implementation Plan 5

4.1 WBS 5

4.2 Schedule 5

4.3 Responsibility Chart 6

4.4 Resource Allocation 6

4.5 Stakeholders 6

5.0 Execution Plan 8

5.1 Project Monitoring 8

5.2 Project Control 8

5.3 Project Auditing 10

5.4 Project Closure 11

6.0 Risk Assessment Management Plan 12

7.0 Financial Plan with Budget 14

7.1 High-Level Details 14

7.2 Budget Justification 15

8.0 Team Credentials 16

References 17

Appendices 18

Appendix A: Work Breakdown Structure 18

Appendix B: Project Schedule 20

Appendix C: Responsibility Chart 23

Appendix D: RACI Matrix 24

Appendix E: Budget Justification 27

Appendix F: Resource Allocation Plan 30

Letter of Transmittal

21/06/2024
Prof. Himlona Palikhe
College of Engineering, Northeastern University
Boston, MA 02115

Dear Prof. Palikhe,

On behalf of Team 2, we are excited to present the final draft of our project proposal titled "Perq's with Zelle: Seamless Financial Transactions with Rewards" for the EMGT 5220 – Project Management course, Summer-1 2024. This proposal represents our dedicated efforts, innovative solutions, and teamwork aimed at enhancing Zelle's financial transaction system with a focus on digital integration and user-centric features.

Our proposal addresses the challenges of enhancing Zelle with new functionalities. "Perq's with Zelle" aims to improve user experience by facilitating seamless peer-to-peer transactions, direct merchant payments, and personalized reward systems. The execution plan outlines our approach to monitoring, controlling, auditing, and concluding the project, ensuring comprehensive oversight and flexible management. We have also included a risk assessment plan to proactively address potential challenges and ensure project success.

The attached final draft includes the Executive Summary, Execution Plan, Risk Assessment Management Plan, and other relevant sections. Each part is aligned with our project's goals, supported by thorough research, and presented professionally.

We hope our proposal meets your expectations and the academic standards of Northeastern University. Our team is committed to applying effective project management principles and contributing to both academic and practical advancements in our field.

Thank you for the opportunity to work on this project and for your guidance throughout the process. We look forward to your feedback and to advancing our project in a way that significantly enhances the future of financial transactions.

Sincerely,
Bangalore Ramachandra, Harshitha
Kumar, Rohith
Magadi Nagaraj, Nayana
Narendra Kumar, Vinay
Ravishankar, Abhijith Raj Urs
Yogendrakumar, Pavan Thaloor

Executive Summary

The "Enhancing Zelle App with Perq: Transforming Financial Transaction Management" project represents a strategic endeavor aimed at revolutionizing digital financial transactions and enhancing the user experience through innovative features and seamless integration. In today's dynamic digital market, Zelle, despite its broad reach, faces a number of problems that prevent ideal user engagement and satisfaction. These include fragmented features across platforms, limited customization options, and less-than-ideal integration with other financial services, collectively impacting user convenience and overall satisfaction.

Perq's approach focuses on enriching the Zelle app with advanced features designed to streamline peer-to-peer transactions, integrate merchant payments seamlessly, introduce personalized rewards systems, enable multi-bank account linkage, and establish a robust digital wallet functionality. These enhancements are strategically aimed at addressing identified pain points in current user experiences, with the goal of significantly increasing user adoption and engagement.

To achieve these goals, the technical implementation plan builds on the existing Zelle infrastructure while providing new modules for rewards management, digital wallet functionality, and merchant payment capabilities. Development adheres to industry-standard technologies and Agile methodologies, ensuring iterative progress, continuous integration of user feedback, and rigorous testing to ensure reliability, scalability, and strict security standards for protecting user data and financial transactions.

The project is funded with an initial investment of $596,828.47, which will cover development expenses, deployment costs, and initial marketing activities. The estimated return on investment includes $582,489.6 in annual revenue from service fees, resulting in a payback period of approximately 1.02 years. This swift payback period underscores the financial viability and potential profitability of the project, justifying the initial expenditure and highlighting its capacity to generate substantial returns within a short timeframe.

Ultimately, the project aims not only to address present market gaps and customer expectations but also to establish Zelle as a leading platform in the competitive arena of digital financial services. Perq aims to create new standards for financial transaction management by providing a seamless, safe, and user-centric experience, opening the path for increased user adoption, engagement, and satisfaction in the digital era. This initiative has the potential to transform how users interact with digital financial services, encouraging innovation and setting new standards for excellence in financial transaction management.

1.0 Introduction

1.1 Problem

In today's fast-paced digital environment, people demand smooth and efficient financial transaction management systems. Despite Zelle's enormous network, many users still underutilize it owing to a lack of critical features that improve user experience and ease. According to recent research (*Payment-Trends*, n.d.), customers want comprehensive solutions for quick peer-to-peer transactions, direct merchant payments, and tailored rewards. The difficulty caused by financial services over multiple platforms makes it harder for consumers to manage their finances and lowers their satisfaction. The dispersion of financial services among numerous applications hampers the user experience by requiring customers to manage their accounts across multiple platforms. This difficulty not only upsets customers, but it also decreases their involvement with any particular platform, lowering their overall satisfaction and efficiency in financial management.

1.2 Solution

Perq's solution to the challenge involves combining several financial services into a single, user-centric mobile application, which will be accomplished through thorough research, design, and testing. Perq will enhance the Zelle app by boosting peer-to-peer transactions and smoothly integrating direct merchant payments, customized incentives, multi-bank account linkage, and a digital wallet feature.

Perq will monitor customer transaction preferences and adapt rewards accordingly, offering personalized coupons or cashback to encourage frequent use and deeper engagement. A collaborative team of professionals will use industry-standard technologies and an iterative development method, incorporating continual feedback and testing to ensure the app meets user demands.

Perq's technique demonstrates optimism in addressing a challenge where others have failed. This confidence originates from its extensive rewards system and simple payment options, which are not available in other alternatives. With the worldwide mobile payment industry expected to exceed $12.06 trillion (*Mobile Payment Market Size - Share & Industry Report | 2027*, n.d.), Perq is well-positioned to acquire a significant market share by closing these important gaps and offering a better user experience.

2.0 Purpose & Objectives

2.1 Purpose

The purpose of this is to enhance user adoption and experience by prioritizing security and addressing diverse financial needs through a holistic platform. By integrating innovative features that bridge current gaps in financial applications, the project aims to overcome existing limitations and improve Zelle as a financial application, emerge as the favored choice for users, merchants, and financial institutions seeking independent and convenient transaction solutions. The project hopes to pave the way for Zelle to become a more appealing, convenient, and popular mode of payment, hence increasing user adoption and engagement.

2.2 Objectives

The primary objective is to enhance the current app by integrating all the existing features into one comprehensive app to differentiate from competitors and drive increased usage to make the current application user-friendly financial tool by improving peer-to-peer transactions, integrating direct merchant payments, and creating a personalized incentives system. Additionally, aiming to integrate various bank accounts for added flexibility, implement a digital wallet function, and establish strong security measures to protect user data and transactions. Strategic partnerships and market research will enhance reliability, expand capabilities, and develop personalized rewards and programs to incentivize users. The focus is on boosting the peer-to-peer transaction success rate by at least 20% within a year and within six months, aim to have 25% of active customers use the digital wallet feature, benefiting consumers and corporate entities with tailored solutions and valuable insights.

Below are the sub-objectives:

Zelle:

- To identify and express the primary challenges contributing to the application's underutilization, despite its enormous network.
- Enhance the overall user experience of Zelle by streamlining the app's interface and functionality.

Customer:

- Provide enhanced security, convenience, and ease of use for payments through the application, alongside offering exciting rewards and benefits to incentivize extensive usage.
- Implement features that allow for personalized recommendations and tailored experiences based on individual user preferences and behaviors, such as customized wallets, notifications, and alerts.

3.0 Technical Overview

The development of the application, integrated with Perq and existing Zelle features, requires a robust and comprehensive technical framework. This section details the technical requirements and considerations across various aspects of the application, including mobile UI, database management, front-end and back-end development, and hosting solutions.

3.1 Mobile UI/UX Design

The user experience is pivotal in developing a user-friendly, intuitive, and engaging interface that boosts user engagement and adoption. Leveraging design tools such as Figma, the design process will incorporate responsive design principles to ensure seamless functionality across various devices and screen sizes. Navigation will be simplified with clear, consistent menus and buttons to streamline the user experience, while maintaining the existing theme, color palette, and design practices to preserve authenticity with the existing UI. Continuous user research, including surveys and testing, will guide design iterations, optimizing user flows for tasks such as sending/receiving money, accessing the digital wallet, and redeeming rewards.

3.2 Database Management

Since the existing Zelle application is being enhanced, the current database infrastructure will be leveraged, while new tables and data for rewards, wallets, and transaction information will be added. This approach ensures high performance and reliability by adhering to the established best practices already used by Zelle. Security measures, including encryption for data at rest and in transit, along with strict access control policies and role-based access to sensitive data, will be maintained. The updated data models will incorporate user profiles, transaction histories, and rewards tracking, ensuring scalability to handle high volumes of transactions and concurrent users. This strategy will ensure seamless and secure data operations while integrating the new features.

3.3 Application Development

The application development team will create a responsive, interactive, and secure mobile application by following Zelle's established practices and utilizing the existing development team. They will utilize modern frameworks like React Native or Flutter for cross-platform front-end development, leveraging JavaScript or Dart. APIs for transaction processing, rewards management, and third-party services will ensure seamless functionality, while state management tools like Redux or MobX and comprehensive testing with Jest, Mocha, or Jasmine will deliver a polished user experience. For the back-end, the team will build secure and scalable infrastructure using frameworks such as Node.js with Express, Django, or Spring Boot, and languages like JavaScript, Python, or Java. RESTful or GraphQL APIs will enable front-end and back-end communication. Secure authentication methods like OAuth 2.0 and multi-factor authentication will protect user data, and real-time transaction processing will integrate smoothly with financial institutions and payment gateways. This approach ensures robust functionality and security for the mobile application.

3.4 Hosting Infrastructure and Security Measures

The hosting infrastructure and security measures will ensure the reliable, scalable, and secure operation of the current application, incorporating both existing architecture and frameworks and supporting the newly proposed features.

The application will leverage cloud services like AWS, utilizing compute resources such as AWS EC2 and EBS. Development and deployment processes will be streamlined through a continuous integration and deployment pipeline using tools like Jenkins, GitHub Actions, or GitLab CI/CD. Monitoring and logging solutions, including AWS CloudWatch, will track performance and identify issues, ensuring the application's reliability and scalability.

The application will employ data encryption using AES-256 and SSL/TLS to secure sensitive information both at rest and in transit.

4.0 Implementation Plan

4.1 WBS

The Work Breakdown Structure (WBS) for the project defines the complete set of activities organized into four major phases: project initiation, project planning, project execution, and project closing

4.1.1 Project Initiation Phase

This phase involves defining the project concept, goals, and objectives. It includes conducting a kick-off meeting, identifying stakeholders, developing the project charter, and performing a feasibility study along with an initial risk assessment. Leadership roles are assigned based on team members' experience.

4.1.2 Project Planning Phase

In this phase, the focus is on detailed planning activities such as identifying customer and product requirements, creating a comprehensive project schedule, and planning for resource management. Necessary tools and environments for development are set up, risk management strategies are established, and communication plans are developed to ensure effective coordination throughout the project.

4.1.3 Project Execution Phase

This phase covers the actual development work, including frontend and backend development, database setup, and testing. Frontend tasks involve UI/UX design and coding, while backend tasks include server setup, API implementation, and database configuration. Testing ensures the solution meets all requirements, and performance monitoring helps keep the project on track. This phase also includes training, deployment, and initial marketing activities.

4.1.4 Project Closing Phase

In the final phase, the application is launched, and deployed to the live environment, and user support and maintenance are provided. Compliance checks are conducted, servers are managed, and performance is optimized. The project is concluded by obtaining stakeholder sign-off, documenting lessons learned, releasing resources, conducting a post-implementation review, and preparing for client handover.

Please refer to Appendix A for the detailed Work Breakdown Structure (WBS).

4.2 Schedule

To ensure that the project is completed successfully, a schedule was created as a tool for monitoring and overseeing project activities. Based on previous experiences with comparable initiatives, activity durations were limited to two weeks or less.

Microsoft Project contains the timetable. This enables the project manager to accurately depict progress using the Gantt chart tool and adjust task durations throughout the project. Appendix B contains the baseline project Gantt chart.

4.3 Responsibility Chart

The RACI Matrix ensures duties are assigned wisely, preventing workload imbalances and clearly identifying who is in charge of each activity. For instance, the development team is responsible (R) for coding new features like the digital wallet and customized rewards system, while the project manager is accountable (A) for overall project delivery. UI/UX designers are consulted (C) to ensure new features integrate seamlessly with the existing design, and financial institutions are informed (I) about relevant progress. There is an additional point of contact from Zelle, the business head, who will oversee the project, provide strategic direction, and ensure alignment with their objectives.

You can find the responsibility chart in Appendix C and D

4.4 Resource Allocation

Strategic resource allocation across multiple domains is necessary for the Perq's with Zelle project to be executed successfully. The importance of human resources makes the dedication of qualified experts—project managers, Product managers, business analysts, technical managers, IT and security managers, analysts, systems architects, engineers, and marketing experts—necessary. From the beginning to the end of the project, these people will cooperatively offer their skills at various phases. Financial resources are also essential for paying for expenses associated with system integration, supplier onboarding, and software development. Effective resource management will ensure the project's smooth progress and promote innovation, quality control, and long-term expansion in the financial services sector. To account for unexpected challenges that might arise during the project, an additional 8% has been added to the budget. This will guarantee that resources are available to deal with any unexpected issues and successfully reduce risks.

For a detailed perspective; please refer to Appendix F.

4.5 Stakeholders

The following list summarizes the stakeholders mentioned in this report;

Project Team

- Project Manager - Pavan Thaloor Yogendrakumar
- Product Manager - Rohit Kumar
- Technical manager- Abijith Raj
- Business Analyst - Vinay N Kumar
- IT and Security Manager - Nayana Magadi Nagaraj
- Marketing Manager - Harshitha Bangalore Ramachandra

Development Team

The development team is essential for creating, testing and deploying the mobile application.

- Software Development Manager
- Full Stack Developers (Frontend and Backend Developers)
- UI/UX Designers
- QA Engineers

- Database Administrators
- Data Architect

Technical Support Team

The team responsible for aiding and supporting the project is composed of the following members of the Support team.

- IT Managers
- IT Administrators
- Application Support Analyst

Security Team

- Security Analyst
- Database Security Specialist
- Database Backup and Recovery Specialist
- Network security Administrator
- Cybersecurity Specialist

Investors/Sponsors

Investors/sponsors contribute financial assistance to Perq by investing funds to develop mobile applications.

- Zelle - Business Head

5.0 Execution Plan

5.1 Project Monitoring

Project monitoring in the Perqs project comprises regular oversight to verify that project activities are on track, risks are controlled, and objectives are accomplished. The monitoring process comprises maintaining key performance indicators, performing frequent evaluations, and taking remedial action as needed

The key features of project monitoring are:

Timely Implementation: To ensure that the perqs software is developed, tested, and implemented on time, project milestones are tracked against the schedule, any delays are addressed, and the causes of delays are identified. Through regular testing and performance monitoring, we ensure that the Perqs app satisfies all quality criteria. This involves both frontend and backend development, with thorough testing of UI/UX design, server setup, and database configuration to ensure that user needs are met.

Integration and Functionality: Ensuring that the perqs software integrates seamlessly with current systems by constantly tracking the integration process, running frequent tests, and quickly fixing any issues that arise.

Risk Management: Identifying, analyzing, and reducing possible risks throughout the project's lifespan. Regular risk assessments and revisions to contingency plans guarantee that the project can successfully deal with unexpected challenges.

Cost Efficiency: Cost efficiency is regularly reviewing project expenses to ensure they remain within budget, comparing costs to industry norms, and making appropriate modifications.

Stakeholder Engagement: Keeping stakeholders informed and engaged with frequent updates and evaluations. This ensures that all parties are aligned with project objectives and that any challenges are addressed collectively.

5.2 Project Control

Project control in the Perqs project ensures adherence to the plan, budget, and quality standards through systematic monitoring and strategic oversight. Key measures include phase-gated reviews of deliverables at milestones, weekly progress tracking with Cost Variance and Schedule Variance metrics, regular team meetings for issue resolution, bi-weekly leadership meetings for strategic decisions, and continuous risk assessment managed through updated risk registers. Technical teams also meet regularly to address progress and technical challenges. These measures collectively ensure the perq project stays on schedule, within budget, and meets its objectives effectively.

Control Type	Description	Frequency	Record/Output	Responsibility
Phase-Gated Controls	Reviews at critical milestones to verify that deliverables meet required standards before proceeding.	At each phase	Phase review reports and approval documentation	Project Manager
Weekly Status Reports	Tracking project progress through weekly reports to monitor ongoing activities and identify issues early.	Weekly	Weekly status reports, progress updates	Project Manager
Cost and Schedule Controls	Monitoring cost and schedule performance using metrics like Cost Variance (CV), Cost Performance Index (CPI), Schedule Variance (SV), and Schedule Performance Index (SPI).	Weekly	Cost and schedule performance reports, variance analysis	Project manager
Technical Review Meetings	Regularly scheduled meetings where technical teams review and discuss progress, technical challenges, and solutions	Weekly/Bi-weekly	Meeting minutes, action items, technical reports	Technical manager
Regular Team Meetings	Facilitating discussions on progress and issue resolution through regular team meetings.	Weekly	Meeting minutes, action items	Project Manager
Leadership Meetings	Strategic decision-making and resource allocation discussions with leadership every 2-3 weeks.	Bi-weekly	Meeting minutes, strategic decision records	Project Manager and Leadership
Risk Assessment and Monitoring	Continuously identifying and mitigating potential risks by updating a risk register and matrix.	Continuous	Risk register, risk matrix, mitigation plans	Project Manager
Detailed Documentation	Maintaining transparency and supporting audits by documenting all control activities to foster continuous improvement.	Continuous	Project documentation, audit logs, improvement reports	Project Manager

5.3 Project Auditing

Project auditing is a systematic and impartial evaluation of a project's numerous parts to assess its performance, identify risks, and verify that it is on track to meet its objectives. The audit process for the Perqs project will focus on examining technical, status, and risk components throughout the project's lifespan. This will guarantee that the project meets its objectives, keeps on time, stays under budget, and successfully manages possible risks.

Heading	Audit type	Objective
Front-End and Back-End Review	Technical Overview	Assess the efficiency and functionality of backend components and user interface.
Database Analysis	Technical Overview	Evaluate the performance, security, and scalability of the database system.
Budget and Timeline Compliance	Project Status Overview	Verify adherence to project budget and timeline.
Stakeholder and Team Coordination	Project Status Overview	Ensure alignment between the RACI matrix and actual roles and responsibilities.
Quality Assurance and Change Control	Project Status Overview	Review the effectiveness of quality assurance processes and change management protocols.
Risk Identification and Mitigation	Risk Audit	Assess the effectiveness of risk identification and mitigation strategies.
Risk Impact Evaluation	Risk Audit	Examine the accuracy and thoroughness of risk impact assessments.
Contingency Planning Assessment	Risk Audit	Review the sufficiency of contingency plans.
Project Objectives Achievement	Final Audit	Evaluate the overall accomplishment of project goals.
Lessons Learned Documentation	Final Audit	Verify the documentation of lessons learned and the frequency of quality audits.
Recommendations for Improvement	Final Audit	Identify areas for enhancement and provide optimization recommendations.

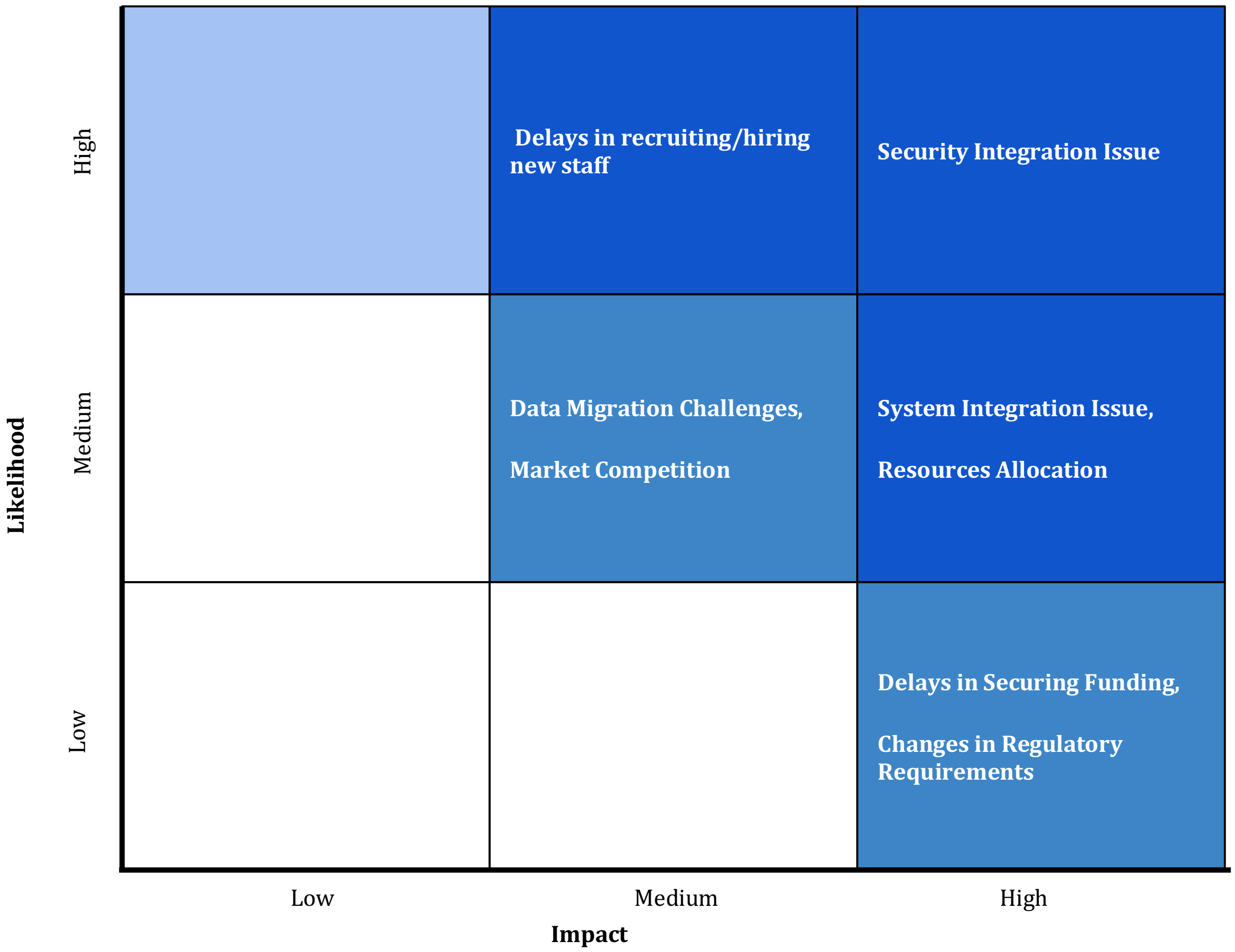
5.4 Project Closure

Project closure for the perqs project includes completing all project operations and formally accepting outputs. This phase guarantees that the perqs software is fully functional and integrated, with all deliverables distributed to stakeholders. A final audit evaluates project success, budget, and schedule adherence, and notes lessons learned for future projects. Documentation, including design documents and user manuals, is saved for future reference. Stakeholder sign-off validates that deliverables fulfill requirements. Team performance is assessed through feedback and acknowledgment, and resources are released for reassignment. A post-implementation support strategy is created. Finally, celebrating project completion enhances confidence and develops a healthy project culture, resulting in a seamless closure and useful lessons for future projects.

6.0 Risk Assessment Management Plan

The project's elementary risks are grouped by likelihood and impact. Each risk is associated with a mitigation strategy and an action plan for addressing it. These risks include system integration issues with Zelle infrastructure, security vulnerabilities in new features, data migration challenges, delays in obtaining funding or investor support, resource allocation issues, budget overruns due to unexpected costs, changes in regulatory requirements, and market competition. The action plans include regular integration tests, constant security monitoring, close monitoring of the data migration process, regular stakeholder updates, resource allocation reviews, budget reviews, legal reviews, and market competition analysis to modify approaches as needed.

Risk Description	Likelihood	Impact	Response	Action Plan
System integration issues with existing Zelle infrastructure	Medium	High	Mitigate	Regular integration tests, feedback from developers
Security vulnerabilities in new features	High	High	Mitigate	Continuous security monitoring, periodic audits
Data migration challenges	Medium	Medium	Mitigate	Monitor data migration process closely, conduct reviews
Budget overruns due to unforeseen costs	Medium	High	Mitigate	Regular budget reviews, financial audits
Delays in securing funding or investor support	Low	High	Mitigate	Regular updates to stakeholders, monitor funding status
Resource allocation issues leading to delays	Medium	High	Mitigate	Regular resource allocation reviews, adjust as needed
Changes in regulatory requirements	Low	High	Mitigate	Regular legal reviews, compliance checks
Market competition reducing user interest	Medium	Medium	Mitigate	Analyze market competition, adjust strategies



Likelihood			
High		Delays in recruiting/hiring new staff	Security Integration Issue
Medium		Data Migration Challenges, Market Competition	System Integration Issue, Resources Allocation
Low			Delays in Securing Funding, Changes in Regulatory Requirements
	Low	Medium	High
		Impact	

Fig

6-1: Risk Matrix

7.0 Financial Plan with Budget

7.1 High-Level Details

Perq, aimed at enhancing the Zelle application, requires an extensive financial plan to ensure the successful execution of the project. This budget guarantees adequate funding for every aspect of the project, from development to user acquisition, thereby supporting the goal of improving user adoption and experience with the Perq-enhanced application. Perq's budget summary is given below; refer to Appendix E for complete budget justification.

Project Budget Overview	
Total Labour Cost	$476,869.00
Total Material Cost	$22,467.84
Total Miscellaneous Cost	$6,450.00
Contingency	$40,462.95
Administrative Cost	$50,578.68
Total Project Budget	$596,828.47

Table 7-1 Project Budget Overview

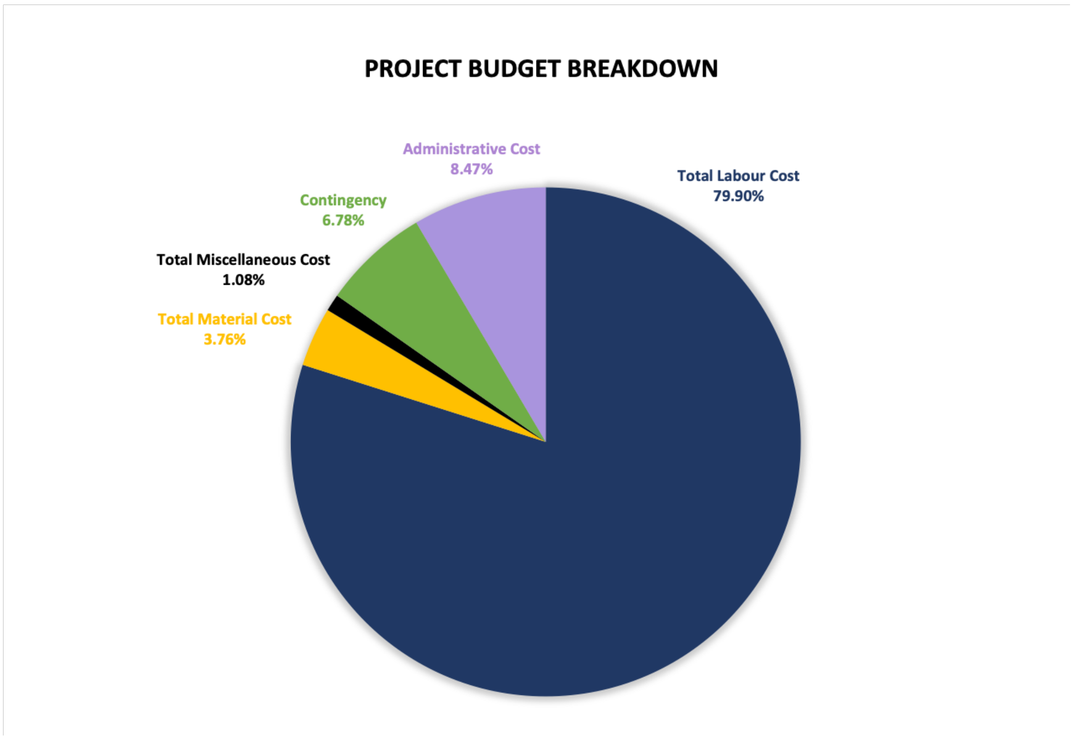


Figure 7-1 Project Budget Breakdown Chart

7.2 Budget Justification

The payback period for the $596,828.47 initial investment in the Zelle financial app development is estimated to be approximately 1.02 years based on projected annual cash flow. Targeting an extensive user base of 60 million, with an anticipated adoption rate of 8%, resulting in 4.8 million users. With each user projected to conduct an average of 44 transactions annually, the total number of transactions would amount to 211.2 million per year. At a service fee rate of 0.001% per transaction and an average transaction value of $275.8, the annual revenue from service fees is estimated at $582,489.6. Consequently, the payback period for this investment is calculated to be approximately 1.02 years. This swift payback period underscores the financial viability and potential profitability of the project, justifying the initial expenditure and highlighting its capacity to generate substantial returns within a short timeframe. (For a detailed budget justification; please refer to Appendix E)

Engineering Economic Technique

Initial Investment: Total Initial Investment: $596,828.47
User Base:

- Total Users: 60M
- Adoption Rate: 8%
- Adopted Users: 4.8M (60M * 8%)

Transactions:

- Transactions per User: 44 transactions/person/year
- Total Transactions: 211.2M transactions/year (4.8M users * 44 transactions/user)

Revenue from Service Fees:

- Service Fee per Transaction: $275.8/transaction * 0.001%
- Annual Revenue: $582,489.6/year (211.2M transactions * $275.8/transaction * 0.001%)

Payback Period Calculation:

- Payback Period= Total Investment/Annual cash flow
- Payback Period: 1.02 years ($596,828.47 / $582,489.6)

Conclusion: The payback period is 1.02 years.

8.0 Team Credentials

Bangalore Ramachandra, Harshitha

Harshitha Bangalore Ramachandra is currently pursuing a Master's of Science in Engineering Management at Northeastern University. Harshitha brings a strong foundation to the program with her Bachelor's degree in Information Science and Engineering. Professionally, she has worked for Amazon for more than two years in the Global Solutions and Risk team. She has led projects that have improved operational efficiency by 1.5 hours, raised revenue by 20% while using fewer resources, and improved data accuracy by 18%. She also possesses prior expertise in guiding and instructing newcomers, ensuring a smooth onboarding process.

Kumar, Rohit

Rohith is a dedicated graduate student pursuing Engineering Management at Northeastern University, building on his robust technical foundation with a Bachelor's degree in Computer Science & Engineering. He has successful internships focused on areas of Database design & Software development. Beyond his core technical expertise, Rohith has demonstrated his organizational skills by coordinating events for NGOs, such as the "Run For Hunger" hosted by his college during his undergraduate studies. He also brings the professional experience of Human Resource Management onto the table that sharpens his leadership abilities and enhances his efficiency in team collaboration, giving him an edge over his peers.

Magadi Nagaraj, Nayana

Nayana is currently pursuing a master's in Engineering Management at Northeastern University. She has over 2.5 years of experience with Amazon, where she worked as a Senior Associate in Global Solutions and Risk Compliance. Here, she managed a variety of programs aimed at improving quality and accuracy, hence increasing operating efficiency. Nayana has worked directly with global leaders to implement strategic solutions and worked across cross-functional teams to improve the productivity of the processes. Her leadership approach focuses on cooperation and constant development, resulting in effective outcomes in complicated, multinational projects.

Narendra Kumar, Vinay

Vinay is a graduate student pursuing his master's in Engineering Management at Northeastern University, with a bachelor's degree in Mechanical Engineering. Vinay is a fresher who came right out of college without any prior experience but he has interned as a quality engineer in a manufacturing plant. Vinay also interned at Automotive Axles Ltd., a large axle manufacturing company that supplies its products to the Indian Army, Volvo Trucks, where he gained experience in Axle manufacturing and machining. Vinay has a strong passion for learning agile methodologies and developing management skills to effectively solve problems.

Ravishankar, Abhijith Raj Urs

Abhijith Raj is currently a graduate student pursuing Engineering Management at Northeastern University. Prior to this, he gained valuable experience as a Full Stack Developer and Product Designer with AWS Solution Architect expertise, accumulating 3 years of experience in the Agile software lifecycle at a startup. With a background in Computer Science and Information Systems, Abhijith has successfully completed numerous projects. Additionally, he has experience leading a team and mentoring interns on front-end technologies and product development. Abhijith is highly proficient in building scalable distributed web applications for the financial services and retail domains. In his previous roles, he increased customer satisfaction by 25%, reduced incidents by 60%, and cut down 2 million calls per year for customer support, among other achievements.

Yogendrakumar, Pavan Thaloor

Pavan T Y is a first-year Engineering Management student with extensive experience in software engineering holding a bachelor's degree in Computer Science. Pavan previously worked as a Software Engineer for Capgemini, where he gained valuable experience on a variety of banking and finance projects. His responsibilities included creating and executing innovative software solutions while working closely with cross-functional teams to achieve project success. Pavan is always polishing his skills in project management, leadership, and teamwork, making him an invaluable asset in any collaborative workplace. His dedication to constant learning and problem solving strengthens his contributions to the field.

References

Mobile Payment Market Size - Share & Industry Report | 2027. (n.d.). Allied Market Research, from https://www.alliedmarketresearch.com/mobile-payments-market

Payment-trends. (n.d.). Retrieved June 4, 2024, from https://www.lightspeedhq.com/blog/new-payment-trends/

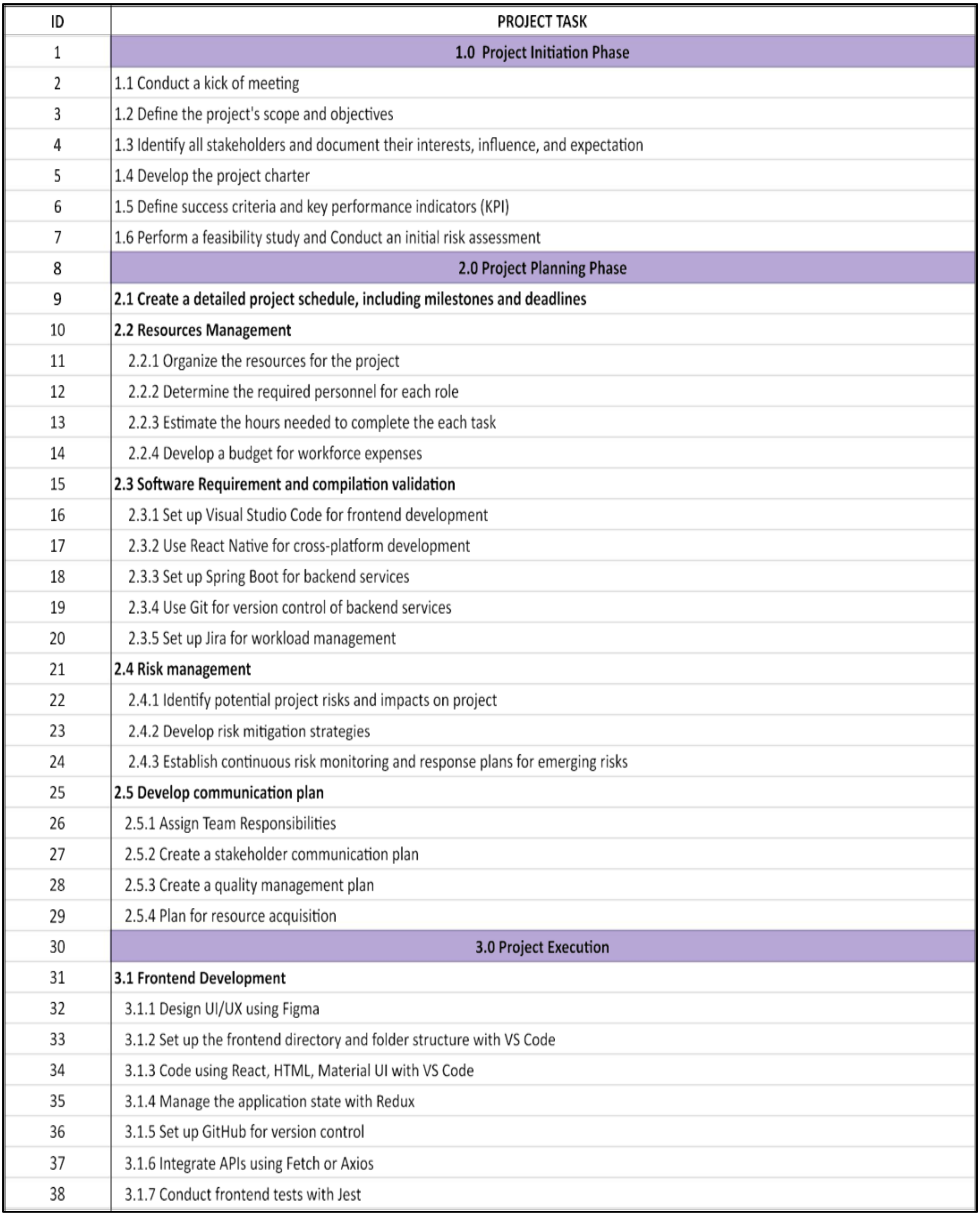
Hollington, J., & Bizzaco, M. (2024, April 14). *What is NFC? How it works and what you can do with it.* Digital Trends. Retrieved June 4, 2024, from *https://www.digitaltrends.com/mobile/what-is-nfc/*

Overview of BLS Wage Data by Area and Occupation: U.S. (2019, April 3). Bureau of Labor Statistics. Retrieved June 4, 2024, from *https://www.bls.gov/bls/blswage.htm*

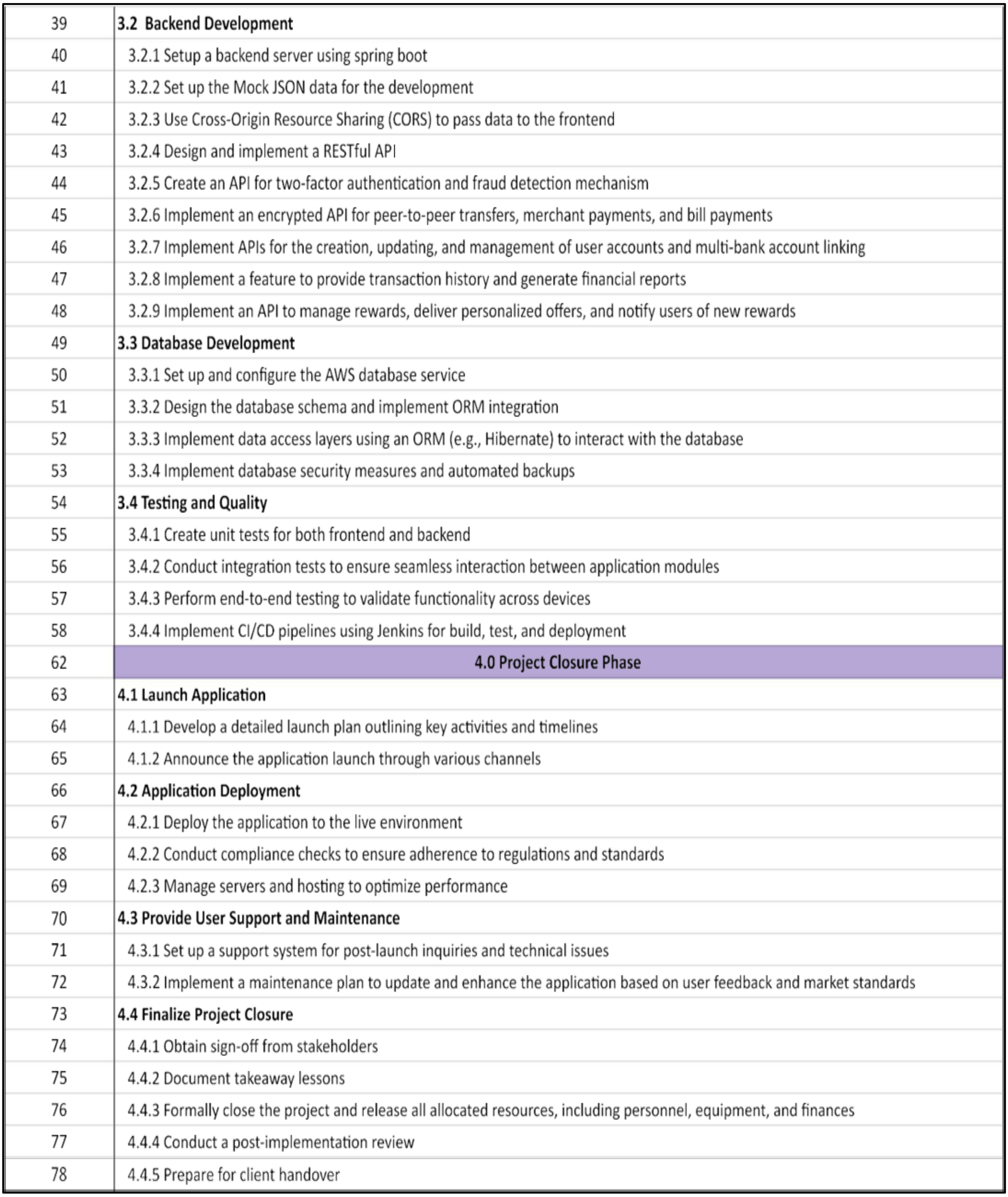
ieeexplore. (n.d.). Wikipedia. Retrieved June 4, 2024, from *https://ieeexplore.ieee.org/abstract/document/5234426*

Appendices

Appendix A: Work Breakdown Structure

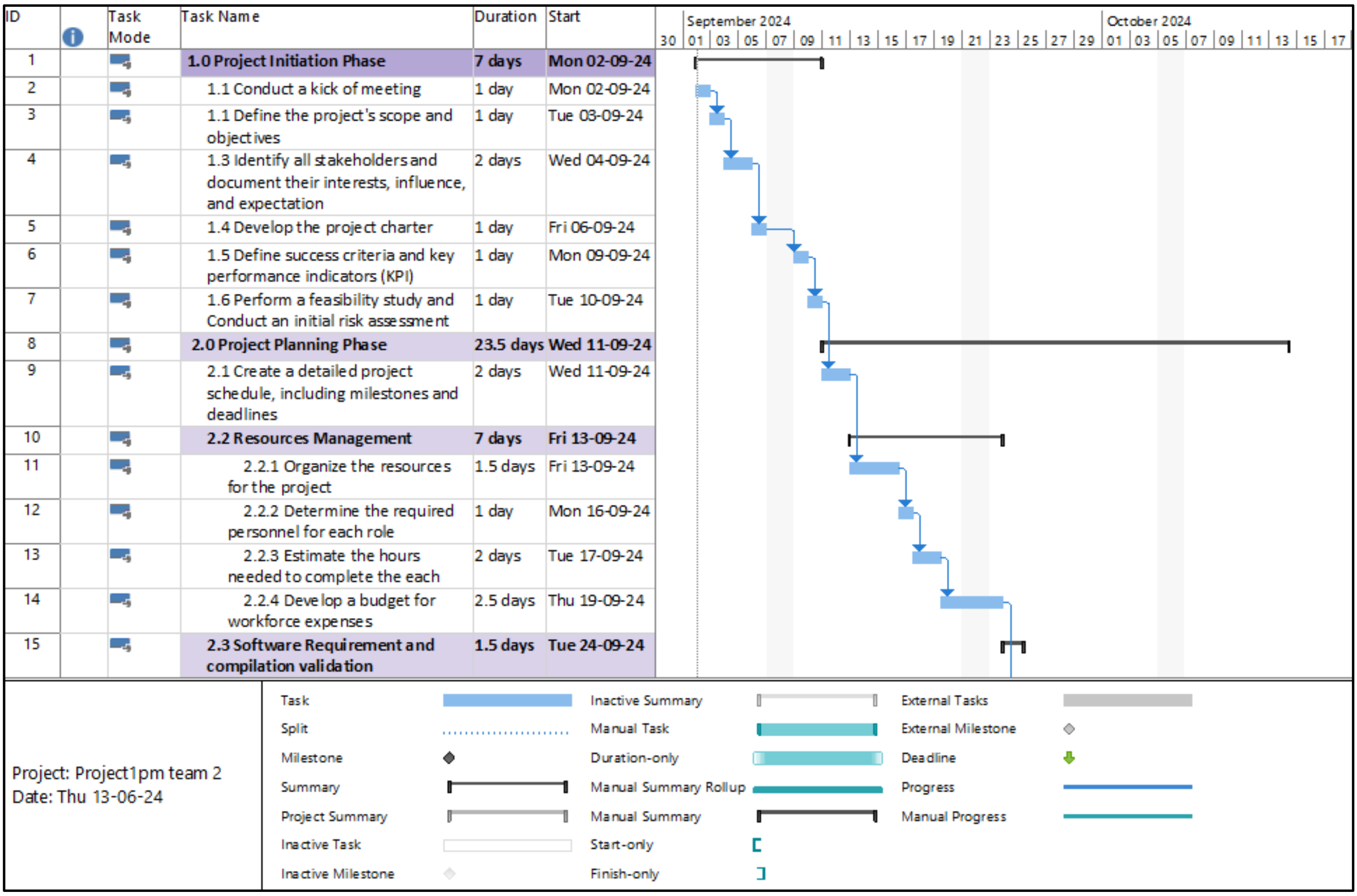


ID	PROJECT TASK
1	**1.0 Project Initiation Phase**
2	1.1 Conduct a kick of meeting
3	1.2 Define the project's scope and objectives
4	1.3 Identify all stakeholders and document their interests, influence, and expectation
5	1.4 Develop the project charter
6	1.5 Define success criteria and key performance indicators (KPI)
7	1.6 Perform a feasibility study and Conduct an initial risk assessment
8	**2.0 Project Planning Phase**
9	**2.1 Create a detailed project schedule, including milestones and deadlines**
10	**2.2 Resources Management**
11	2.2.1 Organize the resources for the project
12	2.2.2 Determine the required personnel for each role
13	2.2.3 Estimate the hours needed to complete the each task
14	2.2.4 Develop a budget for workforce expenses
15	**2.3 Software Requirement and compilation validation**
16	2.3.1 Set up Visual Studio Code for frontend development
17	2.3.2 Use React Native for cross-platform development
18	2.3.3 Set up Spring Boot for backend services
19	2.3.4 Use Git for version control of backend services
20	2.3.5 Set up Jira for workload management
21	**2.4 Risk management**
22	2.4.1 Identify potential project risks and impacts on project
23	2.4.2 Develop risk mitigation strategies
24	2.4.3 Establish continuous risk monitoring and response plans for emerging risks
25	**2.5 Develop communication plan**
26	2.5.1 Assign Team Responsibilities
27	2.5.2 Create a stakeholder communication plan
28	2.5.3 Create a quality management plan
29	2.5.4 Plan for resource acquisition
30	**3.0 Project Execution**
31	**3.1 Frontend Development**
32	3.1.1 Design UI/UX using Figma
33	3.1.2 Set up the frontend directory and folder structure with VS Code
34	3.1.3 Code using React, HTML, Material UI with VS Code
35	3.1.4 Manage the application state with Redux
36	3.1.5 Set up GitHub for version control
37	3.1.6 Integrate APIs using Fetch or Axios
38	3.1.7 Conduct frontend tests with Jest



39	**3.2 Backend Development**
40	3.2.1 Setup a backend server using spring boot
41	3.2.2 Set up the Mock JSON data for the development
42	3.2.3 Use Cross-Origin Resource Sharing (CORS) to pass data to the frontend
43	3.2.4 Design and implement a RESTful API
44	3.2.5 Create an API for two-factor authentication and fraud detection mechanism
45	3.2.6 Implement an encrypted API for peer-to-peer transfers, merchant payments, and bill payments
46	3.2.7 Implement APIs for the creation, updating, and management of user accounts and multi-bank account linking
47	3.2.8 Implement a feature to provide transaction history and generate financial reports
48	3.2.9 Implement an API to manage rewards, deliver personalized offers, and notify users of new rewards
49	**3.3 Database Development**
50	3.3.1 Set up and configure the AWS database service
51	3.3.2 Design the database schema and implement ORM integration
52	3.3.3 Implement data access layers using an ORM (e.g., Hibernate) to interact with the database
53	3.3.4 Implement database security measures and automated backups
54	**3.4 Testing and Quality**
55	3.4.1 Create unit tests for both frontend and backend
56	3.4.2 Conduct integration tests to ensure seamless interaction between application modules
57	3.4.3 Perform end-to-end testing to validate functionality across devices
58	3.4.4 Implement CI/CD pipelines using Jenkins for build, test, and deployment
62	**4.0 Project Closure Phase**
63	**4.1 Launch Application**
64	4.1.1 Develop a detailed launch plan outlining key activities and timelines
65	4.1.2 Announce the application launch through various channels
66	**4.2 Application Deployment**
67	4.2.1 Deploy the application to the live environment
68	4.2.2 Conduct compliance checks to ensure adherence to regulations and standards
69	4.2.3 Manage servers and hosting to optimize performance
70	**4.3 Provide User Support and Maintenance**
71	4.3.1 Set up a support system for post-launch inquiries and technical issues
72	4.3.2 Implement a maintenance plan to update and enhance the application based on user feedback and market standards
73	**4.4 Finalize Project Closure**
74	4.4.1 Obtain sign-off from stakeholders
75	4.4.2 Document takeaway lessons
76	4.4.3 Formally close the project and release all allocated resources, including personnel, equipment, and finances
77	4.4.4 Conduct a post-implementation review
78	4.4.5 Prepare for client handover

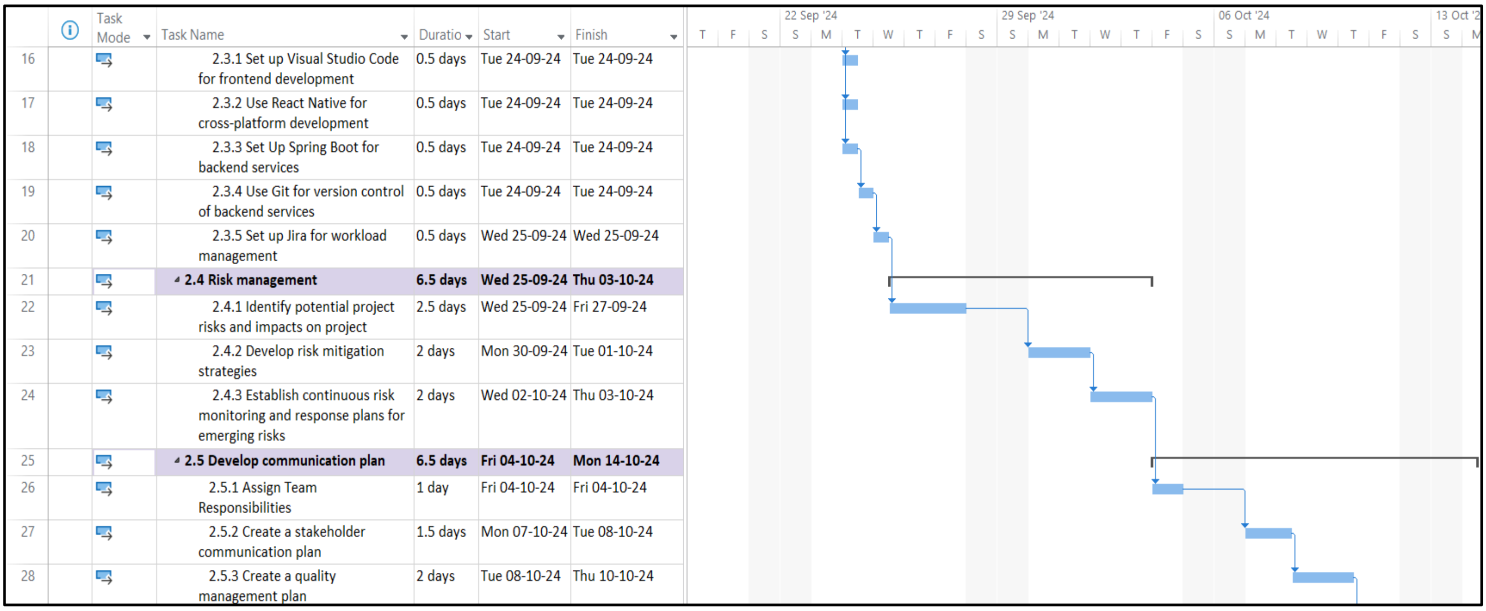
Appendix B: Project Schedule



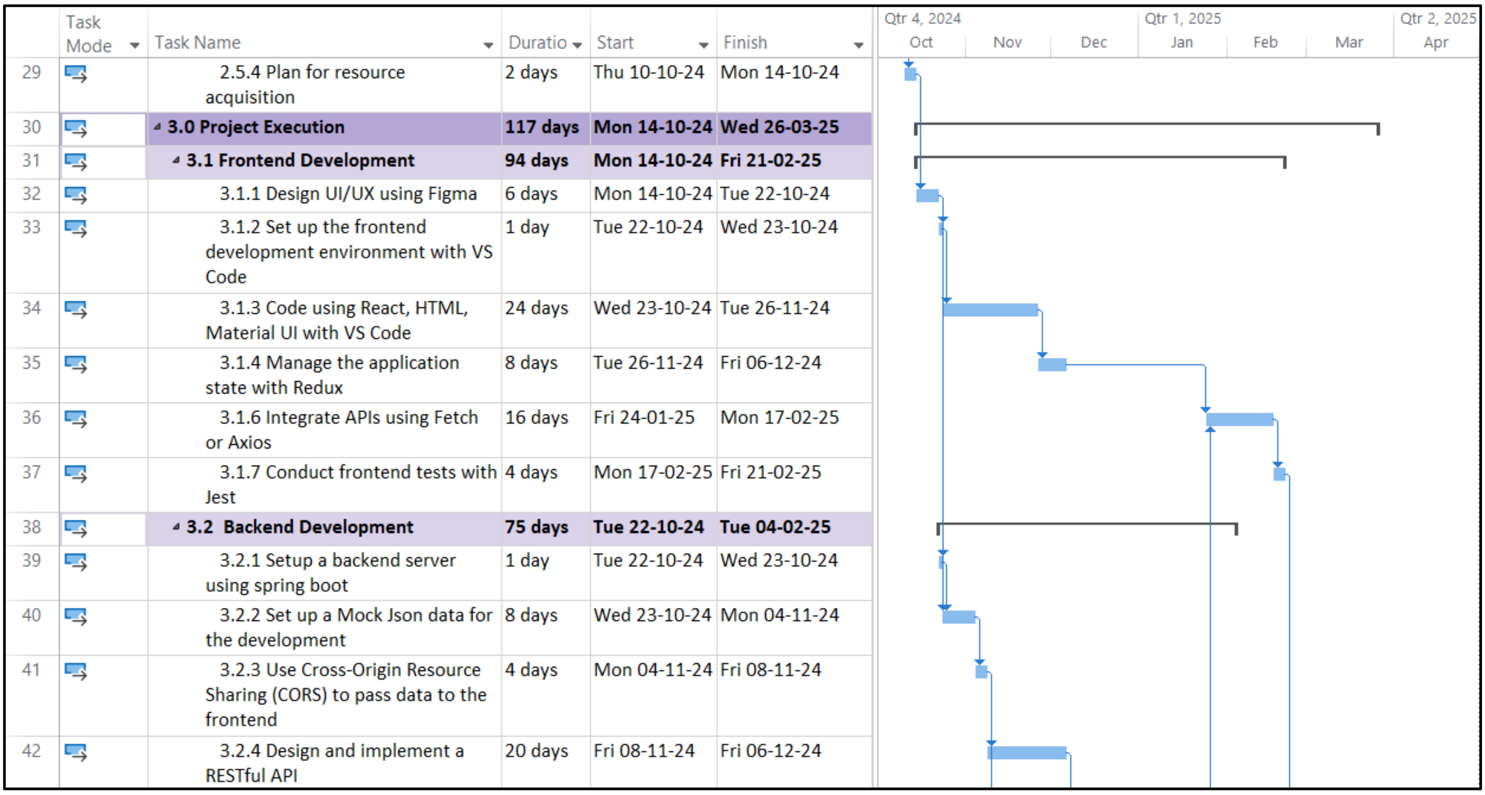
ID	Task Mode	Task Name	Duration	Start
1		**1.0 Project Initiation Phase**	**7 days**	**Mon 02-09-24**
2		1.1 Conduct a kick of meeting	1 day	Mon 02-09-24
3		1.1 Define the project's scope and objectives	1 day	Tue 03-09-24
4		1.3 Identify all stakeholders and document their interests, influence, and expectation	2 days	Wed 04-09-24
5		1.4 Develop the project charter	1 day	Fri 06-09-24
6		1.5 Define success criteria and key performance indicators (KPI)	1 day	Mon 09-09-24
7		1.6 Perform a feasibility study and Conduct an initial risk assessment	1 day	Tue 10-09-24
8		**2.0 Project Planning Phase**	**23.5 days**	**Wed 11-09-24**
9		2.1 Create a detailed project schedule, including milestones and deadlines	2 days	Wed 11-09-24
10		**2.2 Resources Management**	**7 days**	**Fri 13-09-24**
11		2.2.1 Organize the resources for the project	1.5 days	Fri 13-09-24
12		2.2.2 Determine the required personnel for each role	1 day	Mon 16-09-24
13		2.2.3 Estimate the hours needed to complete the each	2 days	Tue 17-09-24
14		2.2.4 Develop a budget for workforce expenses	2.5 days	Thu 19-09-24
15		**2.3 Software Requirement and compilation validation**	**1.5 days**	**Tue 24-09-24**

Project: Project1pm team 2
Date: Thu 13-06-24

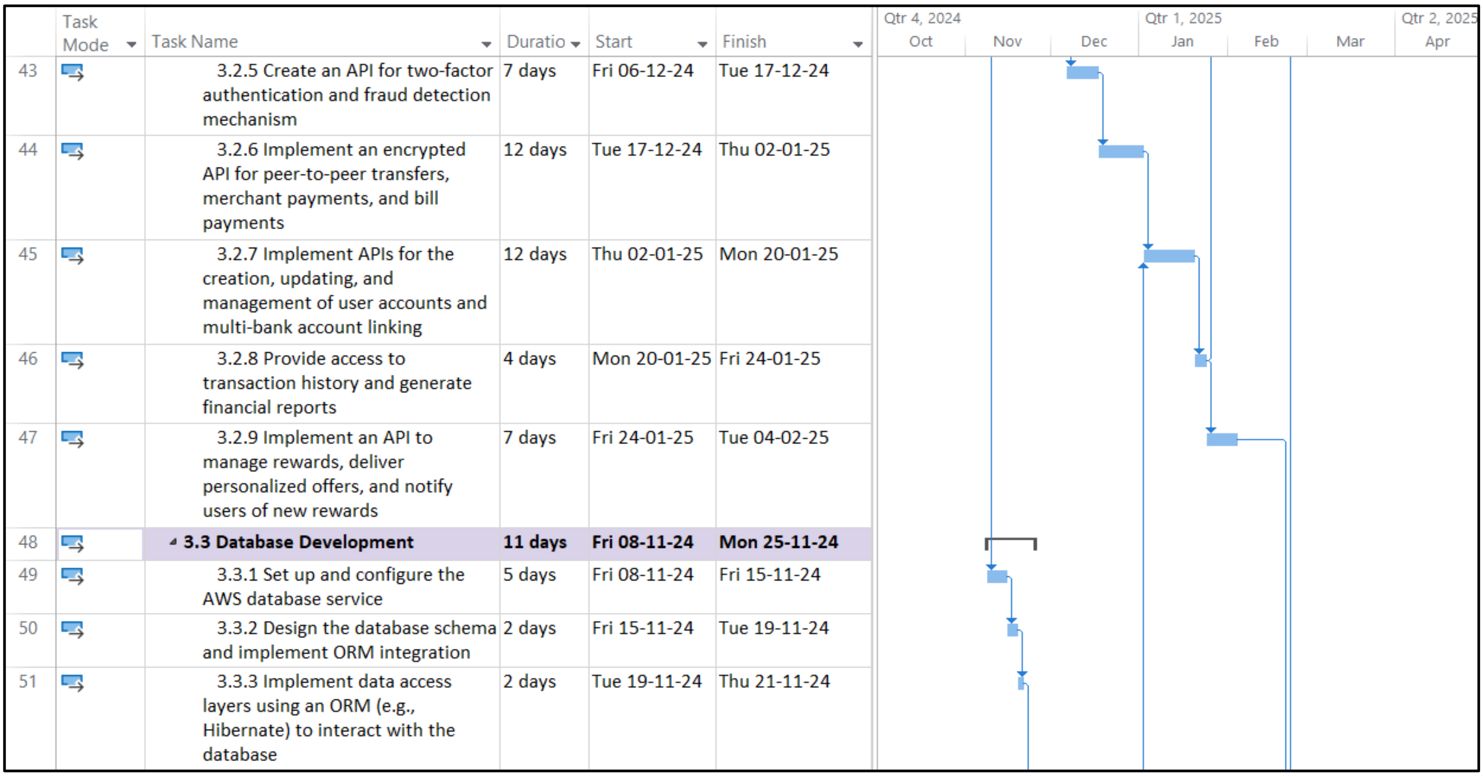
Task		Inactive Summary		External Tasks	
Split		Manual Task		External Milestone	◆
Milestone	◆	Duration-only		Deadline	⬇
Summary		Manual Summary Rollup		Progress	
Project Summary		Manual Summary		Manual Progress	
Inactive Task		Start-only	[		
Inactive Milestone	◆	Finish-only	]		



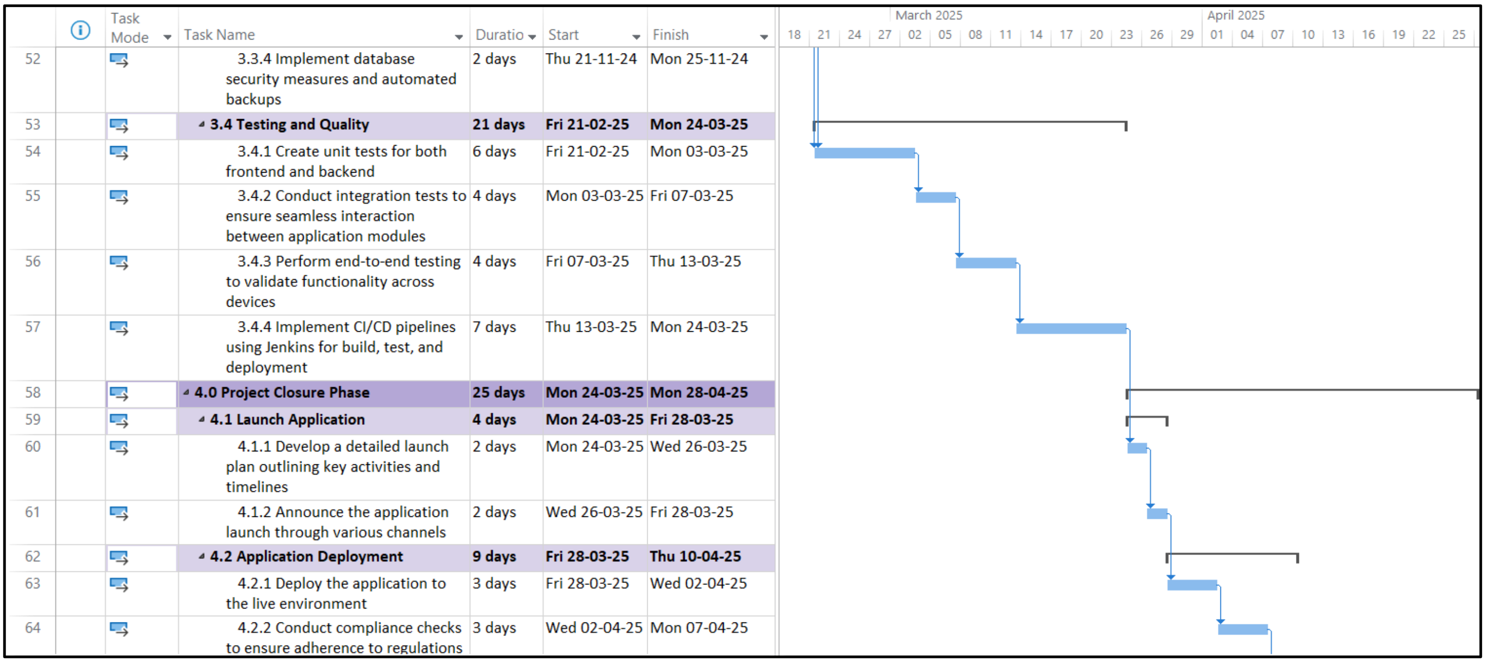
	Task Mode	Task Name	Duratio	Start	Finish
16		2.3.1 Set up Visual Studio Code for frontend development	0.5 days	Tue 24-09-24	Tue 24-09-24
17		2.3.2 Use React Native for cross-platform development	0.5 days	Tue 24-09-24	Tue 24-09-24
18		2.3.3 Set Up Spring Boot for backend services	0.5 days	Tue 24-09-24	Tue 24-09-24
19		2.3.4 Use Git for version control of backend services	0.5 days	Tue 24-09-24	Tue 24-09-24
20		2.3.5 Set up Jira for workload management	0.5 days	Wed 25-09-24	Wed 25-09-24
21		**2.4 Risk management**	**6.5 days**	**Wed 25-09-24**	**Thu 03-10-24**
22		2.4.1 Identify potential project risks and impacts on project	2.5 days	Wed 25-09-24	Fri 27-09-24
23		2.4.2 Develop risk mitigation strategies	2 days	Mon 30-09-24	Tue 01-10-24
24		2.4.3 Establish continuous risk monitoring and response plans for emerging risks	2 days	Wed 02-10-24	Thu 03-10-24
25		**2.5 Develop communication plan**	**6.5 days**	**Fri 04-10-24**	**Mon 14-10-24**
26		2.5.1 Assign Team Responsibilities	1 day	Fri 04-10-24	Fri 04-10-24
27		2.5.2 Create a stakeholder communication plan	1.5 days	Mon 07-10-24	Tue 08-10-24
28		2.5.3 Create a quality management plan	2 days	Tue 08-10-24	Thu 10-10-24



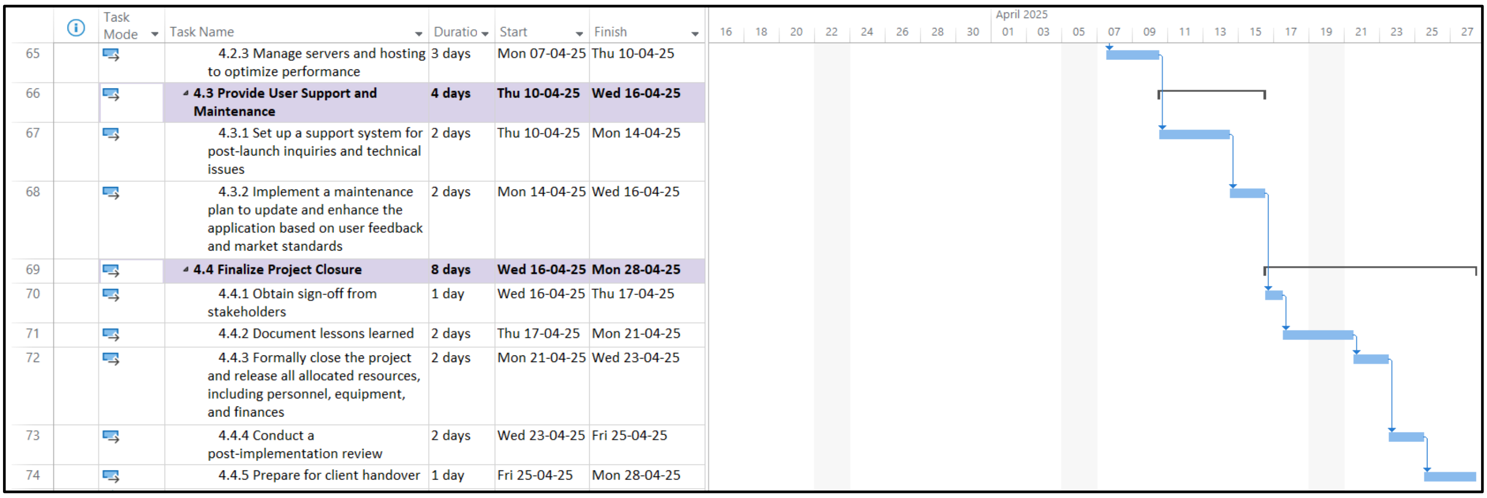
	Task Mode	Task Name	Duratio	Start	Finish
29		2.5.4 Plan for resource acquisition	2 days	Thu 10-10-24	Mon 14-10-24
30		**3.0 Project Execution**	**117 days**	**Mon 14-10-24**	**Wed 26-03-25**
31		**3.1 Frontend Development**	**94 days**	**Mon 14-10-24**	**Fri 21-02-25**
32		3.1.1 Design UI/UX using Figma	6 days	Mon 14-10-24	Tue 22-10-24
33		3.1.2 Set up the frontend development environment with VS Code	1 day	Tue 22-10-24	Wed 23-10-24
34		3.1.3 Code using React, HTML, Material UI with VS Code	24 days	Wed 23-10-24	Tue 26-11-24
35		3.1.4 Manage the application state with Redux	8 days	Tue 26-11-24	Fri 06-12-24
36		3.1.6 Integrate APIs using Fetch or Axios	16 days	Fri 24-01-25	Mon 17-02-25
37		3.1.7 Conduct frontend tests with Jest	4 days	Mon 17-02-25	Fri 21-02-25
38		**3.2 Backend Development**	**75 days**	**Tue 22-10-24**	**Tue 04-02-25**
39		3.2.1 Setup a backend server using spring boot	1 day	Tue 22-10-24	Wed 23-10-24
40		3.2.2 Set up a Mock Json data for the development	8 days	Wed 23-10-24	Mon 04-11-24
41		3.2.3 Use Cross-Origin Resource Sharing (CORS) to pass data to the frontend	4 days	Mon 04-11-24	Fri 08-11-24
42		3.2.4 Design and implement a RESTful API	20 days	Fri 08-11-24	Fri 06-12-24



	Task Mode	Task Name	Duratio	Start	Finish
43		3.2.5 Create an API for two-factor authentication and fraud detection mechanism	7 days	Fri 06-12-24	Tue 17-12-24
44		3.2.6 Implement an encrypted API for peer-to-peer transfers, merchant payments, and bill payments	12 days	Tue 17-12-24	Thu 02-01-25
45		3.2.7 Implement APIs for the creation, updating, and management of user accounts and multi-bank account linking	12 days	Thu 02-01-25	Mon 20-01-25
46		3.2.8 Provide access to transaction history and generate financial reports	4 days	Mon 20-01-25	Fri 24-01-25
47		3.2.9 Implement an API to manage rewards, deliver personalized offers, and notify users of new rewards	7 days	Fri 24-01-25	Tue 04-02-25
48		**3.3 Database Development**	**11 days**	**Fri 08-11-24**	**Mon 25-11-24**
49		3.3.1 Set up and configure the AWS database service	5 days	Fri 08-11-24	Fri 15-11-24
50		3.3.2 Design the database schema and implement ORM integration	2 days	Fri 15-11-24	Tue 19-11-24
51		3.3.3 Implement data access layers using an ORM (e.g., Hibernate) to interact with the database	2 days	Tue 19-11-24	Thu 21-11-24

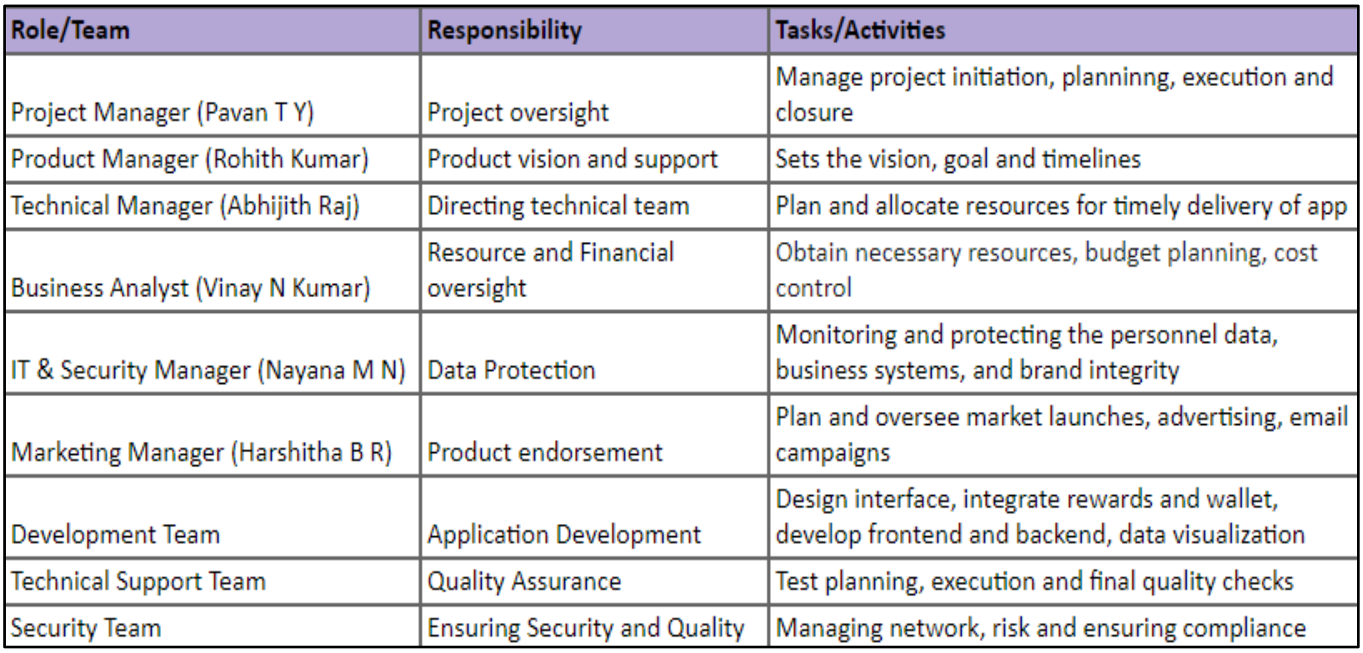


	ⓘ	Task Mode	Task Name	Duratio	Start	Finish
52			3.3.4 Implement database security measures and automated backups	2 days	Thu 21-11-24	Mon 25-11-24
53			**4 3.4 Testing and Quality**	**21 days**	**Fri 21-02-25**	**Mon 24-03-25**
54			3.4.1 Create unit tests for both frontend and backend	6 days	Fri 21-02-25	Mon 03-03-25
55			3.4.2 Conduct integration tests to ensure seamless interaction between application modules	4 days	Mon 03-03-25	Fri 07-03-25
56			3.4.3 Perform end-to-end testing to validate functionality across devices	4 days	Fri 07-03-25	Thu 13-03-25
57			3.4.4 Implement CI/CD pipelines using Jenkins for build, test, and deployment	7 days	Thu 13-03-25	Mon 24-03-25
58			**4 4.0 Project Closure Phase**	**25 days**	**Mon 24-03-25**	**Mon 28-04-25**
59			**4 4.1 Launch Application**	**4 days**	**Mon 24-03-25**	**Fri 28-03-25**
60			4.1.1 Develop a detailed launch plan outlining key activities and timelines	2 days	Mon 24-03-25	Wed 26-03-25
61			4.1.2 Announce the application launch through various channels	2 days	Wed 26-03-25	Fri 28-03-25
62			**4 4.2 Application Deployment**	**9 days**	**Fri 28-03-25**	**Thu 10-04-25**
63			4.2.1 Deploy the application to the live environment	3 days	Fri 28-03-25	Wed 02-04-25
64			4.2.2 Conduct compliance checks to ensure adherence to regulations	3 days	Wed 02-04-25	Mon 07-04-25



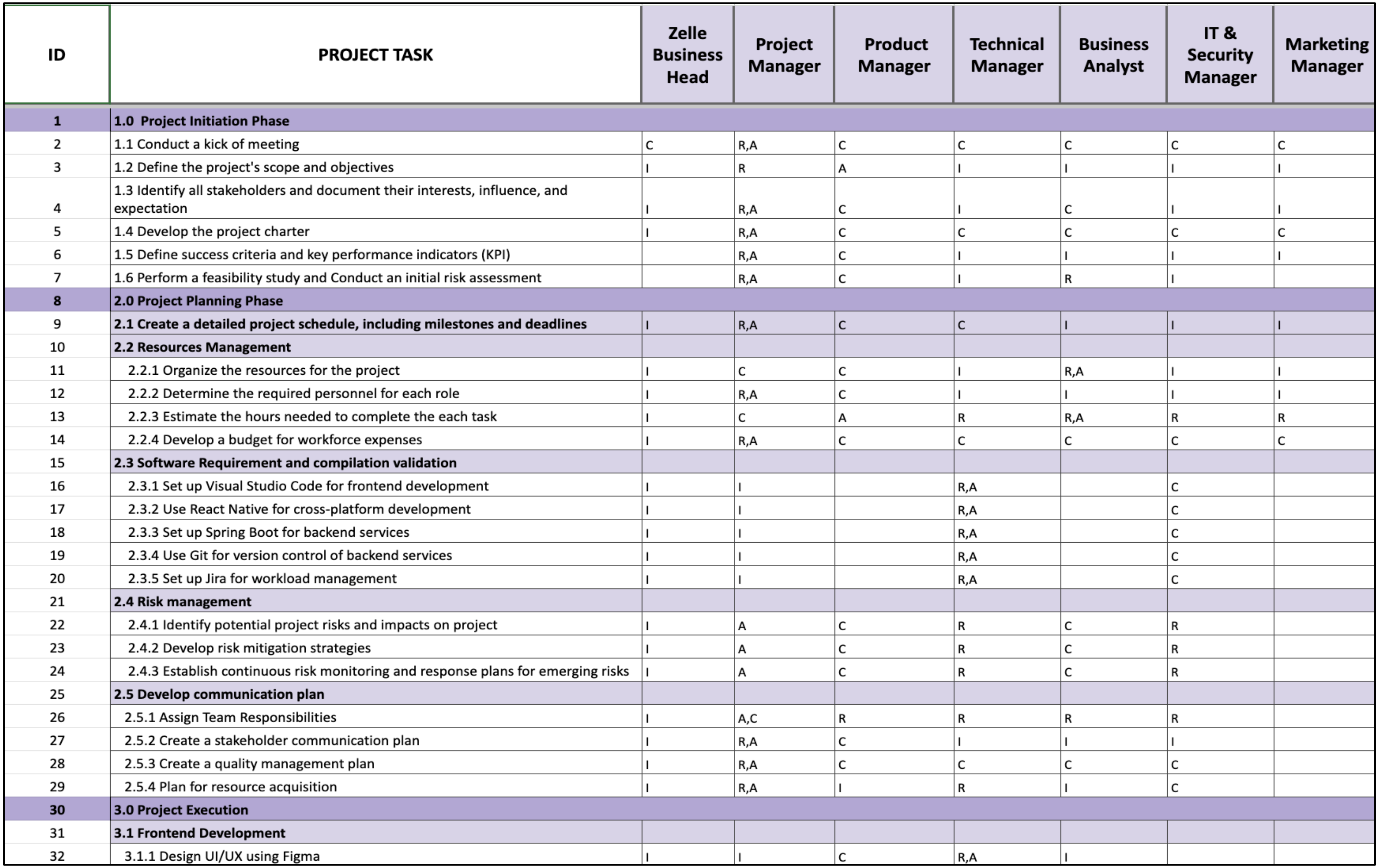
	ⓘ	Task Mode	Task Name	Duratio	Start	Finish
65			4.2.3 Manage servers and hosting to optimize performance	3 days	Mon 07-04-25	Thu 10-04-25
66			**4 4.3 Provide User Support and Maintenance**	**4 days**	**Thu 10-04-25**	**Wed 16-04-25**
67			4.3.1 Set up a support system for post-launch inquiries and technical issues	2 days	Thu 10-04-25	Mon 14-04-25
68			4.3.2 Implement a maintenance plan to update and enhance the application based on user feedback and market standards	2 days	Mon 14-04-25	Wed 16-04-25
69			**4 4.4 Finalize Project Closure**	**8 days**	**Wed 16-04-25**	**Mon 28-04-25**
70			4.4.1 Obtain sign-off from stakeholders	1 day	Wed 16-04-25	Thu 17-04-25
71			4.4.2 Document lessons learned	2 days	Thu 17-04-25	Mon 21-04-25
72			4.4.3 Formally close the project and release all allocated resources, including personnel, equipment, and finances	2 days	Mon 21-04-25	Wed 23-04-25
73			4.4.4 Conduct a post-implementation review	2 days	Wed 23-04-25	Fri 25-04-25
74			4.4.5 Prepare for client handover	1 day	Fri 25-04-25	Mon 28-04-25

Appendix C: Responsibility Chart



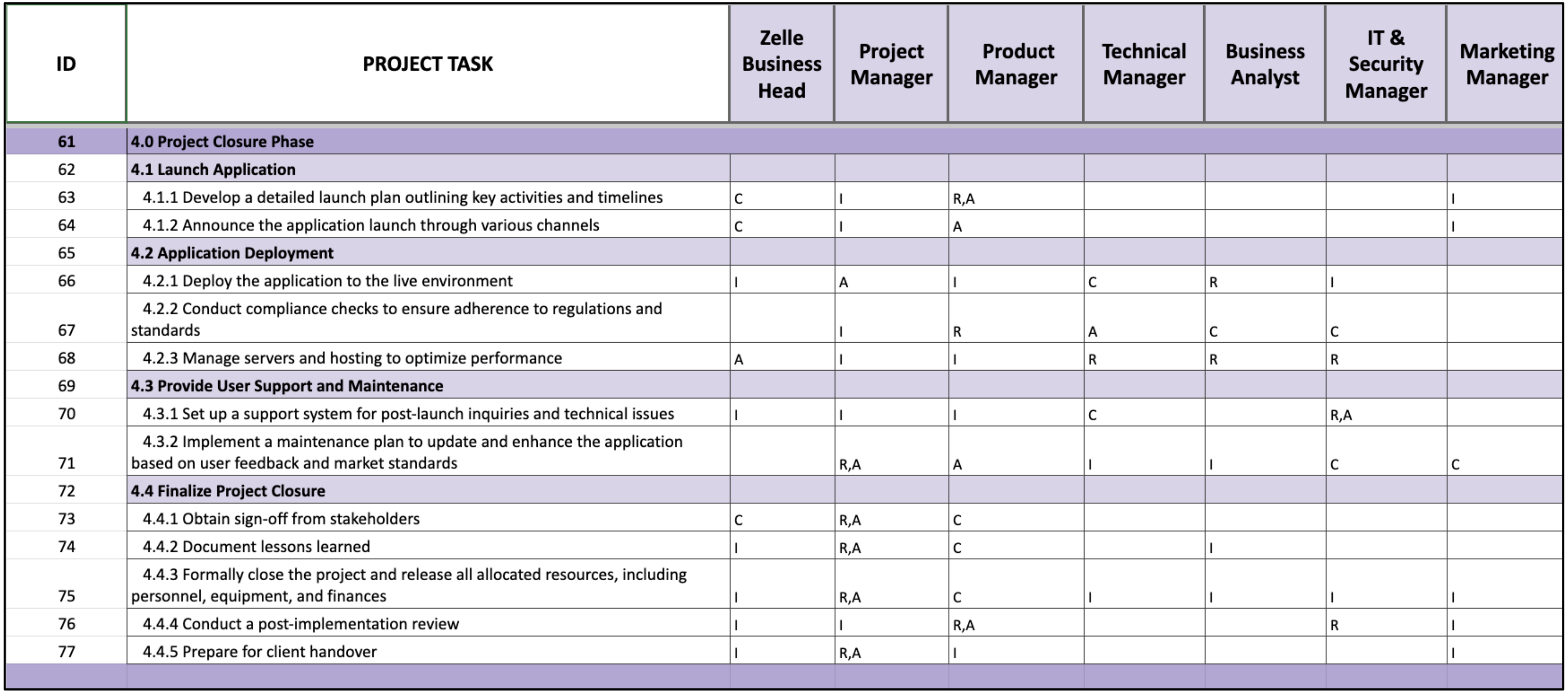
Role/Team	Responsibility	Tasks/Activities
Project Manager (Pavan T Y)	Project oversight	Manage project initiation, planninng, execution and closure
Product Manager (Rohith Kumar)	Product vision and support	Sets the vision, goal and timelines
Technical Manager (Abhijith Raj)	Directing technical team	Plan and allocate resources for timely delivery of app
Business Analyst (Vinay N Kumar)	Resource and Financial oversight	Obtain necessary resources, budget planning, cost control
IT & Security Manager (Nayana M N)	Data Protection	Monitoring and protecting the personnel data, business systems, and brand integrity
Marketing Manager (Harshitha B R)	Product endorsement	Plan and oversee market launches, advertising, email campaigns
Development Team	Application Development	Design interface, integrate rewards and wallet, develop frontend and backend, data visualization
Technical Support Team	Quality Assurance	Test planning, execution and final quality checks
Security Team	Ensuring Security and Quality	Managing network, risk and ensuring compliance

Appendix D: RACI Matrix



ID	PROJECT TASK	Zelle Business Head	Project Manager	Product Manager	Technical Manager	Business Analyst	IT & Security Manager	Marketing Manager
1	**1.0 Project Initiation Phase**							
2	1.1 Conduct a kick of meeting	C	R,A	C	C	C	C	C
3	1.2 Define the project's scope and objectives	I	R	A	I	I	I	I
4	1.3 Identify all stakeholders and document their interests, influence, and expectation	I	R,A	C	I	C	I	I
5	1.4 Develop the project charter	I	R,A	C	C	C	C	C
6	1.5 Define success criteria and key performance indicators (KPI)		R,A	C	I	I	I	I
7	1.6 Perform a feasibility study and Conduct an initial risk assessment		R,A	C	I	R	I	
8	**2.0 Project Planning Phase**							
9	**2.1 Create a detailed project schedule, including milestones and deadlines**	I	R,A	C	C	I	I	I
10	**2.2 Resources Management**							
11	2.2.1 Organize the resources for the project	I	C	C	I	R,A	I	I
12	2.2.2 Determine the required personnel for each role	I	R,A	C	I	I	I	I
13	2.2.3 Estimate the hours needed to complete the each task	I	C	A	R	R,A	R	R
14	2.2.4 Develop a budget for workforce expenses	I	R,A	C	C	C	C	C
15	**2.3 Software Requirement and compilation validation**							
16	2.3.1 Set up Visual Studio Code for frontend development	I	I		R,A		C	
17	2.3.2 Use React Native for cross-platform development	I	I		R,A		C	
18	2.3.3 Set up Spring Boot for backend services	I	I		R,A		C	
19	2.3.4 Use Git for version control of backend services	I	I		R,A		C	
20	2.3.5 Set up Jira for workload management	I	I		R,A		C	
21	**2.4 Risk management**							
22	2.4.1 Identify potential project risks and impacts on project	I	A	C	R	C	R	
23	2.4.2 Develop risk mitigation strategies	I	A	C	R	C	R	
24	2.4.3 Establish continuous risk monitoring and response plans for emerging risks	I	A	C	R	C	R	
25	**2.5 Develop communication plan**							
26	2.5.1 Assign Team Responsibilities	I	A,C	R	R	R	R	
27	2.5.2 Create a stakeholder communication plan	I	R,A	C	I	I	I	
28	2.5.3 Create a quality management plan	I	R,A	C	C	C	C	
29	2.5.4 Plan for resource acquisition	I	R,A	I	R	I	C	
30	**3.0 Project Execution**							
31	**3.1 Frontend Development**							
32	3.1.1 Design UI/UX using Figma	I	I	C	R,A	I		

ID	PROJECT TASK	Zelle Business Head	Project Manager	Product Manager	Technical Manager	Business Analyst	IT & Security Manager	Marketing Manager
33	3.1.2 Set up the frontend directory and folder structure with VS Code	I	I		R,A			
34	3.1.3 Code using React, HTML, Material UI with VS Code	I	I		R,A			
35	3.1.4 Manage the application state with Redux	I	I	C	R,A			
36	3.1.6 Integrate APIs using Fetch or Axios	I	I		R,A			
37	3.1.7 Conduct frontend tests with Jest	I	I		R,A			
38	**3.2 Backend Development**							
39	3.2.1 Setup a backend server using spring boot	I	I		R,A		C	
40	3.2.2 Set up the Mock JSON data for the development	I	I		R,A		C	
41	3.2.3 Use Cross-Origin Resource Sharing (CORS) to pass data to the frontend	I	I		R,A		C	
42	3.2.4 Design and implement a RESTful API	I	I	C	R,A		C	
43	3.2.5 Create an API for two-factor authentication and fraud detection mechanism	I	I		R,A		C	
44	3.2.6 Implement an encrypted API for peer-to-peer transfers, merchant payments, and bill payments	I	I		R,A		C	
45	3.2.7 Implement APIs for the creation, updating, and management of user accounts and multi-bank account linking	I	I	C	R,A		C	I
46	3.2.8 Implement a feature to provide transaction history and generate financial reports	I	I	C	R,A		C	I
47	3.2.9 Implement an API to manage rewards, deliver personalized offers, and notify users of new rewards	I	I	C	R,A		C	I
48	**3.3 Database Development**							
49	3.3.1 Set up and configure the AWS database service	I	I		R,A		C	
50	3.3.2 Design the database schema and implement ORM integration	I	I		R,A		C	
51	3.3.3 Implement data access layers using an ORM (e.g., Hibernate) to interact with the database	I	I		R,A		C	
52	3.3.4 Implement database security measures and automated backups	I	I		R,A		R,C	
53	**3.4 Testing and Quality**							
54	3.4.1 Create unit tests for both frontend and backend	I	I	C	R		R,A	
55	3.4.2 Conduct integration tests to ensure seamless interaction between application modules	I	I	C	R		R,A	
56	3.4.3 Perform end-to-end testing to validate functionality across devices	I	I	C	R		R,A	
57	3.4.4 Implement CI/CD pipelines using Jenkins for build, test, and deployment	I	I	C	R		R,A	
58	**3.5 Performance Monitoring**							
59	3.5.1 Establish metrics to monitor project performance and progress		A	C	R	R	C	I
60	3.5.2 Conduct regular performance reviews and status meetings		A	C	R	R	C	I



ID	PROJECT TASK	Zelle Business Head	Project Manager	Product Manager	Technical Manager	Business Analyst	IT & Security Manager	Marketing Manager
61	**4.0 Project Closure Phase**							
62	**4.1 Launch Application**							
63	4.1.1 Develop a detailed launch plan outlining key activities and timelines	C	I	R,A				I
64	4.1.2 Announce the application launch through various channels	C	I	A				I
65	**4.2 Application Deployment**							
66	4.2.1 Deploy the application to the live environment	I	A	I	C	R	I	
67	4.2.2 Conduct compliance checks to ensure adherence to regulations and standards		I	R	A	C	C	
68	4.2.3 Manage servers and hosting to optimize performance	A	I	I	R	R	R	
69	**4.3 Provide User Support and Maintenance**							
70	4.3.1 Set up a support system for post-launch inquiries and technical issues	I	I	I	C		R,A	
71	4.3.2 Implement a maintenance plan to update and enhance the application based on user feedback and market standards		R,A	A	I	I	C	C
72	**4.4 Finalize Project Closure**							
73	4.4.1 Obtain sign-off from stakeholders	C	R,A	C				
74	4.4.2 Document lessons learned	I	R,A	C		I		
75	4.4.3 Formally close the project and release all allocated resources, including personnel, equipment, and finances	I	R,A	C	I	I	I	I
76	4.4.4 Conduct a post-implementation review	I	I	R,A			R	I
77	4.4.5 Prepare for client handover	I	R,A	I				I

Assumptions

IT/Project/Product Manager (R/A): When the Manager is designated as both Accountable (A) and Responsible (R) in the RACI matrix, it indicates that the Manager is Accountable for the task. the actual execution is typically carried out by team members such as developers who are considered Responsible (R) for performing the work under the Manager's accountability.

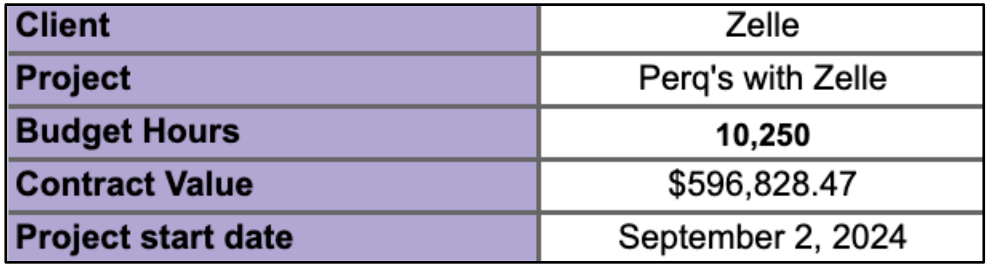
Appendix E: Budget Justification

Project Name: Perq's with Zelle Estimated Start time : September 2nd 2024								Total Cost		$596,828.47
ID	PROJECT TASK	Resources								Task Total Cost
		Labor				Equipment			Miscellaneous	
		No. of People	Working Hours	Hourly Rate	Estimated Cost	Quantity	Unit Price	Estimated Cost		
1	1.0 Project Initiation Phase									
2	1.1 Conduct a kick of meeting	12	6.00	$57.00	$4,104.00	9.00	$800.00	$7,200.00		$11,304.00
3	1.2 Define the project's scope and objectives	4	6.00	$57.00	$1,368.00					$1,368.00
4	1.3 Identify all stakeholders and document their interests, influence, and expectation	4	16.00	$57.00	$3,648.00					$3,648.00
5	1.4 Develop the project charter	4	8.00	$57.00	$1,824.00					$1,824.00
6	1.5 Define success criteria and key performance indicators (KPI)	4	8.00	$57.00	$1,824.00					$1,824.00
7	1.6 Perform a feasibility study and Conduct an initial risk assessment	4	8.00	$61.00	$1,952.00					$1,952.00
									Phase total cost:	$21,920.00
8	2.0 Project Planning Phase									
9	2.1 Create a detailed project schedule, including milestones and deadlines	6	18.00	$57.00	$6,156.00					$6,156.00
10	2.2 Resources Management									
11	2.2.1 Organize the resources for the project	3	12.00	$65.00	$2,340.00					$2,340.00
12	2.2.2 Determine the required personnel for each role	3	8.00	$57.00	$1,368.00					$1,368.00
13	2.2.3 Estimate the hours needed to complete the each task	5	16.00	$66.00	$5,280.00					$5,280.00
14	2.2.4 Develop a budget for workforce expenses	6	20.00	$57.00	$6,840.00					$6,840.00
15	2.3 Software Requirement and compilation validation									
16	2.3.1 Set up Visual Studio Code for frontend development	2	4.00	$72.00	$576.00					$576.00
17	2.3.2 Use React Native for cross-platform development	3	3.00	$72.00	$648.00	2.00	$100.00	$200.00		$848.00
18	2.3.3 Set up Spring Boot for backend services	2	4.00	$72.00	$576.00					$576.00
19	2.3.4 Use Github and Git for version control of backend services	2	4.00	$72.00	$576.00	8.00	$21.00	$168.00		$744.00
20	2.3.5 Set up Jira for workload management	4	3.00	$72.00	$864.00	8.00	$12.48	$99.84		$963.84
21	2.4 Risk management									
22	2.4.1 Identify potential project risks and impacts on project	4	18.00	$72.00	$5,184.00					$5,184.00
23	2.4.2 Develop risk mitigation strategies	4	16.00	$72.00	$4,608.00					$4,608.00
24	2.4.3 Establish continuous risk monitoring and response plans for emerging risks	3	16.00	$72.00	$3,456.00					$3,456.00
25	2.5 Develop communication plan									
26	2.5.1 Assign Team Responsibilities	3	6.00	$70.00	$1,260.00					$1,260.00
27	2.5.2 Create a stakeholder communication plan	4	12.00	$57.00	$2,736.00					$2,736.00
28	2.5.3 Create a quality management plan	4	16.00	$57.00	$3,648.00					$3,648.00
29	2.5.4 Plan for resource acquisition	7	15.00	$57.00	$5,985.00					$5,985.00
									Phase total cost:	$52,568.84

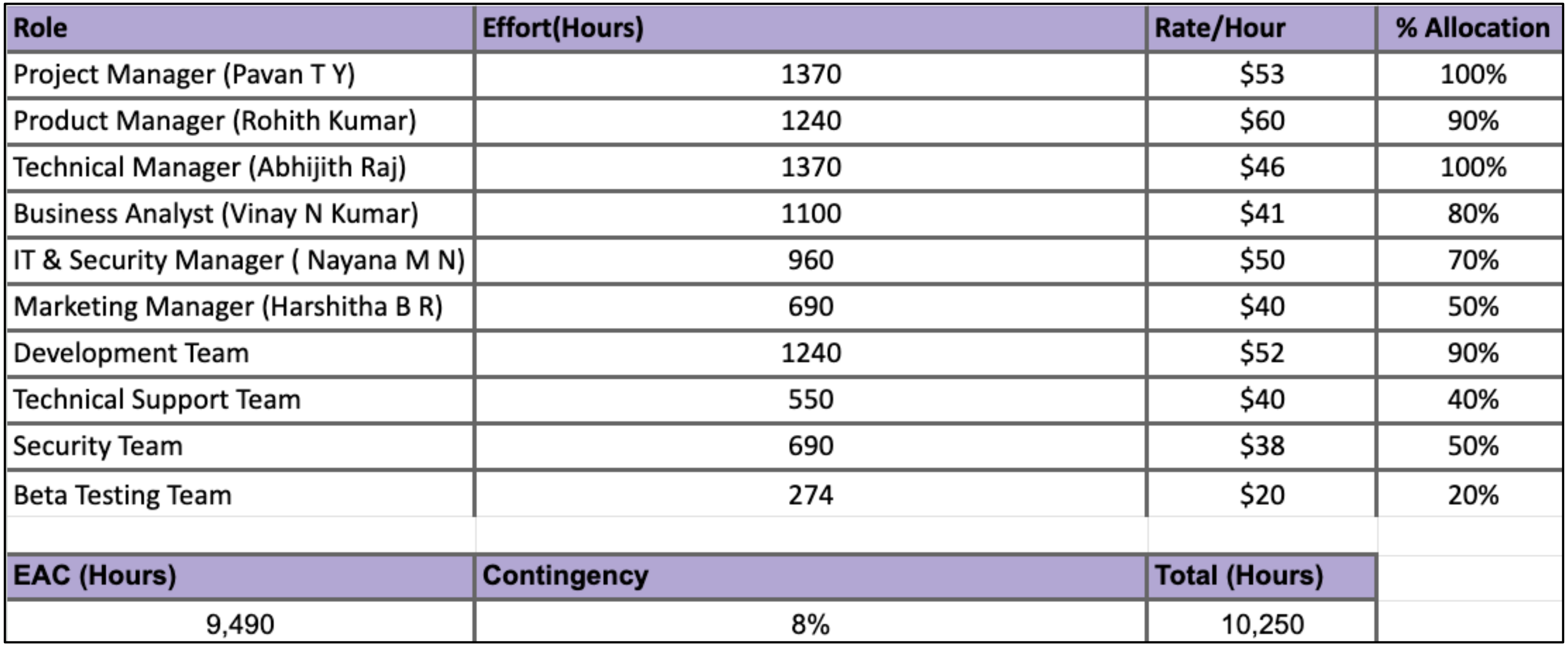
ID	PROJECT TASK	Resources								Task Total Cost
		Labor				Equipment			Miscellaneous	
		No. of People	Working Hours	Hourly Rate	Estimated Cost	Quantity	Unit Price	Estimated Cost		
30	3.0 Project Execution									
31	3.1 Frontend Development									
32	3.1.1 Design UI/UX using Figma	2	48.00	$72.00	$6,912.00	2.00	$800.00	$1,600.00		$8,512.00
33	3.1.2 Set up the frontend directory and folder structure with VS Code	4	8.00	$72.00	$2,304.00					$2,304.00
34	3.1.3 Code using React, HTML, Material UI with VS Code	4	190.00	$72.00	$54,720.00	4.00	$800.00	$600.00		$55,320.00
35	3.1.4 Manage the application state with Redux	3	56.00	$72.00	$12,096.00					$12,096.00
36	3.1.5 Integrate APIs using Fetch or Axios	3	160.00	$72.00	$34,560.00					$34,560.00
37	3.1.6 Conduct frontend tests with Jest	4	32.00	$72.00	$9,216.00					$9,216.00
38	3.2 Backend Development									
39	3.2.1 Setup a backend server using spring boot	6	5.00	$72.00	$2,160.00	6.00	$800.00	$4,800.00		$6,960.00
40	3.2.2 Set up the Mock JSON data for the development	3	80.00	$72.00	$17,280.00					$17,280.00
41	3.2.3 Use Cross-Origin Resource Sharing (CORS) to pass data to the frontend	4	40.00	$72.00	$11,520.00					$11,520.00
42	3.2.4 Design and implement a RESTful API	6	160.00	$72.00	$69,120.00					$69,120.00
43	3.2.5 Create an API for two-factor authentication and fraud detection mechanism	4	56.00	$72.00	$16,128.00					$16,128.00
44	3.2.6 Implement an encrypted API for peer-to-peer transfers, merchant payments, and bill payments	4	96.00	$72.00	$27,648.00					$27,648.00
45	3.2.7 Implement APIs for the creation, updating, and management of user accounts and multi-bank account linking	5	96.00	$72.00	$34,560.00					$34,560.00
46	3.2.8 Implement a feature to provide transaction history and generate financial reports	4	32.00	$72.00	$9,216.00					$9,216.00
47	3.2.9 Implement an API to manage rewards, deliver personalized offers, and notify users of new rewards	4	56.00	$72.00	$16,128.00					$16,128.00
48	3.3 Database Development									
49	3.3.1 Set up and configure the AWS database service	2	40.00	$72.00	$5,760.00	2.00	$800.00	$1,600.00		$7,360.00
50	3.3.2 Design the database schema and implement ORM integration	2	10.00	$72.00	$1,440.00					$1,440.00
51	3.3.3 Implement data access layers using an ORM (e.g., Hibernate) to interact with the database	1	15.00	$72.00	$1,080.00					$1,080.00
52	3.3.4 Implement database security measures and automated backups	2	16.00	$72.00	$2,304.00					$2,304.00
53	3.4 Testing and Quality									
54	3.4.1 Create unit tests for both frontend and backend	4	48.00	$70.00	$13,440.00	4.00	$800.00	$3,200.00		$16,640.00
55	3.4.2 Conduct integration tests to ensure seamless interaction between application modules	2	32.00	$70.00	$4,480.00					$4,480.00
56	3.4.3 Perform end-to-end testing to validate functionality across devices	2	32.00	$70.00	$4,480.00					$4,480.00
57	3.4.4 Implement CI/CD pipelines using Jenkins for build, test, and deployment	2	56.00	$70.00	$7,840.00					$7,840.00

ID	PROJECT TASK	Resources								Task Total Cost
		Labor				Equipment			Miscellaneous	
		No. of People	Working Hours	Hourly Rate	Estimated Cost	Quantity	Unit Price	Estimated Cost		
58	3.5 Performance Monitoring									
59	3.5.1 Establish metrics to monitor project performance and progress	4	13.00	$68.00	$3,536.00					$3,536.00
60	3.5.2 Conduct regular performance reviews and status meetings	4	10.00	$68.00	$2,720.00					$2,720.00
									Phase total cost:	$382,448.00
61	4.0 Project Closure Phase									
62	4.1 Launch Application									
63	4.1.1 Develop a detailed launch plan outlining key activities and timelines	4	15.00	$72.00	$4,320.00					$4,320.00
64	4.1.2 Announce the application launch through various channels	2	15.00	$72.00	$2,160.00				$2,000.00	$4,160.00
65	4.2 Application Deployment									
66	4.2.1 Deploy the application to the live environment	5	24.00	$65.00	$7,800.00					$7,800.00
67	4.2.2 Conduct compliance checks to ensure adherence to regulations and standards	2	24.00	$72.00	$3,456.00					$3,456.00
68	4.2.3 Manage servers and hosting to optimize performance	3	20.00	$68.00	$4,080.00	2.00	$1,500.00	$3,000.00		$7,080.00
69	4.3 Provide User Support and Maintenance									
70	4.3.1 Set up a support system for post-launch inquiries and technical issues	2	15.00	$70.00	$2,100.00				$2,000.00	$4,100.00
71	4.3.2 Implement a maintenance plan to update and enhance the application based on user feedback and market standards	2	16.00	$65.00	$2,080.00				$2,000.00	$4,080.00
72	4.4 Finalize Project Closure									
73	4.4.1 Obtain sign-off from stakeholders	1	8.00	$72.00	$576.00					$576.00
74	4.4.2 Document lessons learned	1	13.00	$72.00	$936.00				$250.00	$1,186.00
75	4.4.3 Formally close the project and release all allocated resources, including personnel, equipment, and finances	6	16.00	$72.00	$6,912.00					$6,912.00
76	4.4.4 Conduct a post-implementation review	2	15.00	$70.00	$2,100.00				$200.00	$2,300.00
77	4.4.5 Prepare for client handover	5	8.00	$72.00	$2,880.00					$2,880.00
									Phase total cost:	$48,850.00
				Total Labor Cost:	$476,869.00		Total Material Cost:	$22,467.84	$6,450.00	$505,786.84
								Contingency	8%	$40,462.95
								Administrative	10%	$50,578.68
									Total Project Cost	$596,828.47

Appendix F: Resource Allocation Plan



Client	Zelle
Project	Perq's with Zelle
Budget Hours	**10,250**
Contract Value	$596,828.47
Project start date	September 2, 2024



Role	Effort(Hours)	Rate/Hour	% Allocation
Project Manager (Pavan T Y)	1370	$53	100%
Product Manager (Rohith Kumar)	1240	$60	90%
Technical Manager (Abhijith Raj)	1370	$46	100%
Business Analyst (Vinay N Kumar)	1100	$41	80%
IT & Security Manager (Nayana M N)	960	$50	70%
Marketing Manager (Harshitha B R)	690	$40	50%
Development Team	1240	$52	90%
Technical Support Team	550	$40	40%
Security Team	690	$38	50%
Beta Testing Team	274	$20	20%
EAC (Hours)	**Contingency**	**Total (Hours)**	
9,490	8%	10,250	