



IT423
Introduction to IT project management

< درّیّی >

Prepared by

Group members

Group members	
Taif Alrubaan (Leader)	442202301
Basma Alsulaim	441200595
Raghad Alboqami	442200455
Alanoud Alarifi	442201803
Arwa Mesloub	443204895

Supervised by
Dr. Mashael Alsaleh

1st Semester 1446

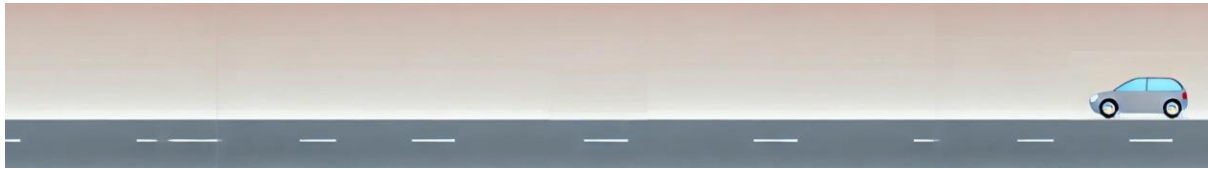


TABLE OF CONTENTS

1 Introduction	3
2 project initiation	3
2.1 proposal form.....	4
3 Project scope	6
3.1 Scope management plan	6
3.2 Requirements Management plan	7
3.2.1 Requirements Traceability Matrix (RTM)	8
3.3 Main functional and non-functional requirements.....	10
3.4 Scope Statement.....	11
3.5 WBS	12
3.6 WBS dictionary.....	14
4 project planning (WBDS table & Chart).....	16
5 technical outcomes and charts (Using Microsoft Project).....	20
5.1 Network Diagram with critical path	20
5.2 A Complete Gantt Chart.....	21
5.3 Project Timeline.....	21
5.4 Task allocation schedule	22
5.5 Task dependencies table.....	22
5.6 Hardware and Software tools	23
6 Cost estimation.....	23
6.1 Development Cost	23
6.2 Tools Cost	25
6.3 Total cost	26
7 Quality Attributes.....	27
8 External and internal	28
9 REFERENCES	29



1. INTRODUCTION

In today's world, many women struggle to find trusted and reliable driving instructors, making the process of learning to drive even more challenging. Instructors, on the other hand, face difficulties managing their growing businesses while staying connected with their students. This is where the درّیني app steps in.

درّیني is a mobile application designed specifically for women driving instructors to help them efficiently manage their students, track progress, and simplify their business operations. With its many functions, such as skill demonstrations, feedback gathering, and calendar management, the app helps teachers remain on top of things and stay in constant contact with their pupils. درّیني helps instructors improve their company and attract new students by offering a dependable platform for both parties. And that's only the start; درّیني provides additional resources to help driving instructors in all facets of their work.

This report is divided into three sections: the introduction, project initiation (including a detailed project description, project requirements, dates, and duration), and project scope (covering the scope management plan, requirements management plan, main functional and non-functional requirements, scope statement, Work Breakdown Structure (WBS), and WBS dictionary).

2. PROJECT INITIATION

In the following form you are going to answer the following:

- Project initiation
- Project Detailed Description
- Project Requirements
- Project Dates and Duration



2.1 PROPOSAL FORM

TABLE 1: [PROPOSAL FORM]

Student Name:
Domain:
Suggested Project Title: درّیبنی
Project Detailed Description: A. Problem Finding trustworthy and reliable driving instructors can be a challenge for women learning to drive, which could slow their progress. Women driving instructors also face challenges in effectively running their businesses, monitoring their students, and communicating with clarity. Traditional methods of business management, such as manual scheduling and feedback collection, are time-consuming and can limit growth and student engagement. B. Application solution The درّیبنی app gives both female driving instructors and students an intuitive mobile platform to address their respective challenges. For instructors , the app simplifies business management with features like skill demonstrations, gathering student feedback, scheduling lessons, and tracking student progress. These tools help instructors improve student engagement, streamline their operations, and establish their reputations while enhancing their training careers. For students , the app offers an easy way to search for certified instructors based on location and availability, schedule lessons, track their learning progress, and receive feedback from instructors. Students can also make payments through the app, rate and review their instructors, and stay in constant communication with them via in-app messaging. This ensures that students not only receive high-quality driving lessons but also have seamless learning experience. C. Application Type: iOS and Android mobile application
Project Requirements: Software: <ul style="list-style-type: none">○ IDE (Visual Studio Code)○ Mobile Application Development Language (React Native)



- Cloud Database Management System (Firebase)
- Google Calendar API
- Google Maps API
- Payment Gateway
- Figma
- GitHub
- Microsoft Word
- Google Forms
- WhatsApp
- Zoom

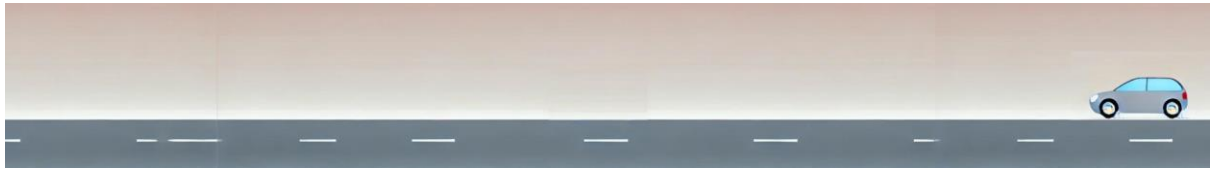
Hardware:

- **Computers for software development.**
- **Smartphones (IOS, Android) for testing.**

Project Start Date: September 1, 2024

Project End Date: March 1, 2025

Duration: 6 Months



3. PROJECT SCOPE

3.1 SCOPE MANAGEMENT PLAN

The **Scope Management Plan** for the "درّبيني" app ensures that the project remains focused and aligned with its objectives by managing and controlling any potential changes to the project scope.

The plan begins with a **detailed scope statement**, outlining the app's core objectives and key features, such as **user registration, booking, lesson scheduling, and progress tracking**. Exclusions, such as features that are out of scope for this phase (e.g., integration with third-party platforms or advanced analytics), will also be defined to manage expectations and avoid scope creep.

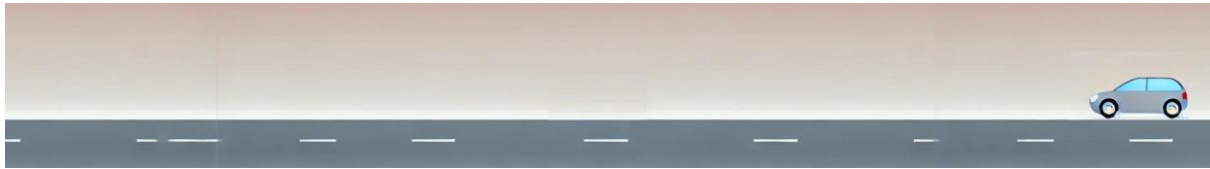
We will employ a **top-down approach** to develop the Work Breakdown Structure (WBS), breaking the project into smaller, manageable tasks. This will facilitate **clear responsibility assignment** and **improve tracking** throughout the project.

Deliverables will be regularly reviewed by stakeholders to ensure that they meet the agreed-upon requirements, with formal acceptance documented as part of the approval process.

Any proposed changes to the project scope, timeline, or resources will follow a **formal change control process** to ensure that their impact is properly understood and managed. The steps in the change control process are as follows:

1. **Document the Change Request:** A formal document will be created, specifying the change, its rationale, and the potential impact on the project.
2. **Analyze the Impact:** The project team will evaluate the change's effect on the timeline, budget, resources, and deliverables.
3. **Stakeholder Approval:** The change request will be submitted to key stakeholders for formal approval.
4. **Update Project Plans:** Once approved, the necessary updates will be made to the project plan, scope, schedule, and other documentation.
5. **Implementation and Monitoring:** The approved change will be implemented, and its progress will be closely monitored to ensure it does not negatively affect other deliverables.

This process ensures that any modifications to the project scope are carefully considered and aligned with the project's overall goals and resources.



3.2 REQUIREMENTS MANAGEMENT PLAN

To gather the requirements for the "دَرِّبِنِي" app, we will use a combination of surveys and interviews with key stakeholders, including potential users (e.g., women learning to drive) and certified trainers. This approach ensures that a broad and relevant set of requirements is collected, fully aligned with the app's objectives and user needs.

To effectively manage these requirements, we will implement a Requirements Traceability Matrix (RTM). The RTM will track each requirement throughout the project lifecycle by mapping it to its source, category, priority, and status. This matrix will allow the project team to:

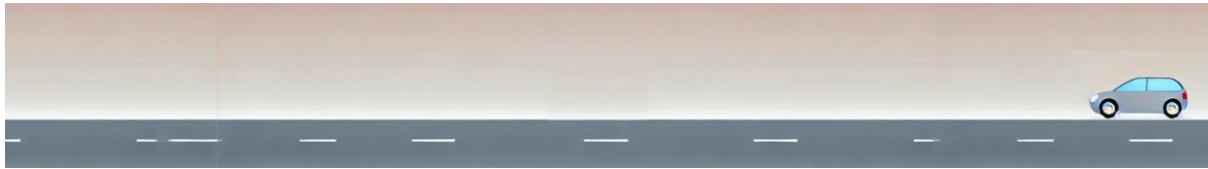
- Monitor progress: Track the implementation and fulfillment of each requirement.
- Ensure traceability: Confirm that every requirement is tied back to the stakeholder who provided it, as well as the business goals it supports.
- Maintain transparency: Ensure that stakeholders can see the status of their requests and understand how each requirement fits into the project.

The RTM will include fields such as Requirement No., Name, Category, Source, Status, and Priority to ensure each requirement is properly documented and prioritized. By organizing the requirements using the **MoSCoW prioritization technique**, each requirement will be categorized into one of four categories:

- Must-Have: These are critical to the project's success, such as user registration, login, and search functionalities.
- Should-Have: These are important but not essential to the core functionality of the system, such as notifications and user preferences.
- Could-Have: These are desirable features that would enhance the user experience but are not mandatory, such as personalized dashboards or advanced statistics for trainers.
- Won't-Have: These are features that have been identified as unnecessary for the current project phase but could be included in future releases, such as virtual driving simulations or social sharing features.


By applying the MoSCoW prioritization, the most critical functionalities like user registration, booking, and scheduling lessons will be developed first, while less critical features can be deferred if necessary.

Using this structured approach, we will ensure that all identified needs are addressed and smoothly integrated into the app's development. The RTM will also be updated regularly, ensuring continuous alignment between project objectives and stakeholder expectations, ultimately contributing to the project's successful delivery.




3.2.1 REQUIREMENTS TRACEABILITY MATRIX (RTM)

Req. No.	Requirement	Category	Source	Priority / Status
R01	User registration for trainees	Functional	Stakeholder Interviews	Must-Have/ Completed
R02	User login functionality	Functional	Stakeholder Interviews	Must-Have/ Completed
R03	Search for trainers based on location and availability	Functional	Stakeholder Interviews	Must-Have / In Progress
R04	Book, reschedule, and cancel driving lessons	Functional	Project Scope Statement	Must-Have/ Approval
R05	View upcoming lessons and lesson history	Functional	Project Scope Statement	Should-Have / In Progress
R06	View driving progress and feedback from trainers	Functional	Stakeholder Interviews	Must-Have / Completed
R07	Secure payment gateway integration	Functional	Project Scope Statement	Must-Have / In Progress
R08	Rate and review trainers after each lesson	Functional	Stakeholder Interviews	Should-Have / Planned
R09	Notifications for upcoming lessons and schedule changes	Functional	Project Scope Statement	Could-Have / Planned
R10	In-app messaging for communication with trainers	Functional	Stakeholder Interviews	Should-Have/Pending Approval
R11	Trainer registration	Functional	Stakeholder Interviews	Must-Have / In Progress



R12	Trainer login functionality	Functional	Project Scope Statement	Must-Have Completed	/
R13	Trainer profile management.	Functional	Stakeholder Interviews	Must-Have In Progress	/
R14	Trainer sets availability for lessons	Functional	Project Scope Statement	Should-Have Planned	/
R15	Trainer accepts or declines lesson bookings	Functional	Stakeholder Interviews	Must-Have / Planned	
R16	Trainer views lesson schedule and tracks trainee progress	Functional	Project Scope Statement	Must-Have In Progress	/
R17	Trainer provides feedback and rates users after each lesson	Functional	Stakeholder Interviews	Should-Have Planned	/
R18	Admin manages and monitors system users	Functional	Stakeholder Interviews	Must-Have In Progress	/
R19	Admin approves or rejects trainer profiles	Functional	Stakeholder Interviews	Must-Have / Planned	
R20	System availability of 99%	Non-Functional	Requirements Management Plan	Must-Have Completed	/
R21	System response time within 5 seconds	Non-Functional	Requirements Management Plan	Should-Have Pending	/
R22	Monthly system maintenance	Non-Functional	Requirements Management Plan	Could-Have / Planned	
R23	Usability: Learn to use the system within 15 minutes	Non-Functional	Requirements Management Plan	Should-Have/Planned	



R24	Data encryption and compliance with privacy regulations	Non-Functional	Requirements Management Plan	Must-Have / In Progress
-----	---	----------------	------------------------------	-------------------------

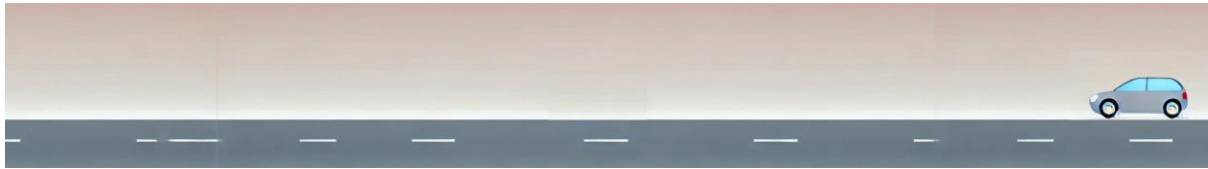
3.3 MAIN FUNCTIONAL AND NON-FUNCTIONAL REQUIREMENTS

Main functional Requirements:

- The user shall be able to register to the system as a new user (trainee).
- The user shall be able to log in to the system.
- The user shall be able to search for driving trainers based on location and availability.
- The user shall be able to book, reschedule, and cancel driving lessons.
- The user shall be able to view their upcoming lessons and lesson history.
- The user shall be able to view their driving progress and feedback from trainers.
- The user shall be able to pay for driving lessons through a secure payment gateway.
- The user shall be able to rate and review trainers after each lesson.
- The user shall receive notifications for upcoming lessons and changes in the schedule.
- The user shall be able to communicate with trainers through in-app messaging.
- The trainer shall be able to register to the system as a new trainer.
- The trainer shall be able to log in to the system.
- The trainer shall be able to create and manage their profile, including certifications and experience.
- The trainer shall be able to set their availability for lessons.
- The trainer shall be able to accept or decline lesson bookings.
- The trainer shall be able to view their lesson schedule and track the trainee's progress.
- The trainer shall be able to provide feedback and rate users after each lesson.
- The admin shall be able to manage and monitor system users, including trainers and trainees.
- The admin shall be able to approve or reject trainer profiles before they go live.

Non-Functional Requirements:

- Availability: The system shall be available 99% of the time to all users.



- Performance: The system response time shall be within 5 seconds.
- Maintainability: The system shall undergo maintenance every month.
- Usability: Users shall be able to learn how to use the system within 15 minutes without instructions.
- Security: The system shall protect users' personal and payment information through encryption and compliance with data privacy regulations.

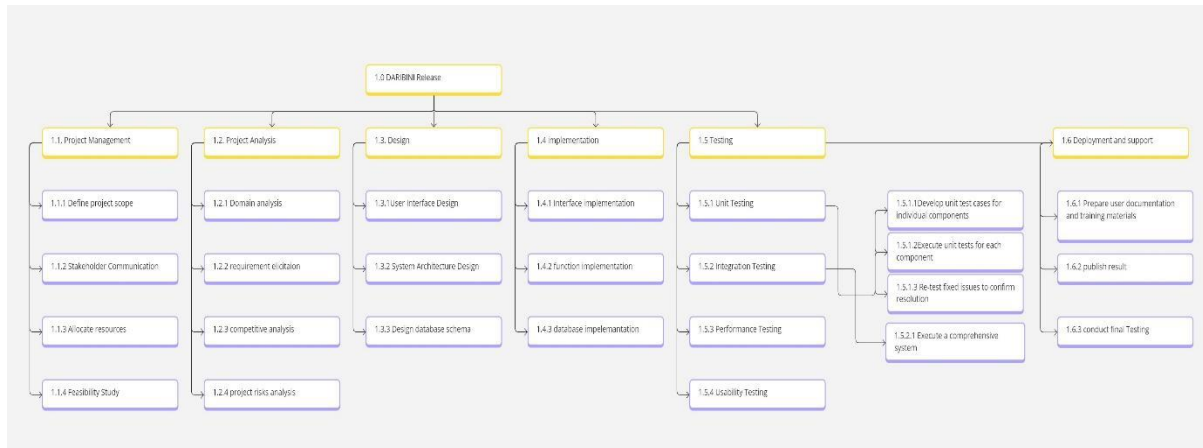
3.4 SCOPE STATEMENT

TABLE 2: [SCOPE STATEMENT]


Scope Description	The "دَرَبيني" app is a mobile platform tailored for women driving instructors and their students to manage driving lessons and improve learning experiences. Instructors can use features such as user registration, lesson scheduling, progress tracking, feedback collection, and payment processing to streamline their business operations. Students, on the other hand, can search for certified instructors, schedule lessons, track their progress, and communicate directly with their instructors through the app. The app will support the Arabic language and be available on both Android and iOS platforms.
Deliverables	A fully functional mobile application on both the App Store and Google Play, meeting all the functional and non-functional requirements for both students and instructors.
Acceptance Criteria	All core features and functionalities of "دَرَبيني" should be implemented and working as intended, including functional and non-functional requirements, without bugs, errors, nor usability issues
Constrains	The project must be completed within a six-month timeframe and requires a team proficient in mobile app development and UI/UX design. All development must adhere to the project's defined scope and resources.
Assumptions	The team has access to the necessary hardware and technology required to develop the system. The team possesses the necessary skills, expertise, and knowledge in web development to efficiently complete the project.



3.5 WBS CHART:



1.0 دربینی Release			
	1.1	Project Management	
		1.1.1 Define project scope	
		1.1.2 Stakeholder Communication	
		1.1.3 Allocate resources	
		1.1.4 Feasibility Study	
	1.2	Project Analysis	
		1.2.1 Domain analysis	
		1.2.2 Requirement elicitation	
		1.2.3 Competitive analysis	
		1.2.4 Project risks analysis	



	1.3 DESIGN		
		1.3.1 User Interface Design	
		1.3.2 System Architecture Design	
		1.3.3 Design database schema	
	1.4 IMPLEMENTATION		
		1.4.1Interface implementation	
		1.4.2Function implementation	
		1.4.3Database implementation	
	1.5 TESTING AND DEPLOYMENT		
		1.5.1 Unit Testing	
			1.5.1.1 Develop unit test cases for individual components
			1.5.1.2 Execute unit tests for each component
			1.5.1.3 Retest fixed issues to confirm resolution
		1.5.2 Integration Testing	
			1.5.2.1 Execute a comprehensive system
		1.5.3 Performance Testing	
		1.5.4Prepare user documentation and training materials	

			
		1.5.5 Publish result	

3.6 WBS DICTIONARY

Wbs Dictionary
Project Title: "دربيني"
Wbs Item Number: 1.2.1
Wbs Item Name: Domain analysis
<p>Description: This task is about researching and understanding everything related to driving instruction, especially for female instructors. It involves looking at how driving schools work, what instructors need to manage their students and business, and what students expect when they choose a driving teacher. The goal is to gather all the information needed to make sure the app supports both instructors and learners in the best way possible.</p> <p><i>This task relies on talking to instructors, students, and experts in the field to make sure we have a clear picture of how the app should work.</i></p>
Wbs Dictionary
Project Title: "دربيني"
Wbs Item Number: 1.3.1
Wbs Item Name: user interface design
<p>Description: This task is about creating a simple and easy-to-use design for the دربنی app. It includes making the app look good and ensuring it is easy to navigate for both driving instructors and learners. The design will focus on important features like booking lessons, tracking progress, and sending messages.</p> <p>This task depends on the needs of trainers and students, and feedback from early design examples (based on 1.2.2 Requirement Elicitation) will help improve the final version.</p>
Wbs Dictionary
Project Title: "دربيني"
Wbs Item Number: 1.5.2
WBS Item Name: Integration Testing



Description: This task involves testing how different parts of the دريبي app work together. The goal is to make sure everything runs smoothly. We will identify and fix any issues that come up when these parts are connected.

This task depends on the successful completion of individual component testing (like **1.4.1 Interface Implementation, 1.4.2 Function Implementation**) and making sure all pieces of the app are working as planned.

Wbs Dictionary

Project Title: "دريبي"

Wbs Item Number: 1.2.3

WBS Item Name: Competitive Analysis

Description: In this task, we will study other apps and solutions that provide similar services to driving instructors and students. We will look at their features, strengths, and weaknesses to understand how our app can stand out. This research will help us find opportunities to make our app better than the competition.

This task depends on researching competing apps to identify strengths and weaknesses.

Wbs Dictionary

Project Title: "دريبي"

Wbs Item Number: 1.4.3

WBS Item Name: Database Implementation

Description: This task involves creating the database that will store all the important information, like user profiles, lesson schedules, and payment records. It is crucial that the database is secure and organized so the app works smoothly and efficiently.

This task depends on creating a secure and reliable database that will store important information like user profiles and lesson schedules. It relies on proper organization and secure management to ensure the app functions smoothly.

Wbs Dictionary

Project Title: "دريبي"

Wbs Item Number: 1.5.3

WBS Item Name: Performance Testing



Description: This task involves testing how different parts of the دريبي app work together. The goal is to make sure everything runs smoothly. We will identify and fix any issues that come up when these parts are connected.

This task depends on ensuring the app can handle high usage without crashing.

Wbs Dictionary

Project Title: "دريبي"

Wbs Item Number: 1.5.5

WBS Item Name: Publish Result


Description: This is the final step where we officially release the app to the public. It includes uploading the app to the App Store and Google Play, making sure everything works correctly, and ensuring it's available for users to download.


This task depends on completing all development and testing phases and relies on proper app deployment processes to make it available on both iOS and Android platforms.

4 project planning (WBDS table & Chart)

Phase Name	Start Date	End Date	Durati on	Cost	Qualificatio ns	Tools	Deliverables
1.1Project Managemen t	01/09/202 4	20/09/202 4	21 Days	7,510.43 SAR	Project Managemen t Skills	MS Project, Word ,Zoom	Scope Document
1.1.1Define project scope	01/09/202 4	03/09/202 4	3 days	1,500 SA R		MS Projec t, Word	
1.1.2 Stakeholder	03/09/202 4	5/09/2024	3 Days	1,500 SAR		Zoom	

Communicati on							
1.1.3 Allocate resources	03/09/202 4	10/09/202 5	8 days	2,000 SAR		MS Project	
1.1.4 Feasibility Study	05/09/202 5	20/09/202 5	17 days	2,51043S AR		word	
1.2 Project Analysis	21/09/202 5	18/10/202 5	28 days	15,020.86 SAR	Research, Data Collection	MS Word, Excel Whatsap p	Domain Analysis Report
1.2.1 Domain analysis	21/09/202 5	25/09/202 5	5 days	3,755 SAR		MS Word, Excel, WhatsA pp	
1.2.2 Requirement elicitation	26/09/202 4	04/10/202 4	9 days	4,820 SAR		Excel	
1.2.3 Competitive analysis	05/10/202 4	11/10/202 4	7 days	3,222 SAR		MS Word, Excel	
1.2.4 Project risks analysis	12/10/202 4	18/10/202 4	7 days	3,223 SAR			
1.3 Design	19/10/202 4	27/10/202 4	9 days	18,779 SAR	Expertise in UI/UX design, system architecture	Figma, Firebase	Complete system design, including User

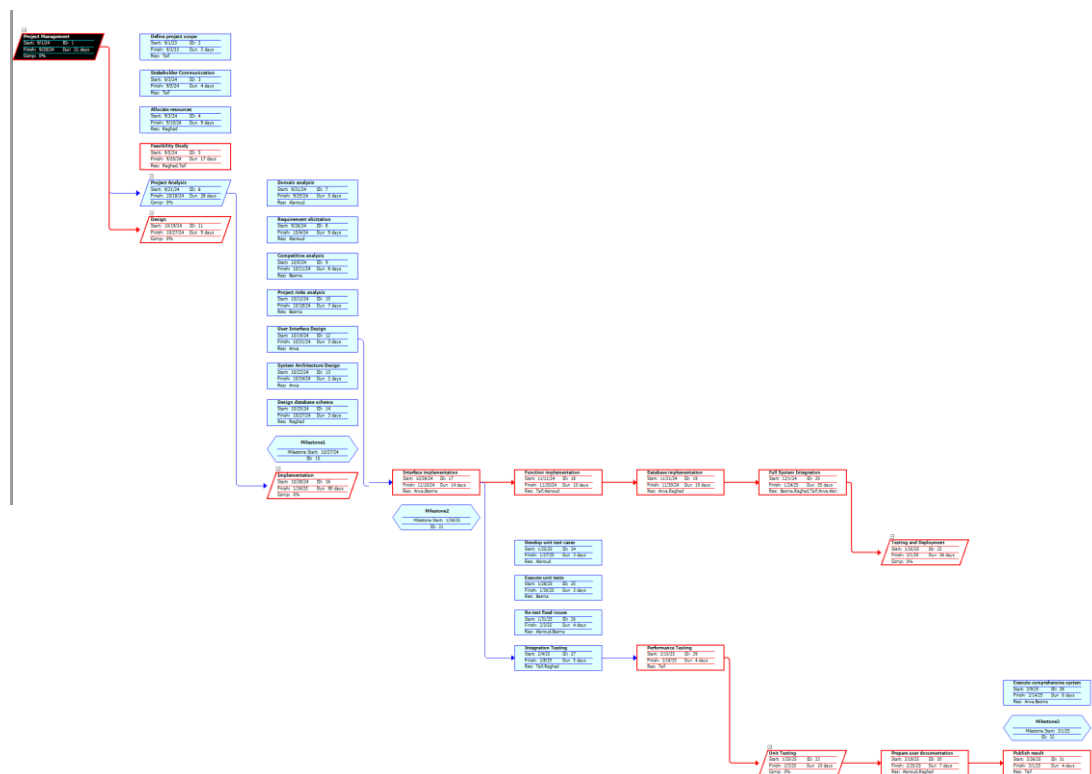
							
1.3.1 User Interface Design	19/10/2024	21/10/2024	3 days	6,000 SAR	, and database schema design.		Interface, Architecture, and Database Designs
1.3.2 System Architecture Design	22/10/2024	24/10/2024	3 days	7,500 SAR			
1.3.3 Design database schema	25/10/2024	27/10/2024	3 days	5,279 SAR			
1.4 Implementation	28/10/2024	24/01/2025	90 days	30,046 SAR	Proficiency in front-end, backend, and database development with experience in system integration.	Visual Studio Code, GitHub, Firebase	Fully functional system with implemented interface, backend functions, and connected database
1.4.1 Interface implementation	28/10/2024	10/11/2024	14 days	8,000 SAR			
1.4.2 Function implementation	11/11/2024	20/11/2024	10 days	10,500 SAR			
1.4.3 Database implementation	21/11/2024	30/11/2024	10 days	5,546			

							
1.4.4 Full system integration	1/12/2024	24/01/2025	55 days	6,000 SAR			
1.5 Testing and Deployment	25/01/2025	01/03/2025	37 days	13,145 SAR	Expertise in software testing, debugging, and deployment processes. Experience with test automation tools and user documentation preparation.	JIRA, Selenium, Google Calendar API	Fully tested system, comprehensive test reports, and user documentation ready for deployment.
1.5.1 Unit Testing	25/01/2025	03/02/2025	10 days	3,000 SAR			
1.5.1.1 Develop unit test cases	25/01/2025	27/01/2025	3 days				
1.5.1.2 Execute unit tests	28/01/2025	30/01/2025	3 days				
1.5.1.3 Re-test fixed issues	31/01/2025	03/02/2025	4 days				
1.5.2 Integration Testing	04/02/2025	08/02/2025	5 days	2,145 SAR			
1.5.2.1 Execute comprehensive system test	09/02/2025	14/02/2025	6 days)				
1.5.3 Performance Testing	15/02/2025	18/02/2025	4 day	2,000 SAR			

1.5.4 Prepare user documentation and training materials	19/02/2025	25/02/2025	7 day	4,000 SAR			
1.5.5 Publish result	26/02/2025	01/03/2025	4 day	2,000 SAR			

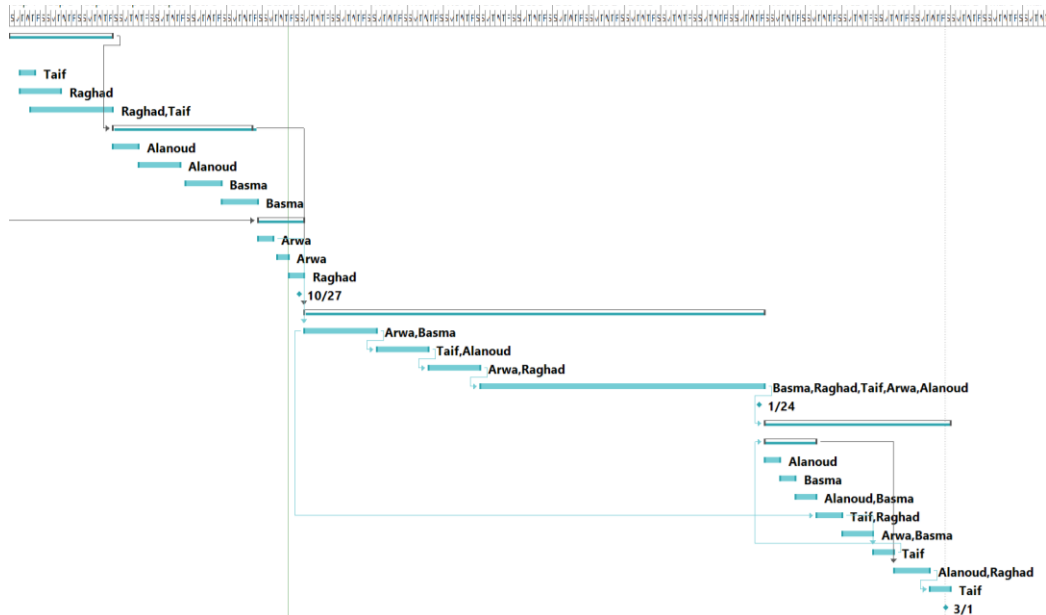
5 technical outcomes and charts (Using Microsoft Project)

5.1 Network Diagram with critical path





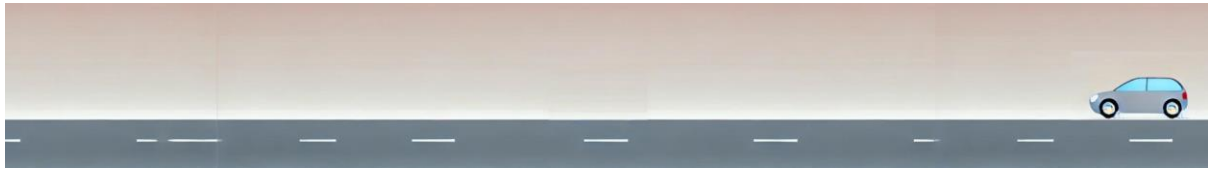
5.2 A Complete Gantt Chart



5.3 Project Timeline



	Task Name	Resource Names	Duration	Start	Finish	Predecessors
1	Project Management		21 days	Sun 9/1/24 12:00 AM	Fri 9/20/24 5:00 PM	
2	Define project scope	Taif	3 days	Fri 9/1/23 12:00 AM	Sun 9/3/23 5:00 PM	
3	Stakeholder Commu	Taif	4 days	Tue 9/3/24 12:00 AM	Thu 9/5/24 5:00 PM	
4	Allocate resources	Raghad	9 days	Tue 9/3/24 12:00 AM	Tue 9/10/24 5:00 PM	
5	Feasibility Study	Raghad, Taif	17 days	Thu 9/5/24 12:00 AM	Fri 9/20/24 5:00 PM	
6	Project Analysis		28 days	Sat 9/21/24 12:00 AM	Fri 10/18/24 12:00 AM	1
7	Domain analysis	Alanoud	5 days	Sat 9/21/24 8:00 AM	Wed 9/25/24 5:00 PM	
8	Requirement elicitat	Alanoud	9 days	Thu 9/26/24 12:00 AM	Fri 10/4/24 12:00 AM	
9	Competitive analysis	Basma	8 days	Sat 10/5/24 12:00 AM	Fri 10/11/24 5:00 PM	
10	Project risks analysis	Basma	7 days	Sat 10/12/24 12:00 AM	Fri 10/18/24 5:00 PM	
11	Design		9 days	Sat 10/19/24 8:00 AM	Sun 10/27/24 5:00 PM	155
12	User Interface Design	Arwa	3 days	Sat 10/19/24 8:00 AM	Mon 10/21/24 5:00 PM	
13	System Architecture	Arwa	2 days	Tue 10/22/24 5:00 PM	Thu 10/24/24 5:00 PM	
14	Design database sch	Raghad	3 days	Fri 10/25/24 8:00 AM	Sun 10/27/24 5:00 PM	
15	Milestone1		0 days	Sun 10/27/24 8:00 AM	Sun 10/27/24 8:00 AM	
16	Implementation		90 days	Mon 10/28/24 12:00 A	Fri 1/24/25 5:00 PM	6
17	Interface implement	Arwa, Basma	14 days	Mon 10/28/24 8:00 AM	Sun 11/10/24 5:00 PM	12
18	Function implement	Taif, Alanoud	10 days	Mon 11/11/24 8:00 AM	Wed 11/20/24 5:00 PM	17
19	Database implement	Arwa, Raghad	10 days	Thu 11/21/24 8:00 AM	Sat 11/30/24 5:00 PM	18
20	Full System Integrati	Basma, Raghad,	55 days	Sun 12/1/24 8:00 AM	Fri 1/24/25 5:00 PM	19
21	Milestone2		0 days	Fri 1/24/25 8:00 AM	Fri 1/24/25 8:00 AM	
22	Testing and Deployme		36 days	Sat 1/25/25 8:00 AM	Sat 3/1/25 5:00 PM	20
23	Unit Testing		10 days	Sat 1/25/25 8:00 AM	Mon 2/3/25 5:00 PM	29
24	Develop unit test c	Alanoud	3 days	Sat 1/25/25 8:00 AM	Mon 1/27/25 5:00 PM	
25	Execute unit tests	Basma	3 days	Tue 1/28/25 8:00 AM	Thu 1/30/25 5:00 PM	
26	Re-test fixed issues	Alanoud, Basma	4 days	Fri 1/31/25 8:00 AM	Mon 2/3/25 5:00 PM	
27	Integration Testing	Taif, Raghad	5 days	Tue 2/4/25 8:00 AM	Sat 2/8/25 5:00 PM	1755
28	Execute comprehens	Arwa, Basma	6 days	Sun 2/9/25 12:00 AM	Fri 2/14/25 5:00 PM	
29	Performance Testing	Taif	4 days	Sat 2/15/25 8:00 AM	Tue 2/18/25 5:00 PM	27
30	Prepare user docum	Alanoud, Raghad	7 days	Wed 2/19/25 8:00 AM	Tue 2/25/25 5:00 PM	23
31	Publish result	Taif	4 days	Wed 2/26/25 8:00 AM	Sat 3/1/25 5:00 PM	30
32	Milestone3		0 days	Sat 3/1/25 8:00 AM	Sat 3/1/25 8:00 AM	

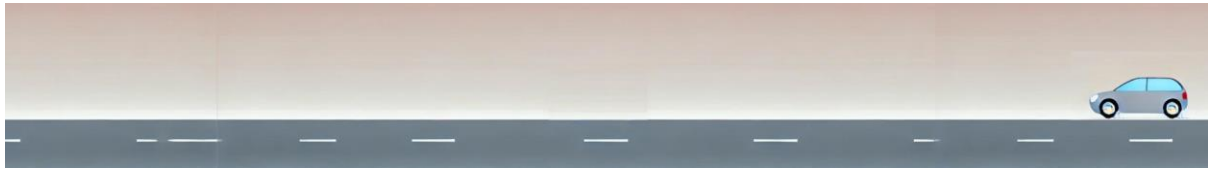


5.4 Task allocation schedule

	Task Name	Resource Names	Duration	Start	Finish	Predecessors
1	Project Management		21 days	Sun 9/1/24 12:00 AM	Fri 9/20/24 5:00 PM	
2	Define project scope	Taif	3 days	Fri 9/1/23 12:00 AM	Sun 9/3/23 5:00 PM	
3	Stakeholder Commu	Taif	4 days	Tue 9/3/24 12:00 AM	Thu 9/5/24 5:00 PM	
4	Allocate resources	Raghad	9 days	Tue 9/3/24 12:00 AM	Tue 9/10/24 5:00 PM	
5	Feasibility Study	Raghad, Taif	17 days	Thu 9/5/24 12:00 AM	Fri 9/20/24 5:00 PM	
6	Project Analysis		28 days	Sat 9/21/24 12:00 AM	Fri 10/18/24 12:00 AM	1
7	Domain analysis	Alanoud	5 days	Sat 9/21/24 8:00 AM	Wed 9/25/24 5:00 PM	
8	Requirement elicita	Alanoud	9 days	Thu 9/26/24 12:00 AM	Fri 10/4/24 12:00 AM	
9	Competitive analysis	Basma	8 days	Sat 10/5/24 12:00 AM	Fri 10/11/24 5:00 PM	
10	Project risks analysis	Basma	7 days	Sat 10/12/24 12:00 AM	Fri 10/18/24 5:00 PM	
11	Design		9 days	Sat 10/19/24 8:00 AM	Sun 10/27/24 5:00 PM	15S
12	User Interface Design	Arwa	3 days	Sat 10/19/24 8:00 AM	Mon 10/21/24 5:00 PM	
13	System Architecture	Arwa	2 days	Tue 10/22/24 5:00 PM	Thu 10/24/24 5:00 PM	
14	Design database sch	Raghad	3 days	Fri 10/25/24 8:00 AM	Sun 10/27/24 5:00 PM	
15	Milestone1		0 days	Sun 10/27/24 8:00 AM	Sun 10/27/24 8:00 AM	
16	Implementation		90 days	Mon 10/28/24 12:00 A	Fri 1/24/25 5:00 PM	6
17	Interface implement	Arwa, Basma	14 days	Mon 10/28/24 8:00 AM	Sun 11/10/24 5:00 PM	12
18	Function implement	Taif, Alanoud	10 days	Mon 11/11/24 8:00 AM	Wed 11/20/24 5:00 PM	17
19	Database implement	Arwa, Raghad	10 days	Thu 11/21/24 8:00 AM	Sat 11/30/24 5:00 PM	18
20	Full System Integrati	Basma, Raghad,	55 days	Sun 12/1/24 8:00 AM	Fri 1/24/25 5:00 PM	19
21	Milestone2		0 days	Fri 1/24/25 8:00 AM	Fri 1/24/25 8:00 AM	
22	Testing and Deployme		36 days	Sat 1/25/25 8:00 AM	Sat 3/1/25 5:00 PM	20
23	Unit Testing		10 days	Sat 1/25/25 8:00 AM	Mon 2/3/25 5:00 PM	29
24	Develop unit test c	Alanoud	3 days	Sat 1/25/25 8:00 AM	Mon 1/27/25 5:00 PM	
25	Execute unit tests	Basma	3 days	Tue 1/28/25 8:00 AM	Thu 1/30/25 5:00 PM	
26	Re-test fixed issues	Alanoud, Basma	4 days	Fri 1/31/25 8:00 AM	Mon 2/3/25 5:00 PM	
27	Integration Testing	Taif, Raghad	5 days	Tue 2/4/25 8:00 AM	Sat 2/8/25 5:00 PM	17SS
28	Execute comprehen	Arwa, Basma	6 days	Sun 2/9/25 12:00 AM	Fri 2/14/25 5:00 PM	
29	Performance Testing	Taif	4 days	Sat 2/15/25 8:00 AM	Tue 2/18/25 5:00 PM	27
30	Prepare user docum	Alanoud, Raghad	7 days	Wed 2/19/25 8:00 AM	Tue 2/25/25 5:00 PM	23
31	Publish result	Taif	4 days	Wed 2/26/25 8:00 AM	Sat 3/1/25 5:00 PM	30
32	Milestone3		0 days	Sat 3/1/25 8:00 AM	Sat 3/1/25 8:00 AM	

5.5 Task dependencies table

	Task Name	Resource Names	Duration	Start	Finish	Predecessors
1	Project Management		21 days	Sun 9/1/24 12:00 AM	Fri 9/20/24 5:00 PM	
2	Define project scope	Taif	3 days	Fri 9/1/23 12:00 AM	Sun 9/3/23 5:00 PM	
3	Stakeholder Commu	Taif	4 days	Tue 9/3/24 12:00 AM	Thu 9/5/24 5:00 PM	
4	Allocate resources	Raghad	9 days	Tue 9/3/24 12:00 AM	Tue 9/10/24 5:00 PM	
5	Feasibility Study	Raghad, Taif	17 days	Thu 9/5/24 12:00 AM	Fri 9/20/24 5:00 PM	
6	Project Analysis		28 days	Sat 9/21/24 12:00 AM	Fri 10/18/24 12:00 AM	1
7	Domain analysis	Alanoud	5 days	Sat 9/21/24 8:00 AM	Wed 9/25/24 5:00 PM	
8	Requirement elicita	Alanoud	9 days	Thu 9/26/24 12:00 AM	Fri 10/4/24 12:00 AM	
9	Competitive analysis	Basma	8 days	Sat 10/5/24 12:00 AM	Fri 10/11/24 5:00 PM	
10	Project risks analysis	Basma	7 days	Sat 10/12/24 12:00 AM	Fri 10/18/24 5:00 PM	
11	Design		9 days	Sat 10/19/24 8:00 AM	Sun 10/27/24 5:00 PM	15S
12	User Interface Design	Arwa	3 days	Sat 10/19/24 8:00 AM	Mon 10/21/24 5:00 PM	
13	System Architecture	Arwa	2 days	Tue 10/22/24 5:00 PM	Thu 10/24/24 5:00 PM	
14	Design database sch	Raghad	3 days	Fri 10/25/24 8:00 AM	Sun 10/27/24 5:00 PM	
15	Milestone1		0 days	Sun 10/27/24 8:00 AM	Sun 10/27/24 8:00 AM	
16	Implementation		90 days	Mon 10/28/24 12:00 A	Fri 1/24/25 5:00 PM	6
17	Interface implement	Arwa, Basma	14 days	Mon 10/28/24 8:00 AM	Sun 11/10/24 5:00 PM	12
18	Function implement	Taif, Alanoud	10 days	Mon 11/11/24 8:00 AM	Wed 11/20/24 5:00 PM	17
19	Database implement	Arwa, Raghad	10 days	Thu 11/21/24 8:00 AM	Sat 11/30/24 5:00 PM	18
20	Full System Integrati	Basma, Raghad,	55 days	Sun 12/1/24 8:00 AM	Fri 1/24/25 5:00 PM	19
21	Milestone2		0 days	Fri 1/24/25 8:00 AM	Fri 1/24/25 8:00 AM	
22	Testing and Deployme		36 days	Sat 1/25/25 8:00 AM	Sat 3/1/25 5:00 PM	20
23	Unit Testing		10 days	Sat 1/25/25 8:00 AM	Mon 2/3/25 5:00 PM	29
24	Develop unit test c	Alanoud	3 days	Sat 1/25/25 8:00 AM	Mon 1/27/25 5:00 PM	
25	Execute unit tests	Basma	3 days	Tue 1/28/25 8:00 AM	Thu 1/30/25 5:00 PM	
26	Re-test fixed issues	Alanoud, Basma	4 days	Fri 1/31/25 8:00 AM	Mon 2/3/25 5:00 PM	
27	Integration Testing	Taif, Raghad	5 days	Tue 2/4/25 8:00 AM	Sat 2/8/25 5:00 PM	17SS
28	Execute comprehen	Arwa, Basma	6 days	Sun 2/9/25 12:00 AM	Fri 2/14/25 5:00 PM	
29	Performance Testing	Taif	4 days	Sat 2/15/25 8:00 AM	Tue 2/18/25 5:00 PM	27
30	Prepare user docum	Alanoud, Raghad	7 days	Wed 2/19/25 8:00 AM	Tue 2/25/25 5:00 PM	23
31	Publish result	Taif	4 days	Wed 2/26/25 8:00 AM	Sat 3/1/25 5:00 PM	30
32	Milestone3		0 days	Sat 3/1/25 8:00 AM	Sat 3/1/25 8:00 AM	



5.6 Hardware and Software tools

- project plan 365

6 Cost estimation

6.1 Development Cost

- AFP (Adjusted Function Points)

In order to calculate the Adjusted Function Points, we will use the following formula:

Adjusted Function Points = Total Function Points * Adjustment Factor

The formula used to calculate the Total Function Points is as follows:

Total Function Points = Number of inputs FPs + Number of outputs FPs + Number of files FPs + Number of inquiry FPs + Number of interfaces FPs

- **Number of inputs FPs:** The count of unique user inputs that enter the system.
- **Number of outputs FPs:** The count of unique user outputs generated by the system.
- **Number of files FPs:** The count of unique data groups or files that the system internally maintains.
- **Number of inquiry FPs:** The count of unique user inquiries or data retrievals from the system.
- **Number of interfaces FPs:** The count of unique interfaces with external systems.

By assuming the values, we calculate function points as follows: Function Points = 30 + 35 + 10 + 30 + 20 = 125 FPs, Adjustment Factor = 1.2

Adjusted Function Points = 125 * 1.2 = 150 FBs

- ASLOC (number of lines of generated code)

Since we are using **JavaScript** for the project, which has an average **LOC/FP** of **47**[1], we can calculate the Adjusted Source Lines of Code (ASLOC) as follows:

- **ASLOC = AFP * JavaScript LOC/FP**
- **ASLOC = 150 * 47 = 7,050 LOC**
 - Number of staff -Month

Number of staff-months = ASLOC / Productivity

- Productivity = 350
- 7,050 / 350 = 20.14 SM
- 20.14 staff-months / 6 months = 3.36 persons ~ 4

So, we will need approximately 4 people for 6 months.



- Salary for staff development

- Number of people = 4
- Average salary for js developer in Saudi Arabia [2] = 13,700 SAR
- Duration = 6 months

Total salary cost = Number of people * Average salary * Number of months

Total salary cost = 4 * 13,700 * 6 = 328,800 SAR

- Effort estimation using COCOMO model

- Select the constant values

In this project we will be using the Organic COCOMO model to predict the cost estimation. It is the most appropriate model when the project and the project team is small and the problem is well understood. The constants are a and b where a= 2.4, b= 1.05 [3].

- **Cost drivers**

Cost Driver	Rating	Effort Multiplier
RELY	High	1.15
DATA	Nominal	1.00
CPLX	Nominal	1.00
TIME	High	1.11
STOR	High	1.06
TURN	Nominal	1.00
ACAP	Nominal	1.00
AEXP	Nominal	1.00
PCAP	High	0.86
LEXP	High	0.95
TOOL	High	0.91
SCED	Nominal	1.00
Effort Adjustment Factor (Product of Effort Multipliers) = 1.15 * 1.00 * 1.00 * 1.11 * 1.06 * 1.00 * 1.00 * 1.00 * 0.86 * 0.95 * 0.91 * 1.00 = 1.006		



- **Effort using COCOMO model**

$a = 2.4$, $b = 1.05$, $EAF = 1.006$, $KLOC = 7.05$

- $\text{Effort} = (a * (KLOC)^b) * EAF$
- $\text{Effort} = 2.4 * (7.05)^{1.05} * 1.006 = 17.82 \text{ person-months}$
- $\text{Effort} \approx 18 \text{ person-months}$

$\text{Cost} = \text{Salary} * \text{Effort}$


$\text{Cost} = 13,700 \text{ SAR} * 18 = 282,600 \text{ SAR}$

The higher estimate of 328,800 SAR was chosen as best practice dictates using the higher of two estimates when they are within 20% of each other .

6.2 Tools Cost

Using Bottom-Up Estimation method to calculate the cost of each software tool based on its monthly rate, the number of users, and the project duration

Software Tool	Estimated Cost (SAR)
Microsoft Project	$125 \text{ per month} \times 6 = 750$
Figma	$100 \text{ per month} \times 6 = 600$
Firebase	$250 \text{ per month (estimated usage)} \times 6 = 1,500$
Google Maps API	$167 \text{ per month (estimated usage)} \times 6 = 1,002$
JIRA	$100 \text{ per month} \times 6 = 600$
GitHub	$50 \text{ per month} \times 6 = 300$
Microsoft Word & Excel	Free



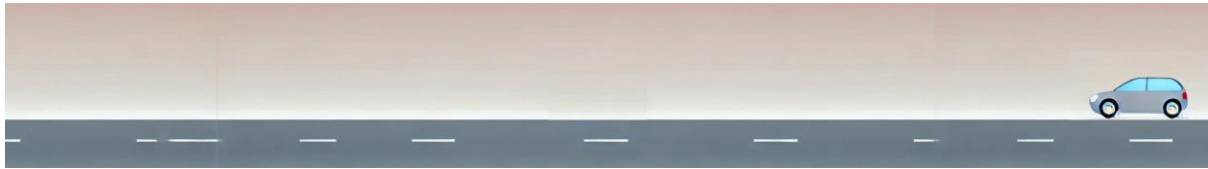
Zoom	Free version
WhatsApp	Free
Selenium	Free
Visual Studio Code	Free
Google Calendar API	Free with basic GCP usage
Payment Gateway	Estimated transaction fees = 500
Total cost	5,252 SAR

Hardware Tool	Estimated cost per Unit (SAR)	Quantity	Total Cost (SAR)
Computers	4,500	4	18,000
Smartphones (iOS and Android)	2,500	2	5,000
Total cost	23,000 SAR		

Total (Hardware + Software): 28,252 SAR

6.3 Total cost

Total Project Cost (Development Cost + Tools Cost): 328,800+28,252= 357,052SAR.



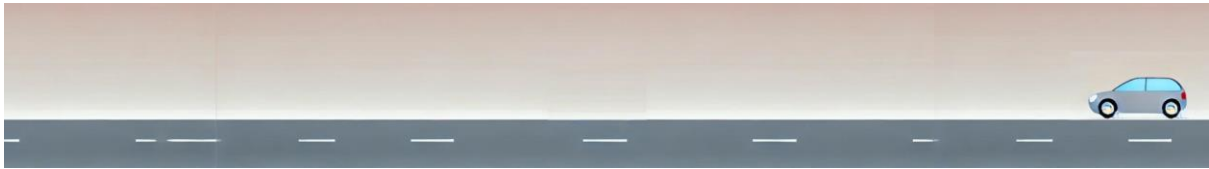
7 Quality Attributes

Quality Attribute	Description	Measurement	Assessment
Availability	The system shall be available 99% of the time to all users.	By using System Availability Formula [4].	$Availability = \frac{Uptime}{(Uptime + Downtime)}$
Performance	The system response time shall be within 5 seconds.	By using the Average Page Loading Time formula [5].	$Page\ Load\ Time = Document\ Completion\ Time - Start\ Time$
Maintainability	The system shall undergo maintenance every month.	By using the Cyclomatic Complexity formula [6].	$CC = Edges - Nodes + 2 \times Predicate\ Nodes$
Usability	Users shall be able to learn how to use the system within 15 minutes without instructions.	By using the Completion Rate/ Effectiveness formula [7].	$Effectiveness = \left(\frac{Number\ of\ Tasks\ Completed\ Successfully}{Total\ Number\ of\ Tasks\ Undertaken} \right) \times 100$
Security	The system shall protect users' personal and payment information through encryption and compliance with data privacy regulations.	By using RSA encryption [8].	$C = P^e \bmod n$ $\text{Where } N = p * q,$ $\text{and } D = e^{-1} \bmod (p-1)(q-1)$



8 External and internal

External Quality Attribute		Internal Quality Attribute
Availability	→	System Availability Measures the percentage of time the system is accessible to users [4].
Performance	→	Average Page Loading Time Calculates the time taken for a page to load, indicating system responsiveness [5].
Maintainability	→	Cyclomatic Complexity Assesses code complexity by counting the independent paths, aiding in maintainability planning [6].
Usability	→	Completion Rate / Effectiveness Evaluates the percentage of successfully completed tasks, indicating ease of use and user efficiency [7].
Security	→	RSA Encryption Uses encryption to protect sensitive information, ensuring data security and privacy compliance [8].



9 REFERENCES

- [1] Admin, "Function Point Languages Table," *QSM Software Project Estimation*, Dec. 06, 2022. <https://www.qsm.com/resources/function-point-languages-table>
- [2] "Javascript Developer Average Salary in Saudi Arabia 2024 - The complete guide." https://www.salaryexplorer.com/average-salary-wage-comparison-saudi-arabia-javascript-developer-c191j10569?expand_article=1
- [3] "Software Engineering | COCOMO Model - javatpoint," *www.javatpoint.com*. <https://www.javatpoint.com/cocomo-model>
- [4] Fiix Software. (n.d.). *System availability*. Retrieved from <https://fiixsoftware.com/glossary/system-availability/#:~:text=For%20example%2C%20let's%20say%20you,hours%20in%20a%20single%20month>
- [5] Stackify. (2020). *Application performance metrics*. Retrieved from <https://stackify.com/application-performance-metrics/>
- [6] Dr. Babasaheb Ambedkar University. (n.d.). *Cyclomatic complexity*. Retrieved from <https://bbau.ac.in/dept/CS/TM/Cyclomatic.pdf>
- [7] UsabilityGeek. (2018). *Usability metrics: A guide to quantify system usability*. Retrieved from <https://usabilitygeek.com/usability-metrics-a-guide-to-quantify-system-usability/>
- [8] Simplilearn. (2023). *RSA algorithm*. Retrieved from <https://www.simplilearn.com/tutorials/cryptography-tutorial/rsa-algorithm>