KING SAUD UNIVERSITY
COLLEGE OF COMPUTER AND INFORMATION SCIENCES
INFORMATION TECHNOLOGY DEPARTMENT

كلية علوم الحاسب والمعلومات
قسم تقنية المعلومات

IT423: INTRODUCTION TO
PROJECT MANAGEMENT

Morshed

Project Release1

Prepared by:

<table>
<tr><td colspan="2">Section: 56672 (Group#1)</td></tr>
<tr><th>Name</th><th>ID</th></tr>
<tr><td>Wiam Baalahatar (Leader)</td><td>443200416</td></tr>
<tr><td>Hind Alhijailan</td><td>443200971</td></tr>
<tr><td>Lamya Aljaithin</td><td>442201857</td></tr>
<tr><td>Sarah Aldbasi</td><td>443200520</td></tr>
<tr><td>Khloud Aldoayan</td><td>443201002</td></tr>
</table>

Supervised by
L.Nora Madi

1st Semester 1446

REVISION TABLE

Page#	Section#	Author	Corrected by (Reviewer)
4	Introduction	Khloud Aldoayan	
5	Project Initiation	Khloud Aldoayan	Wiam Baalhatar
6	Project Initiation	Sarah Aldbasi	Lamya Aljaithin
7	Project Requirements	Sarah Aldbasi	Lamya Aljaithin
8	Scope management plan	Wiam Baalahrar	Hind Alhijailan
9	Requirements management plan	Khloud Aldoayan	Hind Alhijailan
10	Main functional and non-functional requirements	Lamya Aljaithin	Sarah Aldbasi
11	Scope statement	Lamya Aljaithin	Sarah Aldbasi
12	WBS	Hind Alhijailan	Wiam Baalahtar
13	WBS Dictionary	Hind Alhijailan	Wiam Baalahtar
19	WBDS Table	Hind Alhijailan	Wiam Baalahtar - Lamya Aljaithin
20	WBDS Chart	Hind Alhijailan	Wiam Baalahtar
21	Network Diagram with critical path	Wiam Baalahtar	Hind Alhijailan - Lamya Aljaithin – Khloud Aldoayan
23	A complete Gantt Chart	Lamya Aljaithin	Hind Alhijailan – Khloud Aldoayan
26	Project timeline	Khloud Aldoayan +Sarah Aldbasi	Wiam Baalahtar - Lamya Aljaithin
27	Task allocation schedule	Khloud Aldoayan +Sarah Aldbasi	Wiam Baalahtar - Lamya Aljaithin
28	Task dependencies table	Khloud Aldoayan +Sarah Aldbasi	Wiam Baalahtar - Lamya Aljaithin
29	Hardware and Software Tools	Sarah Aldbasi	
31	AFP (Adjusted Function Points)	Wiam Baalahtar	Hind Alhijailan
32	ASLOC (number of lines of generated code)	Wiam Baalahtar	Hind Alhijailan – Khloud Aldoayan

33	Number of staff - Month	Wiam Baalahtar	Hind Alhijailan
34	Salary for development staff	Wiam Baalahtar	Hind Alhijailan
35	Effort estimation using COCOMO model	Khloud Aldoayan	Hind Alhijailan
38	Tools Cost	Hind Alhijailan	Wiam Baalahtar
39	Total cost	Hind Alhijailan	Wiam Baalahtar
40	Quality Attributes	Lamya Aljaithin	Hind Alhijailan
41	External and internal	Sarah Aldbasi	

Roles and Responsibilities

Member	Role	Responsibilities
Wiam Baalahtar	Leader	Scope management plan – Requirements management plan- Domain – AFP – ASLOC - Number of staff -Month -Salary for development staff -Network Diagram with critical path
Khloud Aldoayan	Member	Introduction – Project Detailed Description 'Problem Part' - Effort estimation using COCOMO model - Project timeline -Task allocation schedule-Task dependencies table
Sarah Aldbasi	Member	Solution – Project requirements – project dates and duration - External and internal - Hardware and Software Tools - Project timeline -Task allocation schedule-Task dependencies table
Lamya Aljaithin	Member	Functional requirements – Non-functional requirements – Scope statement – Gnatt chart – Quality attributes - Quality Attributes - A complete Gantt Chart
Hind Alhijailan	Member	WBS – WBS dictionary - Total cost - Tools Cost-WBDS Table - WBDS Chart

Meeting Plans

<table>
<tr><th>Meeting number</th><th>Date</th><th>Time</th><th>Venue</th><th>Attendees</th></tr>
<tr><td colspan="5">All the work was conducted through continuous communication via the WhatsApp group during the available period for working on Phase.</td></tr>
</table>

TABLE OF CONTENTS

TABLE OF TABLES

TABLE OF FIGURE

1. INTRODUCTION

Sometimes, we yearn to explore the other side of the world, uncover new horizons, and embrace fresh experiences. Travel has become a significant part of our lives, offering joy, cultural enrichment, and the opportunity to create lasting memories. It allows us to immerse ourselves in different cultures and craft beautiful experiences we'll cherish for years.

At Morshed, we are revolutionizing the way you discover and book guided tours. Our platform empowers tour guides to showcase their trips with precision and detail, offering insights into itineraries, schedules, pricing, participant limits, and tour types whether private or shared.

With Morshed, travelers gain access to a diverse range of tours and experiences, all in one place. Our integrated chat system allows for seamless communication between travelers and guides, ensuring all your questions are answered before you book. For shared tours, our group chat feature facilitates easy interaction among participants, maintaining privacy and eliminating the need to exchange personal contact information.

Customizable filters let you refine your search based on tour type, budget, and destination, helping you find the perfect guide and experience that aligns with your travel preferences. Discover a new way to explore the world with Morshed, where your ideal adventure is just a click away.

In this report, first, we will include the 'Project Initiation' which contains a Project Detailed Description that talks about the problem that we initiate the app for it, the application solution for the problem, and the type of our application website or app. Project Requirements that include hardware and software tools. And Project Dates and Duration that specify the start and end date, and the duration to finish the app. Second, we are going to talk about 'Project scope' which contains the Scope Management Plan that outlines how our project scope will be defined, validated, and controlled. Requirements Management Plan that outlines how requirements will be identified, documented, analyzed, prioritized, and managed throughout the project lifecycle. The Main functional and non-functional requirements help to define what a system should do and how it should perform. The Scope Statement defines the Scope Description, Acceptance Criteria, Constraints, Assumptions, and Deliverables of our project. Work Breakdown Structure (WBS) is a hierarchical decomposition of our project into smaller, more manageable components or tasks. And WBS Dictionary is a detailed to the WBS that provides additional information about each element in the WBS and offers descriptions, deliverables, activities, and other relevant details for each work package.

2. PROJECT INITIATION

2.1. PROPOSAL FORM

TABLE 1: PROPOSAL FORM

• **Student Name:** Wiam Baalahtar - Khloud Aldoayan - Sarah Aldbasi - Lamya Aljaithin - Hind Alhijailan.
• **Domain:** The Morshed project centers on enhancing the travel experience through technological solutions, specifically focusing on facilitating communication between travellers and tour guides. Through our digital platform, we aim to streamline the process of connecting travellers with tour guides, making it easier to discover and book personalized tour experiences. This platform can address common challenges in the travel industry such as overcoming communication hurdles, thereby ensuring a more seamless and enriched travel experience for users.
• **Suggested Project Title:** Morshed
• **Project Detailed Description:** **A. Problem** In today's bustling travel industry, many adventurers miss out on enriching experiences due to the opaque and cumbersome process of organizing and booking guided tours. A common grievance among these travellers is the lack of detailed information about tour itineraries, pricing, and the nature of the tours (private or public), leading to misaligned expectations. For instance, a family might find themselves on a strenuous hiking tour that was marketed as leisurely a mismatch affecting enjoyment and safety. Furthermore, potential travellers often have specific queries or concerns that remain unresolved due to inadequate communication channels, resulting in hesitation and dissatisfaction. This uncertainty is compounded in group tours where participants seek to interact without compromising their privacy, an existing platforms may not adequately meet a demand. Such inefficiencies not only tarnish the travel experience but also dissuade many from participating in guided tours, as evidenced by recent surveys indicating a 30% decline in satisfaction with traditional tour booking processes.

B. Application solution

Morshed makes it easy for travelers to find and book tours with trusted guides. Tour guides can post all the details about their trips, including the itinerary, price, and how many people can join, so travelers know exactly what to expect. The website helps travelers solve the problem of uncertainty by offering a chat feature where they can directly ask guides any questions. For group tours, there's a safe chat option that allows travelers to talk to each other without sharing personal information. Morshed also lets travelers filter tours based on their budget, destination, and type of trip, making it simple to find the right tour. With easy booking, travelers can quickly secure their tours, making Morshed the go-to website for planning smooth and enjoyable travel experiences.

C. Application Type:

Web application.

- **Project Requirements:**

Software:

- Programming language (Dart).
- IDEs (NetBeans and Visual Studio).
- Database management system (MySQL).
- Zoom.
- WhatsApp.
- Microsoft word.
- Jira.
- GitHub.
- Figma.
- Canva.
- Google Forms.
- Project plan 365.

Hardware:

- Laptops for development and testing.
- Server to host the web application and database.

- **Project Start Date:** 1 September, 2024
- **Project End Date:** 1 March, 2025
- **Duration:** 6 months

3. PROJECT SCOPE

3.1. SCOPE MANAGEMENT PLAN

The Scope Management Plan for the Morshed project outlines the processes and procedures for defining, prioritizing, verifying, and controlling the project's scope. The primary goal of this plan is to ensure that the project remains focused on delivering a high-quality website where tour guides can effectively advertise their trips, meeting the needs of both guides and travelers.

To define the project scope, we will engage in a structured process of gathering, documenting, and categorizing requirements. This involves a thorough analysis of user needs through stakeholder interviews, surveys, and market research, ensuring that the scope is both comprehensive and aligned with project objectives.

A clear prioritization strategy will be applied to the requirements to ensure that critical features are developed first. This plan incorporates specific criteria for evaluating and ranking the importance of requirements, including stakeholder impact, resource availability, and alignment with the project's objectives. This approach ensures that the most valuable functionalities are prioritized, balancing project goals with time and resource constraints.

To organize the scope into manageable tasks, a Work Breakdown Structure (WBS) will be employed. Using a top-down approach, the WBS divides the project into smaller, detailed components, beginning with the overall platform design and branching into specific areas such as user functionality, backend development, and interface design. Each work package will be clearly defined, ensuring straightforward task assignments that enhance project organization and control throughout development.

Once the scope is defined, regular scope verification will be conducted to confirm that project deliverables align with the agreed-upon requirements. Stakeholders will be involved in this process, verifying that aspects such as functionality, usability, and technical specifications meet the initial objectives. Each deliverable will undergo formal acceptance, with documentation and sign-off to confirm it meets the specified criteria.

Change management within the project scope will be controlled through a formal process. Any requests for scope changes will be evaluated for their potential impact on the project's timeline, resources, and budget. Only changes that align with the overall project goals and remain feasible within the project's constraints will be approved, ensuring the project remains focused and scope creep is avoided.

In summary, this Scope Management Plan provides a comprehensive framework for defining, prioritizing, and managing the Morshed project's scope. By establishing clear processes for prioritization, verification, and change control, the plan ensures that all work aligns with project objectives, resource availability, and stakeholder expectations, ultimately contributing to the successful delivery of the Morshed platform.

3.2. REQUIREMENTS MANAGEMENT PLAN

The Requirements Management Plan for the Morshed project outlines the approach for analyzing, documenting, and managing the project requirements. According to the PMBOK® Guide, Sixth Edition, a requirement is defined as "a condition or capability that is necessary to be present in a product, service, or result to satisfy a business need." The goal of this plan is to ensure that all requirements are accurately identified, properly managed, and successfully implemented throughout the project lifecycle.

To achieve this, the project will initiate the requirements gathering process through a targeted survey distribution method. This survey will be strategically designed to capture valuable insights from key stakeholders, including potential users, ensuring that their specific needs and preferences are fully understood. The collected requirements will be meticulously analyzed and documented in a structured format to ensure they are aligned with both the business goals and the overall project objectives. To facilitate seamless tracking and management, a Requirements Traceability Matrix (RTM) will be utilized, ensuring that each requirement is not only accounted for but also successfully integrated into the final product. This approach will guarantee that no requirement is overlooked, and that the platform remains fully aligned with stakeholder expectations throughout the development lifecycle.

The plan will also detail how to prioritize the requirements, ensuring that the most critical features are addressed first. Configuration management activities will be used to manage changes to the requirements, maintaining the integrity of the original scope while accommodating necessary adjustments. Product metrics will be employed to measure the performance and effectiveness of the platform against the documented requirements.

Additionally, this plan will establish clear methods for tracing requirements and capturing their attributes. This will allow the project team to ensure that each requirement is properly handled and integrated, contributing to the overall success of the Morshed platform by meeting the needs of its users.

3.3. MAIN FUNCTIONAL AND NON-FUNCTIONAL REQUIREMENTS

- **Functional Requirements:**

  - Both Tour Guides and Travelers shall be able to register on the website.
  - Both Tour Guides and Travelers shall be able to log in to their accounts.
  - Both Tour Guides and Travelers shall be able to log out of their accounts.
  - Tour Guides shall be able to advertise their tours on the website.
  - Tour Guides shall be able to include details and descriptions in their advertised tours, such as the proposed itinerary, schedule, destination, tour pricing, participant limits, and whether the tour is shared or private.
  - Tour Guides shall be able to view the travelers that booked their tours.
  - Travelers shall be able to browse tours on the website.
  - Travelers shall be able to book tours on the website.
  - Travelers shall be able to see the tour's details, such as the proposed itinerary, schedule, destination, tour pricing, and participant limits, and whether the tour is shared or private.
  - Travelers shall be able to search for a specific tour.
  - Travelers shall be able to filter the tours based on the tour type, destination, and budget.
  - Travelers shall be able to directly send a private message to Tour Guides.
  - Travelers shall be able to join a group chat with participants of the same tour.

  - **Non-Functional Requirements:**

Usability: The system shall be easy to browse and navigate, users should be able quickly find relevant information to complete tasks within 10 minutes.

Availability: The system shall be available 95% of the time, ensuring minimal downtime.

Recoverability: In the event of a system failure, the system shall be restored to full functionality within a maximum of 2 hours to ensure minimal disruption to users.

Response Time: The system shall provide a response time of less than 5 seconds.

Maintainability: System updates, including patches and new features, shall maintain a Maintainability Index (MI) of 70 or higher for all critical components, on a scale of 0–100, to ensure good maintainability and ease of future modifications.

3.4. Scope Statement

TABLE 2: SCOPE STATEMENT

• **Scope Description:** Morshed is a website designed to connect tour guides with travelers. Its primary goal is to establish an easy and trustworthy connection between the two users: Tour guides and Travelers, Morshed provides tour guides with a platform to advertise their upcoming tours, specifying all relevant details and view the Travelers that booked their tours. For travelers, it offers a platform to browse and book various tours, complete with search and filtering options. It also facilitates direct communication with tour guides for any inquiries. Additionally, Morshed features a group chat for travelers participating in the same tour, ensuring their privacy by protecting personal information, such as phone numbers, from being shared. Morshed support English language.
• **Deliverables:** A functional website that is compatible with Google chrome, Firefox and Safari, that serves Tour guides and Travelers and meets all functional and non-functional requirements.
• **Acceptance Criteria:** It’s considered a complete project when all the functional and non-functional requirements are met and delivered within the scheduled timeframe.
• **Constraints:** The project must be completed within 6 months. It requires a scalable database and server infrastructure. The system should handle real-time communication efficiently and meet high standards of quality, including performance, security, reliability, and user satisfaction across all features.
• **Assumptions:** The developers must have the skills, expertise, and knowledge in web development and real time commination that is needed to successfully complete the project.

3.5. WBS

- **WBS Table:**

TABLE 3: WBS TABLE

Morshed Release 1.0			
	1.1 Project initiation		
		1.1.1 Project objectives	
		1.1.2 Project deliverables	
		1.1.3 Resources Allocation	
			1.1.3.1 Determine team members
			1.1.3.2 Determine project’s budget
		1.1.4 Feasibility study	
	1.2 System analysis		
		1.2.1 Conduct meetings	
		1.2.2 Determine requirements	
			1.2.2.1 Benchmarking
			1.2.2.2 Interview stakeholders
			1.2.2.3 Document requirements
			1.2.2.4 Validate requirements
		1.2.3 Define scope	
	1.3 System design		
		1.3.1 Functions specifications	
			1.3.1.1 Determine inputs and outputs
			1.3.1.2 Determine acceptance criteria

		1.3.2 Database design	
		1.3.3 Interface design	
	1.4 System implementation		
		1.4.1 Database implementation	
		1.4.2 Frontend implementation	
		1.4.3 Backend implementation	
	1.5 System testing		
		1.5.1 Tour guides testing	
			1.5.1.1 Develop test cases
			1.5.1.2 Execute tests
			1.5.1.3 Resolve issues
		1.5.2 Travelers testing	
			1.5.2.1 Develop test cases
			1.5.2.2 Execute tests
			1.5.2.3 Resolve issues

3.6. WBS Dictionary

TABLE 4: WBS DICTIONARY

WBS Dictionary Entry September 28
Project Title: Morshred
WBS Item Number: 1.1.2
WBS Item Name: Project deliverables
Description: The task of identifying project deliverables for Morshed Release 1.0 involves outlining all products, services, and results that the project is expected to produce. This includes documentation, system components (such as frontend and backend modules), and system testing reports. Each deliverable should be specific, measurable, and linked to the project objectives to ensure successful completion. This task is dependent on the definition of project objectives (WBS item number 1.1.1) and the resources allocated under WBS item 1.1.3.
WBS Dictionary Entry September 28
Project Title: Morshred
WBS Item Number: 1.1.3.2
WBS Item Name: Determine project's budget
Description: The task of determining the project's budget for Morshed Release 1.0 involves estimating all financial resources required to complete the project. This includes costs associated with human resources (salaries of team members), software and hardware procurement, third-party services, infrastructure, and potential risks. Establishing an accurate budget is essential to ensuring that the project stays within financial constraints and achieves its objectives without overruns. This task depends on the resource allocation defined in WBS item number 1.1.3, and projects deliverables 1.1.2, and is influenced by the team members selected in WBS item number 1.1.3.1.

WBS Dictionary Entry September 28

Project Title: Morshred

WBS Item Number: 1.2.2.1

WBS Item Name: Benchmarking

Description:

The task of benchmarking for Morshed Release 1.0 involves comparing the planned system functionalities against how similar systems operate, identify key performance indicators, and set expectations for the project's features, such as tour guide management, traveler booking systems, and overall user experience. Benchmarking helps to define performance goals, validate project requirements, and ensure that Morshed Release 1.0 is competitive and aligned with user expectations.

This task depends on initial research and the collection of data from competitors, stakeholders, and other relevant systems, as outlined in WBS item number 1.2.2.

WBS Dictionary Entry September 28

Project Title: Morshred

WBS Item Number: 1.2.2.3

WBS Item Name: Document Requirements

Description:

The task of documenting requirements for Morshed Release 1.0 involves accurately recording all functional and non-functional requirements gathered from stakeholders and users. This document will serve as the blueprint for the design and development phases, ensuring that the system aligns with stakeholder expectations. This includes detailing the system's features, such as tour guide and traveller functionalities, database interactions, and user interface requirements.

This task depends on the outcome of interviews and meetings conducted during the requirements gathering process, as outlined in WBS item numbers 1.2.2.1 and 1.2.2.2.

WBS Dictionary Entry September 28
Project Title: Morshred
WBS Item Number: 1.2.2.4
WBS Item Name: Validate requirements
Description: The task of validating requirements for Morshed Release 1.0 involves reviewing and confirming that all gathered functional and non-functional requirements are correct, complete, and aligned with project goals. This process includes checking the documented requirements with stakeholders and ensuring that each requirement is feasible within the project’s scope and resources. Validating the requirements ensures that there are no misunderstandings or misalignments between the project team and stakeholders, and that the system to be developed meets user expectations. This task depends on the completion of the requirements documentation as outlined in WBS item number 1.2.2.3, and feedback from stakeholders involved in WBS item number 1.2.2.2.

WBS Dictionary Entry September 28
Project Title: Morshred
WBS Item Number: 1.2.3
WBS Item Name: Define scope
Description: The task of defining the scope for Morshed Release 1.0 involves determining the boundaries of the project in terms of its deliverables, functionalities, and features. This includes identifying what will be included in the and what will be excluded. It also involves setting clear limits on what will be developed, specifying user, and identifying any constraints or assumptions that will guide project execution. Defining the scope is critical to prevent scope creep and ensure that the project stays focused on its objectives. This task depends on the completion of the requirements gathering and validation activities from WBS items 1.2.2.3 and 1.2.2.4, as it ensures that the scope aligns with the confirmed project requirements.

WBS Dictionary Entry September 28
Project Title: Morshred
WBS Item Number: 1.3.1.1
WBS Item Name: Determine Inputs and Outputs
Description: The task of determining inputs and outputs for the Morshed system involves identifying the data that the system will collect from users and the corresponding information it will generate. This process is critical to ensure that the system's functionality supports user needs and produces the intended results. This task is dependent on the scope and requirements defined in WBS item number 1.2.2 and 1.2.3.

4. PROJECT PLANNING (WBDS TABLE & CHART)

- ## WBDS Table

TABLE 5 WBDS TABLE

Phase Name	Start Date	End Date	Duration	Cost (SAR)	Qualifications	Tools	Deliverables
Project initiation	1 Sep 2024	14 Sep 2024	14 days	8,678.33	• Project sponsor	• MS Word • Zoom • WhatsApp • Project Plan 365	• Feasibility study • Proposal
System analysis	15 Sep	30 Sep	16 days	10,844.95	• System analyst • Stakeholder	• MS Word • Zoom • WhatsApp • Project Plan 365 • JIRA	• Requirements documentation • scope definition
System design	1 Oct	31 Oct	31 days	50,638.28	• System architect • Database designer • UI/UX designer	• Zoom • WhatsApp • Figma • Project Plan 365	• Function specs • Database design • Interface design
System implemen-tation	1 Nov	31 Jan	92 days	106,356.87	• Software developer • Database administrato r	• Zoom • GitHub • Android Studio • Flutter • Firebase • Project Plan 365	• Implemented database • Working website with all requirements implemented
System testing	1 Feb	1 Mar 2025	29 days	28,560.52	• Tester • Developer	• MS Word • WhatsApp	• Report of test results

						• Android Studio • Flutter • Project Plan 365	• Website after resolving all issues

To estimate costs for each phase of the project, we considered several key factors:

1. Personnel Salaries: Costs were determined by calculating a typical monthly salary for each role involved in the project.
2. Tool Costs: Most tools were free, the only tool incurring a cost was Project Plan 365, which is priced at 224.775 SAR for a six-month subscription. To allocate this cost across the five phases of the project, we divided the total by the number of phases, assigning approximately 44.95 SAR to each phase. While this method provides a rough allocation, it serves as a practical estimate, given that the precise impact of tool costs is relatively minor compared to personnel expenses in the overall budget.

By focusing on these factors, we estimated costs for each phase of the project as follows:

Project Initiation:

Project Sponsor estimated salary: 18,500 SAR/month

Daily rate: 18,500 SAR / 30 days = 616.67 SAR (approximately)

Personnel cost: SAR 616.67 × 14 days = 8,633.38 SAR

Tool cost: 44.95 SAR

Total Cost for Project Initiation: **8,678.33 SAR**

System Analysis:

System Analyst estimated salary: 10,250 SAR/month

Daily rate: 10,250 SAR / 30 days = 341.67 SAR

Personnel cost: 341.67 SAR × 16 days = 5,466.67 SAR

Stakeholder estimated salary: 10,000 SAR/month

Daily rate: 10,000 SAR / 30 days = 333.33 SAR

Personnel cost: SAR 333.33 × 16 days = 5,333.33 SAR

Tool cost: 44.95 SAR

Total Cost for System Analysis: **10,844.95 SAR**

System Design:

System Architect estimated salary: 27,000 SAR/month

Daily rate: SAR 27,000 / 30 = 900 SAR

Personnel cost: SAR 900 × 31 days = 27,900 SAR

Database Designer estimated salary: 9,000 SAR/month

Daily rate: SAR 9,000 / 30 = 300 SAR

Personnel cost: SAR 300 × 31 days = 9,300 SAR

UI/UX Designer estimated salary: 13,000 SAR /month

Daily rate: SAR 13,000 / 30 = 433.33 SAR

Personnel cost: SAR 433.33 × 31 days = 13,393.33 SAR

Tool cost: 44.95 SAR

Total Cost for System Design: **50,638.28 SAR**

System Implementation:

Software Developer estimated salary: 18,000 SAR/month

Daily rate: SAR 18,000 / 30 = 600 SAR

Personnel cost: SAR 600 × 92 days = 55,200 SAR

Database Administrator estimated salary: 16,667 SAR /month

Daily rate: SAR 16,667 / 30 = 555.56 SAR

Personnel cost: SAR 555.56 × 92 days = 51,111.92 SAR

Tool cost: 44.95 SAR

Total Cost for System Implementation: **106,356.87 SAR**

System Testing:

Tester estimated salary: 11,500 SAR/month

Daily rate: SAR 11,500 / 30 = 383.33 SAR

Personnel cost: SAR 383.33 × 29 days = 11,115.57 SAR

Software Developer: 18,000 SAR/month

Daily rate: SAR 18,000 / 30 = 600 SAR

Personnel cost: SAR 600 × 29 days = 17,400 SAR

Tool cost: 44.95 SAR

Total Cost for System Testing: **28,560.52 SAR**

- ## WBDS Chart

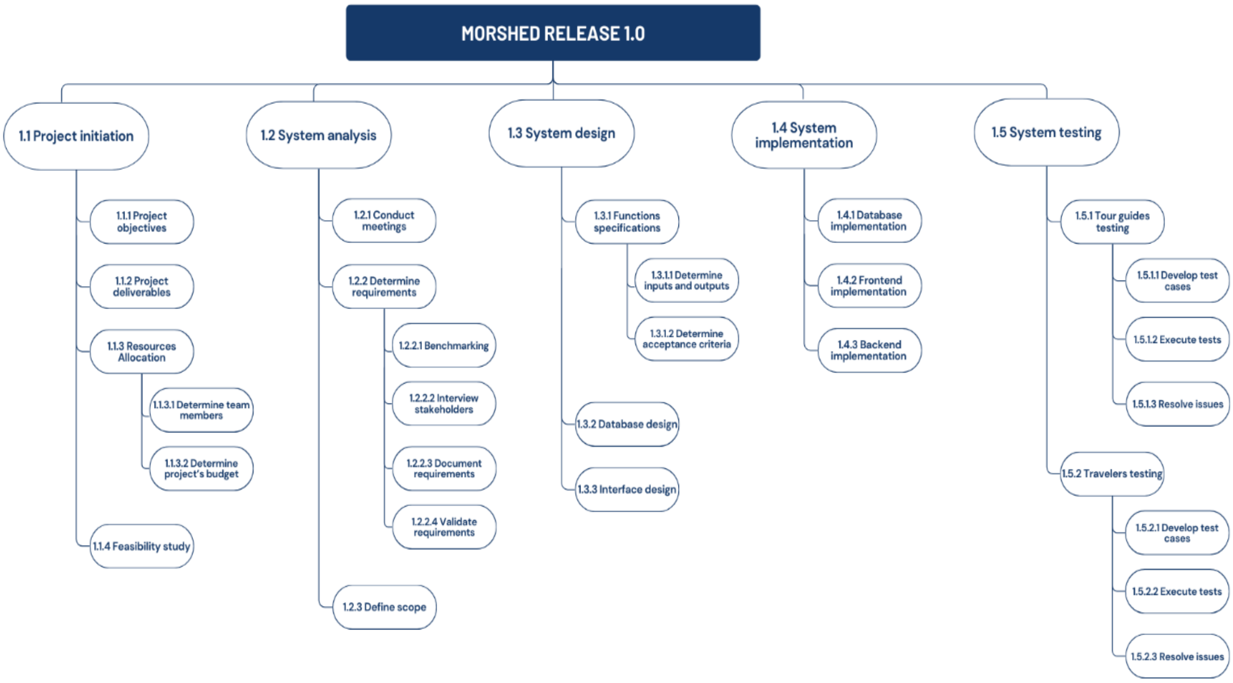
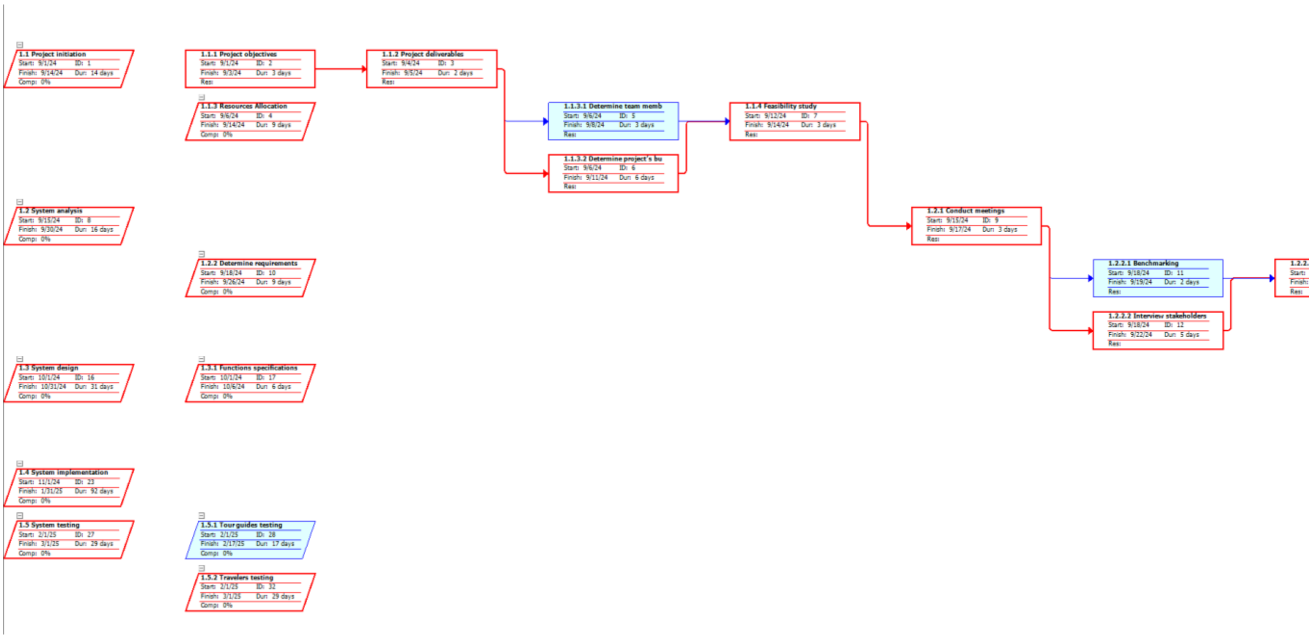
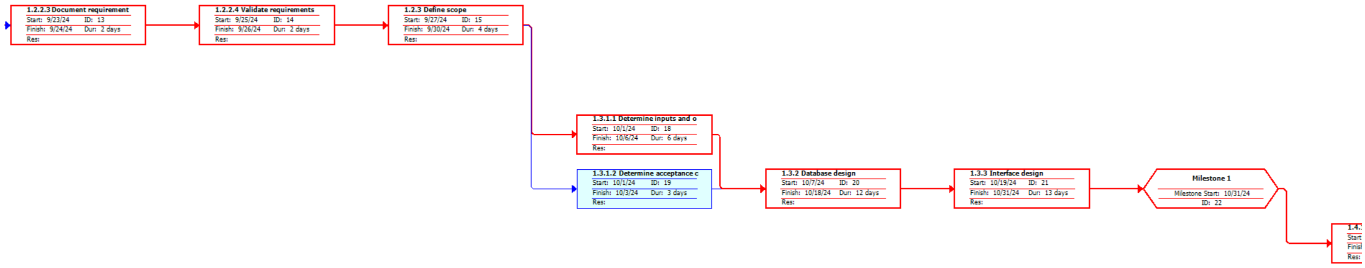

MORSHED RELEASE 1.0
1.1 Project initiation
1.1.1 Project objectives
1.1.2 Project deliverables
1.1.3 Resources Allocation
1.1.3.1 Determine team members
1.1.3.2 Determine project's budget
1.1.4 Feasibility study
1.2 System analysis
1.2.1 Conduct meetings
1.2.2 Determine requirements
1.2.2.1 Benchmarking
1.2.2.2 Interview stakeholders
1.2.2.3 Document requirements
1.2.2.4 Validate requirements
1.2.3 Define scope
1.3 System design
1.3.1 Functions specifications
1.3.1.1 Determine inputs and outputs
1.3.1.2 Determine acceptance criteria
1.3.2 Database design
1.3.3 Interface design
1.4 System implementation
1.4.1 Database implementation
1.4.2 Frontend implementation
1.4.3 Backend implementation
1.5 System testing
1.5.1 Tour guides testing
1.5.1.1 Develop test cases
1.5.1.2 Execute tests
1.5.1.3 Resolve issues
1.5.2 Travelers testing
1.5.2.1 Develop test cases
1.5.2.2 Execute tests
1.5.2.3 Resolve issues

FIGURE 1:WBDS CHART

5. TECHNICAL OUTCOMES AND CHARTS

5.1 NETWORK DIAGRAM WITH CRITICAL PATH


1.1 Project initiation
Start: 9/1/24 ID: 1
Finish: 9/14/24 Dur: 14 days
Comp: 0%
1.1.1 Project objectives
Start: 9/1/24 ID: 2
Finish: 9/3/24 Dur: 3 days
Res:
1.1.2 Project deliverables
Start: 9/4/24 ID: 3
Finish: 9/5/24 Dur: 2 days
Res:
1.1.3 Resources Allocation
Start: 9/6/24 ID: 4
Finish: 9/14/24 Dur: 9 days
Comp: 0%
1.1.3.1 Determine team memb
Start: 9/6/24 ID: 5
Finish: 9/8/24 Dur: 3 days
Res:
1.1.3.2 Determine project's bu
Start: 9/6/24 ID: 6
Finish: 9/11/24 Dur: 6 days
Res:
1.1.4 Feasibility study
Start: 9/12/24 ID: 7
Finish: 9/14/24 Dur: 3 days
Res:
1.2 System analysis
Start: 9/15/24 ID: 8
Finish: 9/30/24 Dur: 16 days
Comp: 0%
1.2.1 Conduct meetings
Start: 9/15/24 ID: 9
Finish: 9/17/24 Dur: 3 days
Res:
1.2.2 Determine requirements
Start: 9/18/24 ID: 10
Finish: 9/26/24 Dur: 9 days
Comp: 0%
1.2.2.1 Benchmarking
Start: 9/18/24 ID: 11
Finish: 9/19/24 Dur: 2 days
Res:
1.2.2.2 Interview stakeholders
Start: 9/18/24 ID: 12
Finish: 9/22/24 Dur: 5 days
Res:
1.2.2.
Start:
Finish:
Res:
1.3 System design
Start: 10/1/24 ID: 16
Finish: 10/31/24 Dur: 31 days
Comp: 0%
1.3.1 Functions specifications
Start: 10/1/24 ID: 17
Finish: 10/6/24 Dur: 6 days
Comp: 0%
1.4 System implementation
Start: 11/1/24 ID: 23
Finish: 1/31/25 Dur: 92 days
Comp: 0%
1.5 System testing
Start: 2/1/25 ID: 27
Finish: 3/1/25 Dur: 29 days
Comp: 0%
1.5.1 Tour guides testing
Start: 2/1/25 ID: 28
Finish: 2/17/25 Dur: 17 days
Comp: 0%
1.5.2 Travelers testing
Start: 2/1/25 ID: 32
Finish: 3/1/25 Dur: 29 days
Comp: 0%


1.2.2.3 Document requirement
Start: 9/23/24 ID: 13
Finish: 9/24/24 Dur: 2 days
Res:
1.2.2.4 Validate requirements
Start: 9/25/24 ID: 14
Finish: 9/26/24 Dur: 2 days
Res:
1.2.3 Define scope
Start: 9/27/24 ID: 15
Finish: 9/30/24 Dur: 4 days
Res:
1.3.1.1 Determine inputs and o
Start: 10/1/24 ID: 18
Finish: 10/6/24 Dur: 6 days
Res:
1.3.1.2 Determine acceptance c
Start: 10/1/24 ID: 19
Finish: 10/3/24 Dur: 3 days
Res:
1.3.2 Database design
Start: 10/7/24 ID: 20
Finish: 10/18/24 Dur: 12 days
Res:
1.3.3 Interface design
Start: 10/19/24 ID: 21
Finish: 10/31/24 Dur: 13 days
Res:
Milestone 1
Milestone Start: 10/31/24
ID: 22
1.4.
Start
Finisl
Res:

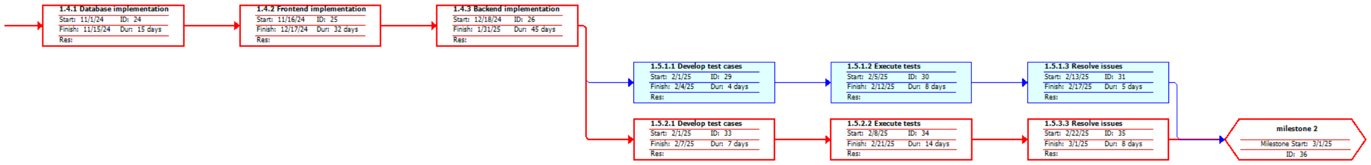

1.4.1 Database implementation
Start: 11/1/24 ID: 24
Finish: 11/15/24 Dur: 15 days
Res:
1.4.2 Frontend implementation
Start: 11/16/24 ID: 25
Finish: 12/17/24 Dur: 32 days
Res:
1.4.3 Backend implementation
Start: 12/18/24 ID: 26
Finish: 1/31/25 Dur: 45 days
Res:
1.5.1.1 Develop test cases
Start: 2/1/25 ID: 29
Finish: 2/4/25 Dur: 4 days
Res:
1.5.1.2 Execute tests
Start: 2/5/25 ID: 30
Finish: 2/12/25 Dur: 8 days
Res:
1.5.1.3 Resolve issues
Start: 2/13/25 ID: 31
Finish: 2/17/25 Dur: 5 days
Res:
1.5.2.1 Develop test cases
Start: 2/1/25 ID: 33
Finish: 2/7/25 Dur: 7 days
Res:
1.5.2.2 Execute tests
Start: 2/8/25 ID: 34
Finish: 2/21/25 Dur: 14 days
Res:
1.5.3.3 Resolve issues
Start: 2/22/25 ID: 35
Finish: 3/1/25 Dur: 8 days
Res:
milestone 2
Milestone Start: 3/1/25
ID: 36

FIGURE 2:NETWORK DIAGRAM

As shown in the figure [2], the network diagram illustrates the sequence of project phases for the Morshed website. It highlights the key stages of the project, from initiation to testing, along with their respective timelines and resources. The diagram provides a comprehensive view of dependencies between tasks, enabling effective project management and ensuring timely completion.

- **Milestones**

The diagram marks **two milestones**:

- ○ **Milestone 1** on **October 31, 2024**, marking the completion of the design phase, including tasks such as "Database design" and "Interface design." This milestone serves as a checkpoint to review the progress of system development and ensure readiness for the implementation phase.
- ○ **Milestone 2** on **March 1, 2025**, signifying the completion of all project activities, including testing and issue resolution. This milestone indicates the project's readiness for closure or deployment and acts as a final checkpoint to assess whether all objectives have been met.

- **Dependencies**:

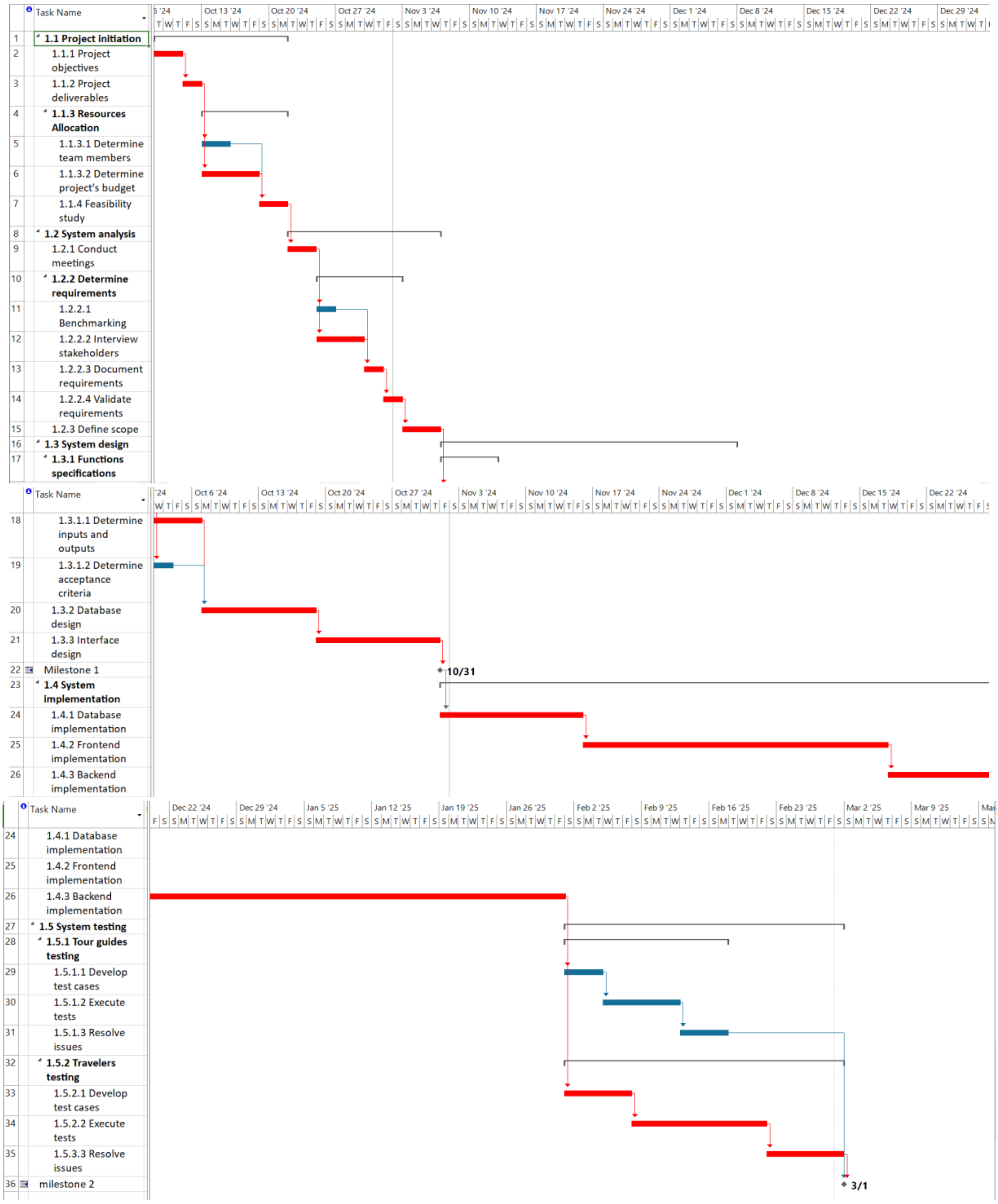
The network diagram includes various task dependencies that ensure each phase of the Morshed website project flows smoothly from one activity to the next. The primary type of dependency used is **Finish-to-Start (FS)**, where a task must be completed before the following task can begin. This dependency type is evident across major phases, including the transition from Project Initiation to System Analysis, and from System Design to System Implementation. In some instances, **Start-to-Start (SS)** and **Finish-to-Finish (FF)** dependencies are applied, particularly during the testing phase, where tasks like "Develop test cases" and "Execute tests" may need to progress concurrently or conclude around the same time to maintain project continuity and alignment.

- **Critical Path**:

The critical path is the longest path through the network diagram and determines the minimum project duration. Delays in any task on this path will directly delay the project's completion.

  - The critical path likely goes through major tasks like "System analysis," "System design," "Database and Backend implementation," and "System testing."
  - Tasks on the critical path appear in red in the diagram, and any delay in these will impact the project’s end date.

1.1. A COMPLETE GANTT CHART



	Task Name
1	1.1 Project initiation
2	1.1.1 Project objectives
3	1.1.2 Project deliverables
4	1.1.3 Resources Allocation
5	1.1.3.1 Determine team members
6	1.1.3.2 Determine project's budget
7	1.1.4 Feasibility study
8	1.2 System analysis
9	1.2.1 Conduct meetings
10	1.2.2 Determine requirements
11	1.2.2.1 Benchmarking
12	1.2.2.2 Interview stakeholders
13	1.2.2.3 Document requirements
14	1.2.2.4 Validate requirements
15	1.2.3 Define scope
16	1.3 System design
17	1.3.1 Functions specifications

	Task Name
18	1.3.1.1 Determine inputs and outputs
19	1.3.1.2 Determine acceptance criteria
20	1.3.2 Database design
21	1.3.3 Interface design
22	Milestone 1
23	1.4 System implementation
24	1.4.1 Database implementation
25	1.4.2 Frontend implementation
26	1.4.3 Backend implementation

	Task Name
24	1.4.1 Database implementation
25	1.4.2 Frontend implementation
26	1.4.3 Backend implementation
27	1.5 System testing
28	1.5.1 Tour guides testing
29	1.5.1.1 Develop test cases
30	1.5.1.2 Execute tests
31	1.5.1.3 Resolve issues
32	1.5.2 Travelers testing
33	1.5.2.1 Develop test cases
34	1.5.2.2 Execute tests
35	1.5.3.3 Resolve issues
36	milestone 2

FIGURE 3: A COMPLETE GANTT CHART

Figure [3] presents a detailed Gantt Chart for the Morshed project, illustrating the timeline across workdays and weekends to provide a view of project progress on a calendar. The chart visualizes key tasks, their start and end dates, and task dependencies, allowing the project's flow to be easily understood.

The project is organized into five main phases:

Project Initiation and **System Analysis**: These foundational phases are scheduled for completion within September 2024, focusing on setting objectives, gathering requirements, and allocating resources.

System Design: Beginning in early October and continuing through the end of the month, this phase covers specifications, database design, and interface planning. The completion of this phase marks **Milestone 1** on October 31, 2024.

System Implementation: Starting in November, the implementation phase includes database, frontend, and backend development. This phase spans through to the end of January 2025.

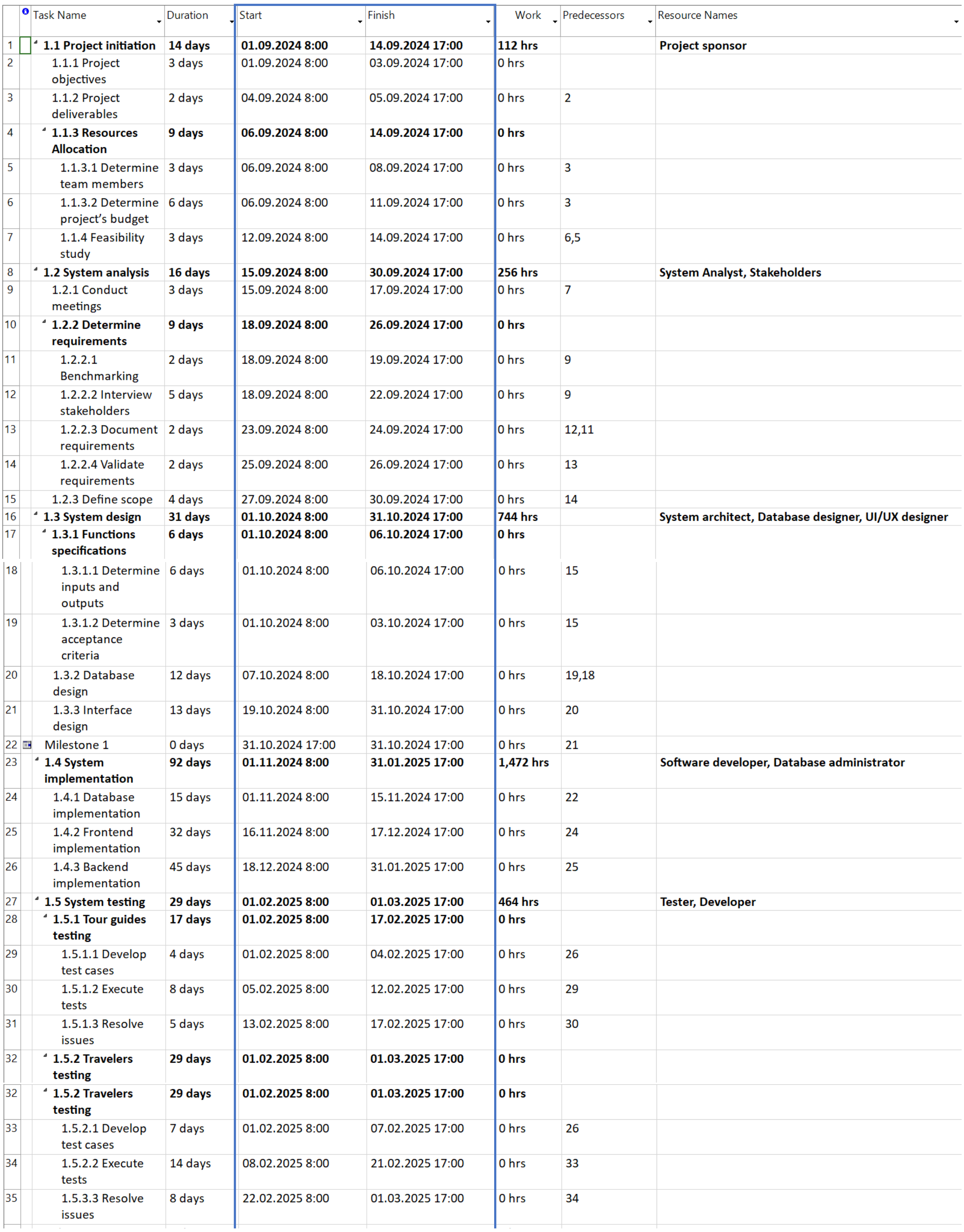
System Testing: Scheduled from February through early March 2025, the testing phase encompasses developing test cases, executing them, and resolving any issues for both tour guides and travelers.

Project Completion: Marked by **Milestone 2** on March 1, 2025, this signifies the end of system testing and the official completion of the Morshed project.

This Gantt Chart not only tracks individual tasks but also emphasizes the chronological flow and interdependencies critical to achieving the project milestones.

5.3. PROJECT TIME LINE

Figure 4 illustrates the project timeline.
note: the duration start and finish dates include weekend and weekdays.

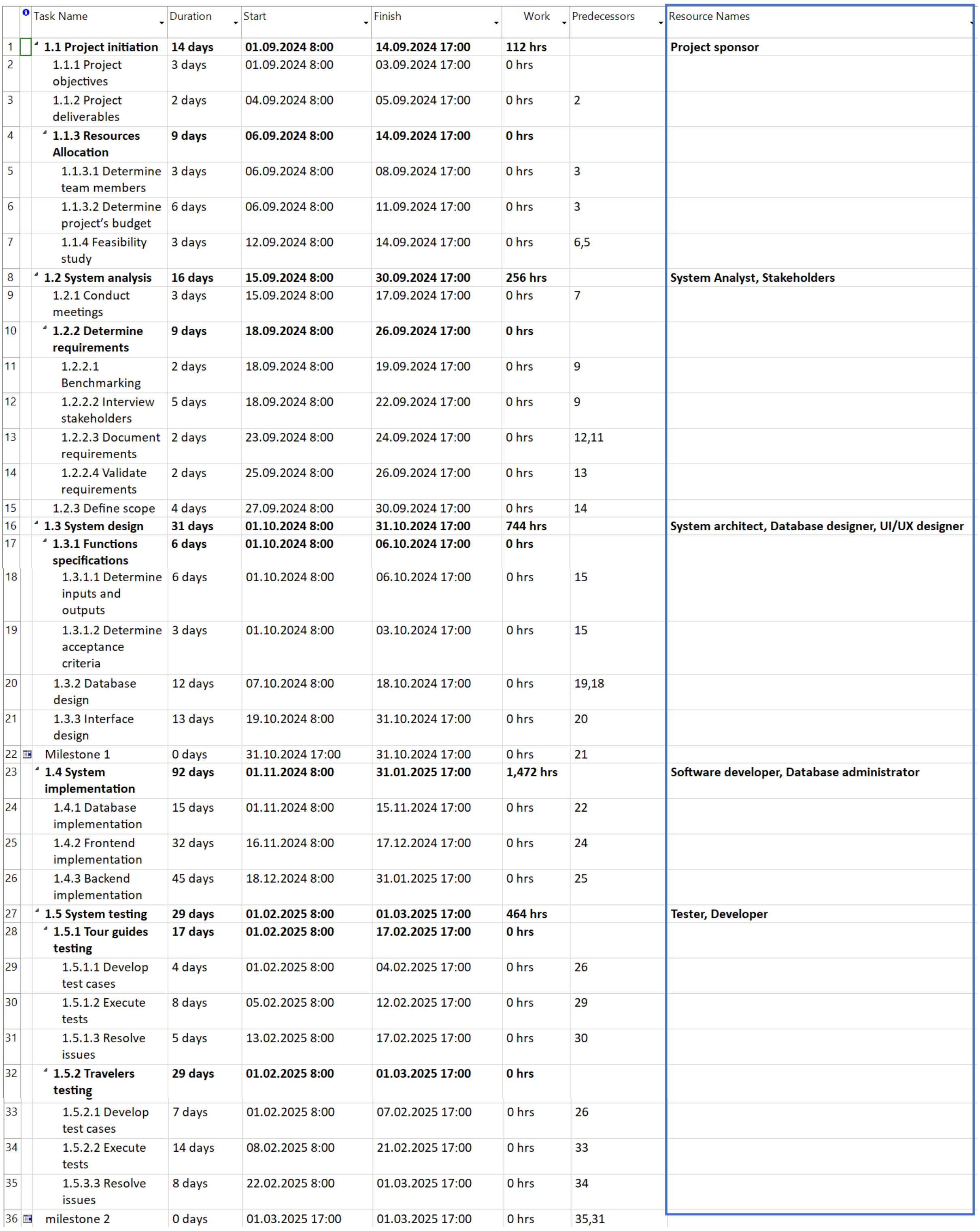


	Task Name	Duration	Start	Finish	Work	Predecessors	Resource Names
1	**1.1 Project initiation**	**14 days**	**01.09.2024 8:00**	**14.09.2024 17:00**	**112 hrs**		**Project sponsor**
2	1.1.1 Project objectives	3 days	01.09.2024 8:00	03.09.2024 17:00	0 hrs		
3	1.1.2 Project deliverables	2 days	04.09.2024 8:00	05.09.2024 17:00	0 hrs	2	
4	**1.1.3 Resources Allocation**	**9 days**	**06.09.2024 8:00**	**14.09.2024 17:00**	**0 hrs**		
5	1.1.3.1 Determine team members	3 days	06.09.2024 8:00	08.09.2024 17:00	0 hrs	3	
6	1.1.3.2 Determine project's budget	6 days	06.09.2024 8:00	11.09.2024 17:00	0 hrs	3	
7	1.1.4 Feasibility study	3 days	12.09.2024 8:00	14.09.2024 17:00	0 hrs	6,5	
8	**1.2 System analysis**	**16 days**	**15.09.2024 8:00**	**30.09.2024 17:00**	**256 hrs**		**System Analyst, Stakeholders**
9	1.2.1 Conduct meetings	3 days	15.09.2024 8:00	17.09.2024 17:00	0 hrs	7	
10	**1.2.2 Determine requirements**	**9 days**	**18.09.2024 8:00**	**26.09.2024 17:00**	**0 hrs**		
11	1.2.2.1 Benchmarking	2 days	18.09.2024 8:00	19.09.2024 17:00	0 hrs	9	
12	1.2.2.2 Interview stakeholders	5 days	18.09.2024 8:00	22.09.2024 17:00	0 hrs	9	
13	1.2.2.3 Document requirements	2 days	23.09.2024 8:00	24.09.2024 17:00	0 hrs	12,11	
14	1.2.2.4 Validate requirements	2 days	25.09.2024 8:00	26.09.2024 17:00	0 hrs	13	
15	1.2.3 Define scope	4 days	27.09.2024 8:00	30.09.2024 17:00	0 hrs	14	
16	**1.3 System design**	**31 days**	**01.10.2024 8:00**	**31.10.2024 17:00**	**744 hrs**		**System architect, Database designer, UI/UX designer**
17	**1.3.1 Functions specifications**	**6 days**	**01.10.2024 8:00**	**06.10.2024 17:00**	**0 hrs**		
18	1.3.1.1 Determine inputs and outputs	6 days	01.10.2024 8:00	06.10.2024 17:00	0 hrs	15	
19	1.3.1.2 Determine acceptance criteria	3 days	01.10.2024 8:00	03.10.2024 17:00	0 hrs	15	
20	1.3.2 Database design	12 days	07.10.2024 8:00	18.10.2024 17:00	0 hrs	19,18	
21	1.3.3 Interface design	13 days	19.10.2024 8:00	31.10.2024 17:00	0 hrs	20	
22	Milestone 1	0 days	31.10.2024 17:00	31.10.2024 17:00	0 hrs	21	
23	**1.4 System implementation**	**92 days**	**01.11.2024 8:00**	**31.01.2025 17:00**	**1,472 hrs**		**Software developer, Database administrator**
24	1.4.1 Database implementation	15 days	01.11.2024 8:00	15.11.2024 17:00	0 hrs	22	
25	1.4.2 Frontend implementation	32 days	16.11.2024 8:00	17.12.2024 17:00	0 hrs	24	
26	1.4.3 Backend implementation	45 days	18.12.2024 8:00	31.01.2025 17:00	0 hrs	25	
27	**1.5 System testing**	**29 days**	**01.02.2025 8:00**	**01.03.2025 17:00**	**464 hrs**		**Tester, Developer**
28	**1.5.1 Tour guides testing**	**17 days**	**01.02.2025 8:00**	**17.02.2025 17:00**	**0 hrs**		
29	1.5.1.1 Develop test cases	4 days	01.02.2025 8:00	04.02.2025 17:00	0 hrs	26	
30	1.5.1.2 Execute tests	8 days	05.02.2025 8:00	12.02.2025 17:00	0 hrs	29	
31	1.5.1.3 Resolve issues	5 days	13.02.2025 8:00	17.02.2025 17:00	0 hrs	30	
32	**1.5.2 Travelers testing**	**29 days**	**01.02.2025 8:00**	**01.03.2025 17:00**	**0 hrs**		
32	**1.5.2 Travelers testing**	**29 days**	**01.02.2025 8:00**	**01.03.2025 17:00**	**0 hrs**		
33	1.5.2.1 Develop test cases	7 days	01.02.2025 8:00	07.02.2025 17:00	0 hrs	26	
34	1.5.2.2 Execute tests	14 days	08.02.2025 8:00	21.02.2025 17:00	0 hrs	33	
35	1.5.3.3 Resolve issues	8 days	22.02.2025 8:00	01.03.2025 17:00	0 hrs	34	

FIGURE 4:PROJECT TIMELINE

5.4. Task allocation schedule

Figure 5 illustrates the task allocation schedule of the project.

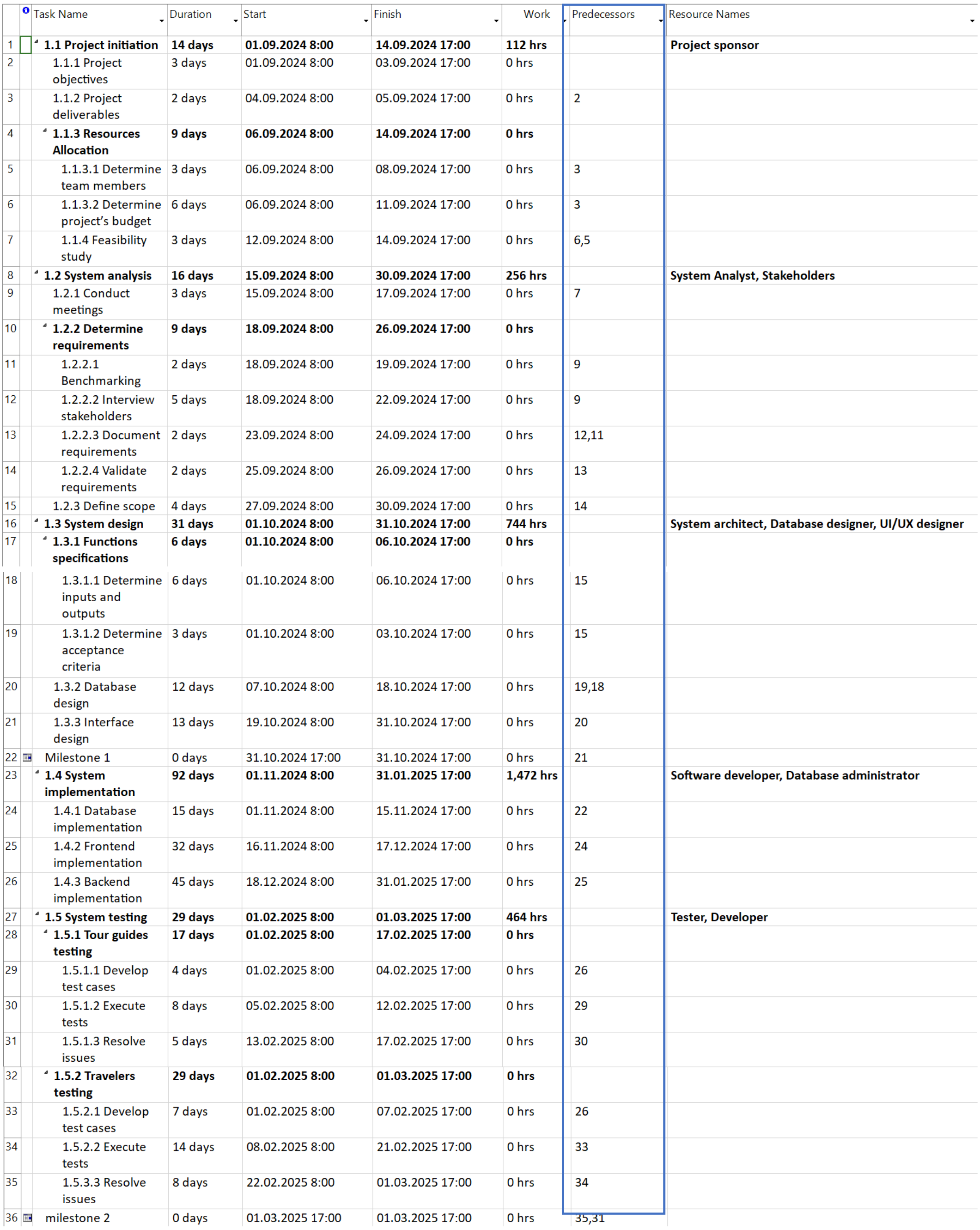


	Task Name	Duration	Start	Finish	Work	Predecessors	Resource Names
1	**1.1 Project initiation**	**14 days**	**01.09.2024 8:00**	**14.09.2024 17:00**	**112 hrs**		**Project sponsor**
2	1.1.1 Project objectives	3 days	01.09.2024 8:00	03.09.2024 17:00	0 hrs		
3	1.1.2 Project deliverables	2 days	04.09.2024 8:00	05.09.2024 17:00	0 hrs	2	
4	**1.1.3 Resources Allocation**	**9 days**	**06.09.2024 8:00**	**14.09.2024 17:00**	**0 hrs**		
5	1.1.3.1 Determine team members	3 days	06.09.2024 8:00	08.09.2024 17:00	0 hrs	3	
6	1.1.3.2 Determine project's budget	6 days	06.09.2024 8:00	11.09.2024 17:00	0 hrs	3	
7	1.1.4 Feasibility study	3 days	12.09.2024 8:00	14.09.2024 17:00	0 hrs	6,5	
8	**1.2 System analysis**	**16 days**	**15.09.2024 8:00**	**30.09.2024 17:00**	**256 hrs**		**System Analyst, Stakeholders**
9	1.2.1 Conduct meetings	3 days	15.09.2024 8:00	17.09.2024 17:00	0 hrs	7	
10	**1.2.2 Determine requirements**	**9 days**	**18.09.2024 8:00**	**26.09.2024 17:00**	**0 hrs**		
11	1.2.2.1 Benchmarking	2 days	18.09.2024 8:00	19.09.2024 17:00	0 hrs	9	
12	1.2.2.2 Interview stakeholders	5 days	18.09.2024 8:00	22.09.2024 17:00	0 hrs	9	
13	1.2.2.3 Document requirements	2 days	23.09.2024 8:00	24.09.2024 17:00	0 hrs	12,11	
14	1.2.2.4 Validate requirements	2 days	25.09.2024 8:00	26.09.2024 17:00	0 hrs	13	
15	1.2.3 Define scope	4 days	27.09.2024 8:00	30.09.2024 17:00	0 hrs	14	
16	**1.3 System design**	**31 days**	**01.10.2024 8:00**	**31.10.2024 17:00**	**744 hrs**		**System architect, Database designer, UI/UX designer**
17	**1.3.1 Functions specifications**	**6 days**	**01.10.2024 8:00**	**06.10.2024 17:00**	**0 hrs**		
18	1.3.1.1 Determine inputs and outputs	6 days	01.10.2024 8:00	06.10.2024 17:00	0 hrs	15	
19	1.3.1.2 Determine acceptance criteria	3 days	01.10.2024 8:00	03.10.2024 17:00	0 hrs	15	
20	1.3.2 Database design	12 days	07.10.2024 8:00	18.10.2024 17:00	0 hrs	19,18	
21	1.3.3 Interface design	13 days	19.10.2024 8:00	31.10.2024 17:00	0 hrs	20	
22	Milestone 1	0 days	31.10.2024 17:00	31.10.2024 17:00	0 hrs	21	
23	**1.4 System implementation**	**92 days**	**01.11.2024 8:00**	**31.01.2025 17:00**	**1,472 hrs**		**Software developer, Database administrator**
24	1.4.1 Database implementation	15 days	01.11.2024 8:00	15.11.2024 17:00	0 hrs	22	
25	1.4.2 Frontend implementation	32 days	16.11.2024 8:00	17.12.2024 17:00	0 hrs	24	
26	1.4.3 Backend implementation	45 days	18.12.2024 8:00	31.01.2025 17:00	0 hrs	25	
27	**1.5 System testing**	**29 days**	**01.02.2025 8:00**	**01.03.2025 17:00**	**464 hrs**		**Tester, Developer**
28	**1.5.1 Tour guides testing**	**17 days**	**01.02.2025 8:00**	**17.02.2025 17:00**	**0 hrs**		
29	1.5.1.1 Develop test cases	4 days	01.02.2025 8:00	04.02.2025 17:00	0 hrs	26	
30	1.5.1.2 Execute tests	8 days	05.02.2025 8:00	12.02.2025 17:00	0 hrs	29	
31	1.5.1.3 Resolve issues	5 days	13.02.2025 8:00	17.02.2025 17:00	0 hrs	30	
32	**1.5.2 Travelers testing**	**29 days**	**01.02.2025 8:00**	**01.03.2025 17:00**	**0 hrs**		
33	1.5.2.1 Develop test cases	7 days	01.02.2025 8:00	07.02.2025 17:00	0 hrs	26	
34	1.5.2.2 Execute tests	14 days	08.02.2025 8:00	21.02.2025 17:00	0 hrs	33	
35	1.5.3.3 Resolve issues	8 days	22.02.2025 8:00	01.03.2025 17:00	0 hrs	34	
36	milestone 2	0 days	01.03.2025 17:00	01.03.2025 17:00	0 hrs	35,31	

FIGURE 5:TASK ALLOCATION SCHEDULE

5.5. Task dependencies table

Figure [6] illustrates the task dependencies of the project



	Task Name	Duration	Start	Finish	Work	Predecessors	Resource Names
1	**1.1 Project initiation**	**14 days**	**01.09.2024 8:00**	**14.09.2024 17:00**	**112 hrs**		**Project sponsor**
2	1.1.1 Project objectives	3 days	01.09.2024 8:00	03.09.2024 17:00	0 hrs		
3	1.1.2 Project deliverables	2 days	04.09.2024 8:00	05.09.2024 17:00	0 hrs	2	
4	**1.1.3 Resources Allocation**	**9 days**	**06.09.2024 8:00**	**14.09.2024 17:00**	**0 hrs**		
5	1.1.3.1 Determine team members	3 days	06.09.2024 8:00	08.09.2024 17:00	0 hrs	3	
6	1.1.3.2 Determine project's budget	6 days	06.09.2024 8:00	11.09.2024 17:00	0 hrs	3	
7	1.1.4 Feasibility study	3 days	12.09.2024 8:00	14.09.2024 17:00	0 hrs	6,5	
8	**1.2 System analysis**	**16 days**	**15.09.2024 8:00**	**30.09.2024 17:00**	**256 hrs**		**System Analyst, Stakeholders**
9	1.2.1 Conduct meetings	3 days	15.09.2024 8:00	17.09.2024 17:00	0 hrs	7	
10	**1.2.2 Determine requirements**	**9 days**	**18.09.2024 8:00**	**26.09.2024 17:00**	**0 hrs**		
11	1.2.2.1 Benchmarking	2 days	18.09.2024 8:00	19.09.2024 17:00	0 hrs	9	
12	1.2.2.2 Interview stakeholders	5 days	18.09.2024 8:00	22.09.2024 17:00	0 hrs	9	
13	1.2.2.3 Document requirements	2 days	23.09.2024 8:00	24.09.2024 17:00	0 hrs	12,11	
14	1.2.2.4 Validate requirements	2 days	25.09.2024 8:00	26.09.2024 17:00	0 hrs	13	
15	1.2.3 Define scope	4 days	27.09.2024 8:00	30.09.2024 17:00	0 hrs	14	
16	**1.3 System design**	**31 days**	**01.10.2024 8:00**	**31.10.2024 17:00**	**744 hrs**		**System architect, Database designer, UI/UX designer**
17	**1.3.1 Functions specifications**	**6 days**	**01.10.2024 8:00**	**06.10.2024 17:00**	**0 hrs**		
18	1.3.1.1 Determine inputs and outputs	6 days	01.10.2024 8:00	06.10.2024 17:00	0 hrs	15	
19	1.3.1.2 Determine acceptance criteria	3 days	01.10.2024 8:00	03.10.2024 17:00	0 hrs	15	
20	1.3.2 Database design	12 days	07.10.2024 8:00	18.10.2024 17:00	0 hrs	19,18	
21	1.3.3 Interface design	13 days	19.10.2024 8:00	31.10.2024 17:00	0 hrs	20	
22	Milestone 1	0 days	31.10.2024 17:00	31.10.2024 17:00	0 hrs	21	
23	**1.4 System implementation**	**92 days**	**01.11.2024 8:00**	**31.01.2025 17:00**	**1,472 hrs**		**Software developer, Database administrator**
24	1.4.1 Database implementation	15 days	01.11.2024 8:00	15.11.2024 17:00	0 hrs	22	
25	1.4.2 Frontend implementation	32 days	16.11.2024 8:00	17.12.2024 17:00	0 hrs	24	
26	1.4.3 Backend implementation	45 days	18.12.2024 8:00	31.01.2025 17:00	0 hrs	25	
27	**1.5 System testing**	**29 days**	**01.02.2025 8:00**	**01.03.2025 17:00**	**464 hrs**		**Tester, Developer**
28	**1.5.1 Tour guides testing**	**17 days**	**01.02.2025 8:00**	**17.02.2025 17:00**	**0 hrs**		
29	1.5.1.1 Develop test cases	4 days	01.02.2025 8:00	04.02.2025 17:00	0 hrs	26	
30	1.5.1.2 Execute tests	8 days	05.02.2025 8:00	12.02.2025 17:00	0 hrs	29	
31	1.5.1.3 Resolve issues	5 days	13.02.2025 8:00	17.02.2025 17:00	0 hrs	30	
32	**1.5.2 Travelers testing**	**29 days**	**01.02.2025 8:00**	**01.03.2025 17:00**	**0 hrs**		
33	1.5.2.1 Develop test cases	7 days	01.02.2025 8:00	07.02.2025 17:00	0 hrs	26	
34	1.5.2.2 Execute tests	14 days	08.02.2025 8:00	21.02.2025 17:00	0 hrs	33	
35	1.5.3.3 Resolve issues	8 days	22.02.2025 8:00	01.03.2025 17:00	0 hrs	34	
36	milestone 2	0 days	01.03.2025 17:00	01.03.2025 17:00	0 hrs	35,31	

FIGURE 6:TASK DEPENDENCIES TABLE

5.6. Hardware and Software Tools

- **Software Tools:**
  - Programming language (Dart).
  - IDEs (NetBeans and Visual Studio).
  - Database management system (MySQL).
  - Zoom.
  - WhatsApp.
  - Microsoft word.
  - Jira.
  - GitHub.
  - Figma.
  - Canva.
  - Google Forms.
  - Project plan 365.

- **Hardware Tools:**
  - Laptops for development and testing.
  - Server to host the web application and database.

- **Human Resources:**
  - **Project Sponsor**: The key individual or group responsible for providing financial resources and overall support for the project. They ensure the project's alignment with business objectives and facilitate high-level decision-making.
  - **System Analyst**: A professional who studies and evaluates the system requirements and designs solutions by analyzing business processes. They act as a bridge between stakeholders and technical teams to ensure system functionality aligns with business needs.
  - **Stakeholder**: Any individual or group who has an interest in or is affected by the project. This includes users, customers, management, and other entities with a vested interest in the project's outcome.
  - **System Architect**: The individual responsible for the overall technical structure of the system. They define the system's architecture, ensure it meets all necessary technical requirements, and guide the development team on technical solutions.
  - **Database Designer**: A specialist who designs and creates the structure of a database. They define how data will be stored, accessed, and managed within the system to ensure efficient data handling and security.

- **UI/UX Designer**: A designer who focuses on creating user-friendly interfaces and ensuring an optimal user experience. They design how the software looks (UI) and feels (UX) to make the system intuitive and accessible for users.
- **Software Developer**: A technical professional who writes, tests, and maintains the code that makes up the software. They implement the system's functionality based on design and technical specifications.
- **Database Administrator**: A professional responsible for the performance, integrity, and security of a database. They ensure that data is efficiently stored, backed up, and accessible while maintaining security protocols.
- **Tester**: An individual who tests the software for bugs, errors, and usability issues. They ensure the software functions as intended and meets all specified requirements before deployment.
- **Developer**: A broad term that can refer to any individual involved in the creation and implementation of software. Developers can specialize in front-end, back-end, or full-stack development, depending on their skills and project needs.

6. Cost estimation

6.1 Development Cost

6.1.1 AFP (Adjusted Function Points)

In estimating the development cost for the Morshed project, we used the Function Point (FP) analysis to evaluate the system’s functional size based on key components like inputs, outputs, files, inquiries, and interfaces. This approach provides a clear estimate of the effort needed by analyzing these components based on their quantity and complexity, then adjusting for additional influencing factors.

- **Identify Function Points for Each Component**
  Each component of the Morshed system, as specified by user requirements, has been assigned function points based on its complexity and functionality. Function points provide a standardized way to quantify the development effort associated with each component. The points for each component are as follows:

  - **Inputs**: Represent any user inputs, like registration and booking forms that users fill out to interact with the system. The number of inputs is counted based on these forms and processes. For Morshed, we identified **25 input points** based on the types and complexity of forms.

  - **Outputs**: Include any data outputs generated by the system in response to user inputs, such as tour details, booking confirmations, or search results. For Morshed, **35 output points** were counted based on the information generated by the system.

  - **Files**: Refer to internal or external databases that store information used by the system, such as tour data, user data, or chat histories. For Morshed, **20 file points** were calculated based on the data storage requirements.

  - **Inquiries**: Are any user-initiated searches or requests for information, such as filtering tours or querying availability. For Morshed, **15 inquiry points** were calculated, corresponding to anticipated search and filter functionality.

  - **Interfaces**: Count the connections with external systems, such as payment gateways or external authentication providers. For Morshed, **5 interface points** were identified based on integration with external systems.

- **Sum the Base Function Points**

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

  Total Function Points=Number of input FPs + Number of output FPs + Number of file FPs + Number of inquiry FPs +Number of interface FPs

  Using Morshed's identified points:

Total Base Function Points = 25 + 35 + 20 + 15 + 5 = 100 FPs

- **Apply Adjustment Factor**

  For this project, an Adjustment Factor of 1.2 was chosen based on the system's complexity and specific requirements. This factor reflects considerations such as high-performance requirements, extensive data handling, and user efficiency.

- **Calculate Adjusted Function Points (AFP)**

  The Adjusted Function Points are then calculated by multiplying the Total Base Function Points by the Adjustment Factor:

  Adjusted Function Points (AFP) = Total Base Function Points × Adjustment Factor

  AFP= 100 × 1.2 = 120 FBs

6.1.2 ASLOC (NUMBER OF LINES OF GENERATED CODE)

Adjusted Source Lines of Code (ASLOC) is a metric used to estimate the amount of code required to implement a system's functionality. It's derived from the Adjusted Function Points (AFP), which, as we calculated in the previous section, quantifies a project's size based on specific functional requirements such as inputs, outputs, files, inquiries, and interfaces. The ASLOC metric is calculated by multiplying the AFP by the Lines of Code per Function Point (LOC/FP), a ratio that varies depending on the programming language used and the project's complexity. For Morshed, developed using Dart, we chose a ratio of 50 LOC/FP based on comparable languages like Java and JavaScript, which have ratios around 47-53 LOC/FP [1]. This estimate aligns with Dart's efficiency in creating user interfaces and handling real-time communication, similar to Morshed's requirements.

To estimate the total lines of code, we use the formula:

ASLOC = Adjusted Function Points × LOC/FP

Applying this to Morshed, with an AFP of 120 and LOC/FP of 50, we get:

ASLOC = 120 × 50 = 6000 lines of code

The resulting 6000 lines of code estimate represents the total amount of coding needed to meet the system's functional requirements. This number plays a critical role in project planning and budgeting, as it provides insight into the developer effort required, helps project managers allocate resources, and allows for accurate cost and schedule estimation. Estimating ASLOC provides a structured and reliable approach to understanding the project's coding scope, ensuring the resources align closely with functional demands.

6.1.3 NUMBER OF STAFF -MONTH

Estimating Staff-Months and Employee-Months is essential to determine the human resources needed for project completion within the scheduled timeframe. Staff-Months reflect the total workload effort required, while Employee-Months focus on the number of employees needed over the project's duration. This calculation depends on the ASLOC (Adjusted Source Lines of Code) estimate from the previous section, divided by a productivity rate. For Morshed, we apply an average productivity rate of 400 lines of code per developer per month, which is commonly referenced as a standard for high-level languages like Dart, JavaScript, and Python. According to Capers Jones, software productivity benchmarks for high-level languages typically range from 300-500 lines of code per month, depending on the complexity and specific environment of the project [2].

Calculating Staff-Months and Employee-Months:

- **Staff-Months Calculation**: The formula to calculate the total Staff-Months required is:

$$\text{Number of Staff} - \text{Months} = \frac{\text{ASLOC}}{\text{Productivity Rate}}$$

For the Morshed project, with an ASLOC of 6000 lines of code and a productivity rate of 400 lines per developer per month, the calculation is:

$$\text{Number of Staff} - \ \text{Months} = \frac{6000}{400} = 15\ \text{Staff} - \text{Months}$$

This means that the project will require a total of 15 Staff-Months to complete.

- **Employee-Months Calculation**: To find the Number of Employee-Months, which indicates the number of employees required based on the project's duration, we divide the Staff-Months by the planned project duration in months. Assuming a project duration of 6 months:

$$\text{Number of Employee} - \ \text{Months} = \frac{\text{Staff} - \text{Months}}{\mathit{Project\ Duration}} = \frac{15}{6}$$
$$= 2.5\ \text{Employee} - \text{Months}$$

Therefore, with a 6-month duration, 2.5 Employee-Months are required, meaning approximately **3 full-time Employees** (rounding up from 2.5).

6.1.4 SALARY FOR DEVELOPMENT STAFF

To determine the total salary cost for the Morshed project, we first estimated the necessary staff based on the complexity and effort involved in the development phase. For a project of this scope and functionality, approximately three full-time developers will be required. Each developer's contribution will ensure that the system meets both functional and non-functional requirements within the specified timeline, supporting user registration, tour advertising, booking functionalities, messaging systems, and search and filtering options.

The average monthly salary for each developer is set at 13,000 SAR, based on typical market rates for software development roles in Saudi Arabia [3]. This rate reflects the expertise needed for web development projects, especially those involving languages like Dart and real-time communication capabilities. With a project duration of six months, the total salary cost is calculated as follows:

$$\text{Total Salary Cost} = \text{Number of Employees} \times \text{Average Monthly Salary} \times \text{Project Duration (in months)}$$

Applying this formula:

$$\text{Total Salary Cost} = 3 \text{ employees} \times 13{,}000 \text{ SAR} \times 6 \text{ months} = 234{,}000 \text{ SAR}$$

The estimated total personnel cost for the development phase of the Morshed project is 234,000 SAR. This cost estimate supports Morshed’s goal of maintaining a skilled team throughout the development period, ensuring the necessary expertise is in place to meet project objectives efficiently and within budget.

6.1.5 EFFORT ESTIMATION USING COCOMO MODEL

Our website is structured around a wide range of critical features, including user registration, tour advertising, booking functionalities, messaging systems, and search and filtering options. These features require careful coordination across user experience design, backend development, and frontend integration. Given the complexity and the need for accuracy in planning and execution, the Detailed COCOMO model is the most appropriate estimation cost approach. Detailed COCOMO allows us to break down the project's scope into specific modules, phases, and activities, aligning with Morshed's defined functional and non-functional requirements.

The choice of the Detailed COCOMO model will enhance our ability to:

- Allocate resources efficiently for each phase of development.
- Identify and manage potential risks.
- Accurately estimate time and budget based on specific feature sets and complexity levels.

The primary reasons for using the Detailed COCOMO Model:

1. **Complex Feature Set:** The Morshed platform includes various intricate features like user registration, tour advertising, and integrated chat systems. Detailed COCOMO provides accurate effort and cost estimations tailored to these specific functionalities.

2. **Visibility into Development Phases:** The model offers clear insights into each development phase, allowing for precise planning and resource allocation as the project progresses.

3. **Customization and Flexibility:** Detailed COCOMO can be adapted to reflect Morshed's unique requirements, ensuring that estimations align closely with project conditions.

4. **Risk Management:** It enables early identification of potential risks, facilitating the development of strategies to mitigate issues related to usability, availability, and recovery times.

5. **Effective Scope Control:** The model helps manage changes effectively, assessing their impact on timelines and costs to minimize scope creep.

6. **Requirement Traceability:** Integrating with a Requirements Traceability Matrix (RTM) ensures all requirements are tracked and met throughout the project lifecycle.

7. **Scalability:** Detailed COCOMO supports scalability, making it easier to incorporate future enhancements as user needs evolve.

- ### Select the constant values

For the Detailed COCOMO model we will use Semi-detached which is a development method that consists of a mixture of experienced and inexperienced team members. We will use these formulas to estimate the effort in staff-month and time for months: [9].

Effort (person-months) = a * (KLOC) ^ b

The constants for the Effort equation are a & b for effort estimation, while a = 3.0 which is chosen for medium to high complexity, reflecting increased development effort, and b = 1.12 which is indicates that effort grows faster with project size, accounting for higher complexity.

Time (months) = c * (Effort) ^ d

The constants for the Time equation are c & d for time estimation, while c = 2.5 which is reflects the extended timeline needed due to complexity, and d = 0.35 which is shows that time increases more slowly than effort.

- ### Cost drivers

In Table [6] Cost drivers with their ratings and effort multipliers are displayed.

TABLE 6:COST DRIVERS

Cost driver	Rating	Effort Multiplier
Product Attributes		
RELY	High	1.15
DATA	Nominal	1.00
CPLX	High	1.15
Computer Attributes		
TIME	Nominal	1.00

STOR	Nominal	1.00
VIRIT	Nominal	1.00
TURN	Nominal	1.00
Personnel Attributes		
ACAP	High	0.86
AEXP	Nominal	1.00
PCAP	High	0.86
VEXP	Nominal	1.00
LEXP	High	0.95
Project Attributes		
MODP	High	0.91
TOOL	Nominal	1.00
SCED	Nominal	1.00
Effort Adjustment factor (Product of Effort Multiplier) = 1.15 * 1.00 * 1.15 * 1.00 * 1.00 * 1.00 * 1.00 * 0.86 * 1.00 * 0.86 * 1.00 * 0.95 * 0.91 * 1.00 * 1.00 = 0.846		

- **Effort using COCOMO model**

Effort = a * (Size) ^ b

Effort = a * (KLOC) ^ b

Effort = 3.0 * (6) ^ 1.12 = 22.32 Person - Months

Adjusted effort = Effort * EAF

Adjusted effort = 22.32 * 0.846 = 19 Staff - Month

Cost = Salary * Adjusted effort

Cost = 13000 * 19 = 247.000 SAR

6.2 TOOLS COST

TABLE 7: TOOLS COST

Type	Tool	Cost
Software	Programming language (Dart)	Free
	IDEs (NetBeans and Visual Studio)	
	Database management system (MySQL)	
	Zoom	
	WhatsApp	
	Microsoft word	
	Jira	
	GitHub	
	Figma	
	Canva	
	Google Forms	
	Project plan 365	37.5 SAR x 6 months = 224.775 SAR
Hardware	3 Asus VivoBook Laptops	3 x 2549 SAR = 7,647 SAR
	Lenovo ThinkSystem ST550 server	16,871 SAR
Total cost		224.775 + 7,647 + 16,871 = 24,742.775

1.2. Total cost

To calculate the total cost, we will start by including **the tools cost, estimated at 24,742.78** SAR. Additionally, we have assessed the development cost of "Morshed" using two estimation methods. **The Function Point (FP) method provided an estimate of 234,000 SAR,** while the **COCOMO model yielded a slightly higher estimate of 247,000 SAR.**

To calculate the percentage difference:

$$\text{Difference} = \frac{|247{,}000 - 234{,}000|}{247{,}000} \times 100 = \mathbf{5.3\%}$$

Since the difference between these estimates is only 5.3%—well within our 20% tolerance threshold, we will adopt the higher estimate from the COCOMO model for a more conservative budgeting approach. This approach will help secure adequate resources for the successful development of "Morshed," minimizing potential financial risks.

Tools cost + Development cost = Total cost

24,742.775 SAR + 247,000 SAR = **271,742.775 SAR**

7. Quality Attributes

TABLE 8: QUALITY ATTRIBUTES

Quality attribute	Description	Measurement	Assessment
Usability	The system shall be easy to browse and navigate, users should be able quickly find relevant information to complete tasks within 10 minutes.	**Time-on-task** Track the average time users take to find relevant information and complete specific tasks [4].	Time on Task= $\frac{Sum\ of\ indiviual\ Times}{Total\ users}$
Availability	The system shall be available 95% of the time,	**System Availability** percentage of the actual operation time divided by the	System Availability= $\frac{Up\ Time}{Total\ Time}\ x\ 100$

	ensuring minimal downtime.	total amount of observational time [5].	
Recoverability	In the event of a system failure, the system shall be restored to full functionality within a maximum of 2 hours to ensure minimal disruption to users.	**Maximum Tolerable Downtime** is defined as the maximum duration a system can remain inoperable before causing severe impact on an organization [6].	Maximum Tolerable Downtime (MTD) = $Recovery\ Point\ Objective(RTO) + Recovery\ Time\ Objective\ (RPO)$
Response Time	The system shall provide a response time of less than 5 seconds	**Average response time (ART)** is a measurement of the amount of time a server or application takes to respond to all its data inputs and requests [7].	Average response time (ART)**=** $\frac{\text{Sum of response time for each user request}}{Total\ Number\ of\ requests}$
Maintainability	System updates, including patches and new features, shall maintain a Maintainability Index (MI) of 70 or higher for all critical components, on a scale of 0–100, to ensure good maintainability and ease of future modifications.	**Maintainability Index** is a software metric which measures how maintainable (easy to support and change) the source code is. The maintainability index is calculated as a factored formula consisting of Lines of Code, Cyclomatic Complexity and Halstead volume [8].	Maintainability Index $\mathrm{MI} = 171 - 5.2 \times ln(\mathrm{V}) - 0.23 \times \mathrm{G} - 16.2 \times ln(\mathrm{LOC})$

8. EXTERNAL AND INTERNAL

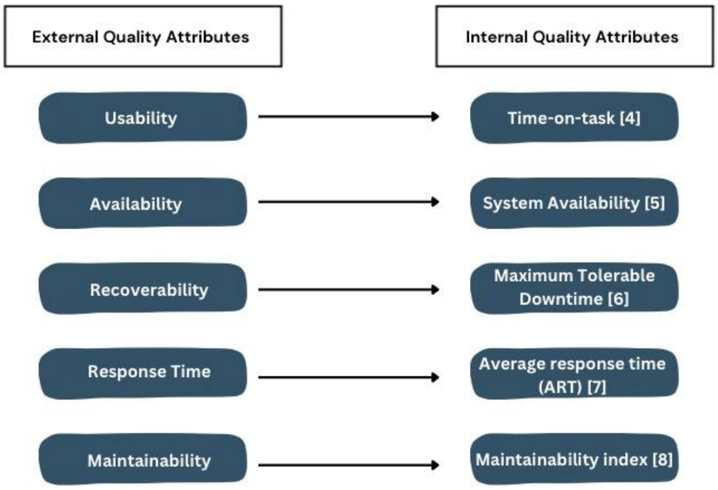

External Quality Attributes
Internal Quality Attributes
Usability
Time-on-task [4]
Availability
System Availability [5]
Recoverability
Maximum Tolerable Downtime [6]
Response Time
Average response time (ART) [7]
Maintainability
Maintainability index [8]

FIGURE 7: EXTERNAL AND INTERNAL ATTRIBUTES

9. REFERENCES

[1] admin, “Function Point Languages Table,” *QSM SLIM-Estimate*, Apr. 20, 2017. https://www.qsm.com/resources/function-point-languages-table

[2] C. Jones, "Thirty Years of IFPUG: Software Economics and Function Point Metrics," *IFPUG*, Version 10.0, Apr. 2017. [Online]. Available: https://ifpug.org/wp-content/uploads/2017/04/IYSM.-Thirty-years-of-IFPUG.-Software-Economics-and-Function-Point-Metrics-Capers-Jones.pdf

[3] Salary Explorer. "Developer / Programmer Average Salary in Saudi Arabia 2024." *Salary Explorer*. [Online]. Available: https://www.salaryexplorer.com/ar/average-salary-wage-comparison-saudi-arabia-developer-programmer-c191j783

[4] “Learn the basics about time on task | Product stash.” https://www.productstash.xyz/product-metrics/time-on-task#:~:text=the%20task%20quickly.-,Time%2Don%2Dtask%20is%20calculated%20by%20recording%20how%20long%20it,by%20the%20number%20of%20users.

[5] “What is system availability? | Learning Center | MaintainX,” May 29, 2024. https://www.getmaintainx.com/learning-center/system-availability

[6] “Maximum Tolerable Downtime - an Overview | ScienceDirect Topics.”

https://www.sciencedirect.com/topics/computer-science/maximum-tolerable-downtime#:~:text=There%20is%20a%20correlation%20between,%2C%20MTD%20%3D%20RTO%20%2B%20WRT.

[7] “5 Types of Response Time Metrics and How to Measure It.” *Indeed Career Guide*,

https://www.indeed.com/career-advice/career-development/response-time-testing

[8] “Maintainability Index (MI).” https://www.projectcodemeter.com/cost_estimation/help/GL_maintainability.htm#:~:text=Maintainability%20Index%20is%20a%20software,Cyclomatic%20Complexity%20and%20Halstead%20volume.

[9] "COCOMO Model," 2021. [Online]. Available: https://www.javatpoint.com/cocomo-model. [Accessed 11 November 2023.