



MINISHELL

Asmaa Aljazza

Basma Eisa

Jun 2025 – July 2025

Acknowledgements

Abstract

Table of content

CHAPTER 1

INTRODUCTION

1.1 Overview

1.2 Problem Statement

1.3 Project objectives

1.4 SDLC Model

1.5 Gant chart

CHAPTER 2

LITERATURE REVIEW

2.1. Introduction to Shell

2.2. Bash

CHAPTER 3

METHODOLOGY

3.1. Overview

3.2. Project Backlog

3.3. Sprint Cycle

Sprint 1:
Sprint Planning

Table 3. 1:Sprint 1_Sprint Planning

ID	User story	Description	Days
S101	As a user, I want to		

Development

Table3. 1:Sprint 1_Sprint Development

ID	Tasks
S101	
S102	
S103	
S104	
S105	

Daily Scrum

Table 3. 2:Sprint 1_Daily Scrum

[illegible]

3.4. Requirements

Listed below are the functional requirements and non-functional requirements of the system. In the priority column, the following shorthand is used:

M – mandatory requirements (something the system must do).

D – desirable requirements (something the system preferably should do).

O – optional requirements (something the system may do).

Functional Requirements

Functional requirements outline the specific functionalities, features, and interactions that a system, software application, or product must exhibit to meet the needs of its users and stakeholders. These requirements detail what the system is supposed to do and describe its expected behaviour under various conditions.

Table 3. 3:Functional Requirements

FID	Functional	Description	Priority
F101			
F102			
F103			
F104			
F105			
F106			
F107			

F108			
-------------	--	--	--

Non-Functional Requirements

Non-functional requirements specify the criteria that are used to judge the operation of a system rather than specific behaviours. Unlike functional requirements, which describe what the system should do, non-functional requirements describe how well the system should perform those functions.

Table 3. 4:Non-Functional Requirements

No.	Function	Description	Priority
1			M
2			
3			
4			
5			D
6			

Tools

Table 3. 5:Tools

No.	Tools Utilized
1	Visual Studio
2	
3	
4	
5	
6	
7	

Programming Languages

Table 3. 6:Programming Languages

No.	Programming Languages
1	
2	

CHAPTER 4

DESIGN MODELS

4.1. **Overview**

4.2. Context Diagram-0

4.3. Data flow Diagram

4.4. Activity Diagrams

CHAPTER 5

EXPERIMENTS AND RESULTS

5.1. Overview

5.2. Testing methodologies

System Design Testing

Table 5. 1:Test Case 1

ID:	Id = Name of functionality	Description:	
T.C:	01.1.01	Target:	Main Flow / The name of expected invalid case
Input:		Output:	
Rules:			
The Steps Should follow to get this output from that input			

5.2.2 Acceptance System Results

User Acceptance Testing (UAT) typically occurs in the final phases of the software development life cycle. During this stage, a questionnaire was administered to a group of individuals who were tasked with using the software and providing feedback by answering various questions about their experience with the site. Including:

Questions to us and to Other student give form to students after give them the code to test it.

Then set the result here.

CHAPTER 6

CONCLUSION

6.1. Overview

6.2. Summary of the project

6.3. Achieved objectives

6.4. Main contributions of the work

6.5. Limitation