



GREEN UNIVERSITY OF BANGLADESH (GUB)

# Operating System Algorithms Tester

## Submitted by

Umme Loara (191002187)

Sadia Binta Sadek (191002088)

Sumiya Akter (191002215)

*A project submitted to the Department of Computer Science & Engineering  
for the partial fulfillment of the degree of  
Bachelor of Science in Computer Science & Engineering*

## Supervised by

Md. Riad Hassan

Lecturer, Department of CSE



Department of Computer Science & Engineering

Green University of Bangladesh  
220/D Begum Rokeya Sarani, Dhaka-1207  
Month, Year

# Declaration

I hereby declare that this thesis is based on the results found by myself. Materials of work found by other researcher are mentioned by reference. This project, neither in whole nor in part, has been previously submitted for any degree.

---

Name: Umme Loara

ID: 191002187

---

Name: Sadia Binta Sadek

ID: 191002088

# Certificate

This is to certify that the thesis entitled **Operating System Algorithms Tester** has been prepared and submitted by **Umme Loara** in partial fulfillment of the requirement for the degree of Bachelor of Science in Computer Science and Engineering on January 6, 2023.

---

Md. Riad Hassan  
Supervisor

Accepted and approved in partial fulfillment of the requirement for the degree Bachelor of Science in Computer Science and Engineering.

---

Prof. Dr. Md. Abdur Razzaque  
Chairman

---

Mr. Tamim Al Mahmud  
Member

---

Mr. Misbah Ul Haque  
Member

---

Prof. Dr. Md. Monirul Islam  
Member (External)

# Acknowledgments

Provide a brief statement acknowledging the efforts of any participants or consultants who are not included as authors of the manuscript. State all of the funding sources for the work, ensuring that the statement adheres to the guidelines provided by the funding institution.

One can also mention and acknowledge supervisor, thesis or project mates, departmental lab facilities, scholarships or workshops, etc. However, acknowledgment must be in a single page.

# **Abstract**

This project is basically a converter. And the main objective is to help the student who are sometimes confused that their algorithms is correct or not and the second objective is to help teacher to get all the algorithm in one place with example that help them to easily explain the algorithms to students.

This project is totally different from any other algorithms converter. It is basically made in Netbeans 8.2

And we test the project manually and the performance is 80

## TABLE OF CONTENTS

<b>Declaration</b> . . . . .	i
<b>Certificate</b> . . . . .	ii
<b>Acknowledgments</b> . . . . .	iii
<b>Abstract</b> . . . . .	iv
<b>List of Figures</b> . . . . .	vii
<b>List of Tables</b> . . . . .	viii
<b>1 Introduction</b> . . . . .	1
1.1 Motivation . . . . .	1
1.2 Aims and Objectives . . . . .	2
1.3 Research Questions . . . . .	2
1.4 Research Contribution . . . . .	2
1.4.1 Contributions . . . . .	2
<b>2 Literature Review</b> . . . . .	3
2.1 Introduction . . . . .	3
2.2 Proposed Idea . . . . .	3
2.3 Conclusion . . . . .	4
<b>3 The Design Methods and Procedures</b> . . . . .	6
3.1 Introduction . . . . .	6
3.2 System Design . . . . .	6
3.2.1 Data Flow Diagram . . . . .	7
3.2.2 UML Use Case Diagram: . . . . .	9

3.2.3	UML Sequence and Communication (Collaboration) Diagram: . . . . .	10
3.2.4	UML Class Diagram: . . . . .	12
3.2.5	System Architecture (Low Level and High Level Design): . . . . .	13
3.3	Financial Planning for the Project . . . . .	15
3.4	Gantt Chart (Around 4 months) . . . . .	15
3.5	Conclusion . . . . .	15
<b>4</b>	<b>Evaluation of the Developed System . . . . .</b>	<b>16</b>
4.1	Introduction . . . . .	16
4.2	Simulation In Bash Replit: . . . . .	16
4.3	Results and Discussions . . . . .	18
4.4	Analysis and Outcome . . . . .	18
<b>5</b>	<b>Conclusion . . . . .</b>	<b>22</b>
5.1	Limitation of the research . . . . .	22
5.2	Practical Implications . . . . .	22
5.3	Future Works . . . . .	23
	<b>References . . . . .</b>	<b>24</b>



# List of Figures

2.1	First Interface [1]	3
2.2	Proposed Method	5
3.1	System Prototype	7
3.2	System Internal Architecture	7
3.3	DFD Level 0	8
3.4	DFD Level 1	9
3.5	Use Case Diagram for OS Algorithms Tester	10
3.6	Sequence and Communication Diagram	12
3.7	UML Class Diagram	13
3.8	High Level Design	14
3.9	Low Level Design	14
3.10	Budget Plan	15
3.11	Gantt Chart	15
4.1	Splash Screen	17
4.2	Test- 1: For number Block is 0	18
4.3	Test-02: For number of File 0	19
4.4	Test-03: For number of block 2 and number of file 1	19
4.5	Test-04: For number of file and Block is negative.	20
4.6	Test-05: For file and block size negative.	20
4.7	Manual Testing.	21

# List of Tables

# Chapter 1

## Introduction

Algorithm is a process or set of rules to be followed in calculations or other problem-solving operations. So, Algorithm refers to a set of rules or instructions that step-by-step define how a work is to be executed upon in order to get the expected results. It is easy to understand the problem and solve the problem. It can give the easy procedure to solve the problem. “Operating System Algorithm Tester” is the software where a group of people find their algorithms in one place and understand the topic through internet.

This project is mainly upholds the convert system of Operating System algorithm. In case a student can be upset when he do not understand his topic (algorithm), this project or product will make for this reason, any time he can use this site, all algorithm problem statement and its solution are described on his screen. Easily he can understand the solution and will get a good result. The operating system algorithm tester will provide instructions and inputs from the user to the output of the specific algorithm and its proper analysis. So that users can easily understand its specific algorithm. This software allows the user to easily access the website without any dollars. This software also allows user to perform various algorithms.

### 1.1 Motivation

1. While implementing operating system algorithms, we felt the necessity of implementation of these in an effective and efficient way so that time consumption can be

reduced and also it can be more users friendly.

2. So, this is going to be more effective as well as user friendly as simple interfaces are going to be used for representation.

3. To find or solve all the algorithm in one place, we choose this project.

## **1.2 Aims and Objectives**

The main objective of the project are:

- To enrich our knowledge in Operating System.
- To develop the idea of algorithms converter.
- To gather knowledge of memory management techniques, CPU Scheduling Algorithm, Contiguous Memory Allocation Technique, Deadlock Avoidance, File allocation strategies.

## **1.3 Research Questions**

1. A teacher can easily find all the algorithms at one place for which it will be helpful for teachers to understand the algorithm with examples. 2. Students can clear their confusion easily by testing their algorithm in our converter that their algorithm is correct or not.

## **1.4 Research Contribution**

### **1.4.1 Contributions**

Many have made many different converters but for specific algorithms. We will put all the algorithms in our project so that a student and a teacher can easily find all the algorithms in one place for which it will be helpful for teachers to understand the algorithms with examples and students can also clear their confusion easily.

# Chapter 2

## Literature Review

### 2.1 Introduction

The related works of this project is create converter based on data structure course algorithms, microprocessor and micro-controllers course algorithm etc. Also we can make any type of converter if the topic concept is clear to us.

### 2.2 Proposed Idea

The figure shows the model of our project. It has 5 parts. Each and every part has some part. The proposed system takes input from user that which operations or which algorithm they want. So our first interface will looks like:

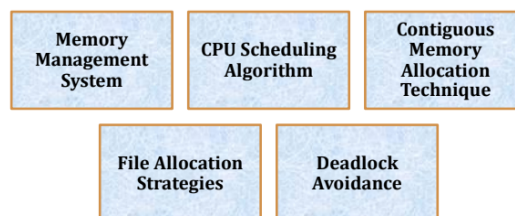


Figure 2.1: First Interface [1]

Users select their options. And then another interface open with proper explanations of this topic. And then after again selecting then another interface is open and user input

their value and find the result with explanations.

Memory Management System:

It has 2 parts:

1. Multiprogramming with a Fixed number of Tasks;
2. Multiprogramming with a Variable number of Tasks;

CPU Scheduling Algorithm:

It has 4 parts:

1. First Come First Serve (FCFS)
2. Shortest Job First (SJF)
3. Priority Scheduling Algorithm
4. Round Robin Algorithm

Contiguous Memory Allocation Technique

It has three parts:

1. Worst-fit memory allocation technique
2. Best-fit memory allocation technique
3. First-fit memory allocation technique

File Allocation Strategies

It has three part:

1. Sequential file allocation strategies.
2. Linked file allocation strategies.
3. Indexed file allocation strategies.

Deadlock Avoidance

1. Banker's Algorithm

## **2.3 Conclusion**

The proposed technique, tools, and methodology are very known and easily collectable for the target user (students of CSE department, Teachers of CSE department) to learn or to taught easily all the algorithms. To find the all algorithms we need a device such

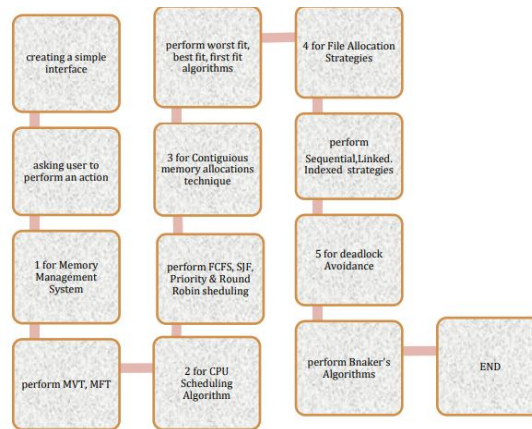


Figure 2.2: Proposed Method

as smart phones, laptops, computers and most importantly if we published the converter in the internet with a website then we need to internet connection.

# **Chapter 3**

## **The Design Methods and Procedures**

### **3.1 Introduction**

This section should be a straightforward description of the methods used in your study. Each method should be described in a separate section. Begin, in a single section, with a statement of the materials used in the study, indicating the vendor and vendor contact information for each material. This information is critical so that readers have the capability to repeat the work in their own institutions. Next describe, in separate sections, each key procedure and technique used in the study. Keep explanations brief and concise. If a specific experimental design is utilized, describe this design in the second section of the Methods, after the materials section. Similarly, if a theoretical or modeling component is utilized, it should also be incorporated in the initial portion of the Methods. Finally, remember to describe the statistical analysis methods that were utilized to analyze the results, most likely in the final section of the Methods section. Although it is typically not recommended, the use of the passive voice is probably appropriate in the Methods section.

### **3.2 System Design**

These are only examples of how one can add as many as sections and subsections needed according to the topic.



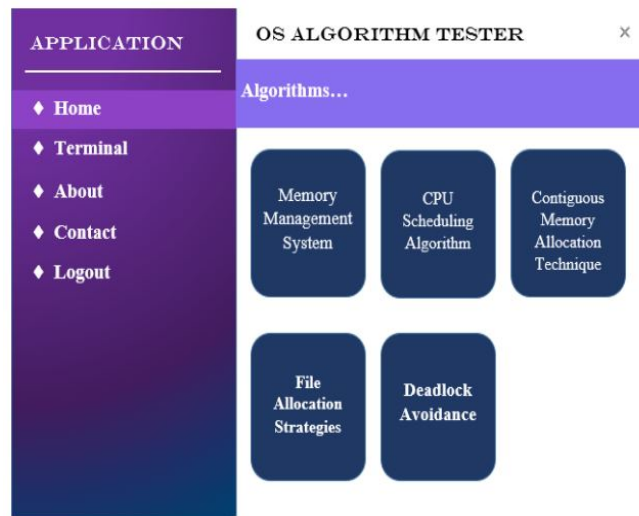


Figure 3.1: System Prototype

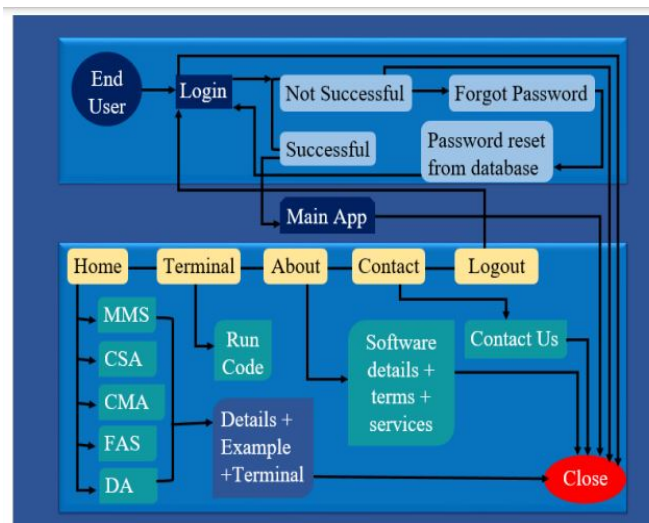


Figure 3.2: System Internal Architecture

### 3.2.1 Data Flow Diagram

Data flow diagram is graphical representation of flow of data in an information system. It is capable of depicting incoming data flow, outgoing data flow and stored data. A neat and clear DFD can depict the right amount of the system requirement graphically.

It shows how data enter and leaves the system, what changes the information, and where data is stored.

### 3.2.1.1 » For DFD Level-0:

Case 1: User will send request for login permission to process and the process will accept or reject depending on the request.

Case 2: Admin panel can store data provided by the user while signing up and cross matches when user is trying to log in.

### 3.2.1.2 » For DFD Level-1:

Case 1(Log in): User will send request for login permission and process will cross match the information provided along with the request with the database.

Case 2(Sign Up): A new user will have to fill up a sign up form in order to in access to the site and process will store the data in the database in a particular way. After that he can log in similarly to the case 1.

Case 3: User will send request asking to access the algorithms and other factors and process will check permission and grant or reject access.

These are some of the possible scenarios that is happening in the figures.

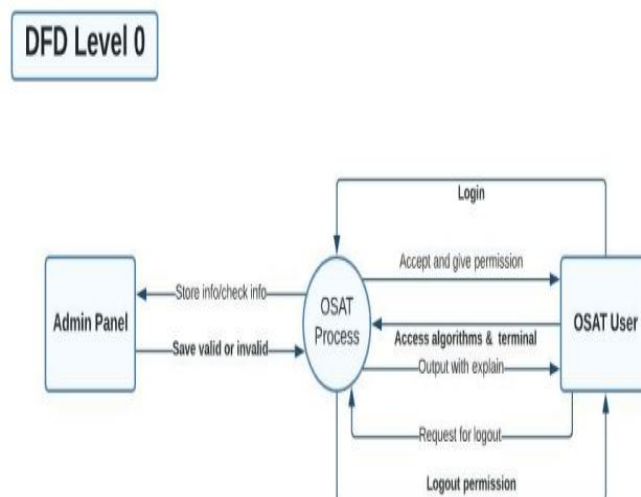


Figure 3.3: DFD Level 0

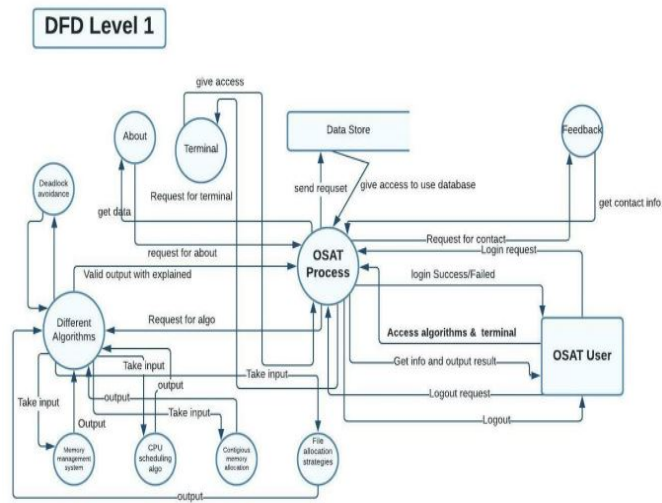


Figure 3.4: DFD Level 1

### 3.2.2 UML Use Case Diagram:

Use case system functionality presents system requirements from a user's perspective. A use case diagram is a dynamic or behavioral image in UML. Actors using case diagrams and models the functionality of a system using cases. A use case diagram is the situations that describe the interactions between a user and a system. And also it can summarize the details of a system's user and their interactions with the system.

1. Association: Association link is shown by connecting an actor to a use case by a solid link. 2. Extend Relationship Between Two Use Cases As the name implies it extends the base use case and adds more functionality to the system. Here are a few things to consider when using the  $\llcorner$  extend  $\lrcorner$  relationship.

(a) The extending use case is dependent on the extended (base) use case. All the terms (algorithms, terminal, feedback) use case doesn't make much sense without the login section use case.

(b) The extending use case is usually optional and can be triggered conditionally. When the user needs help, the use case "Get help on sign in/sign up" will be triggered. In other case, when the user will want to leave a feedback, he will need to login in case he isn't already logged in. Hence, "User login" use case will be triggered.



an application.

1. The actor of the project is user.

2. There used 8 lifelines in the project. And their objects are: User, Store-info, Site, Admin, Operation, Operation 1 and their classes are: Sign up / Sign in, Database, About, Feedback, Algorithms and Terminal.

The project diagram requires 4 reply messages, 2 asynchronous messages and the remaining 9 synchronous messages. The reply messages will be:

- (i) Sending Confirmation: This reply message will basically verify whether the user registration has been confirmed or not to access our project website.

- (ii) Access Accept / Reject: This reply message will be for whether the login information is correct or not.

- (iii) Sending Output According to the input values: This reply message will be the algorithm that the user chooses for the output he wants with explanation.

- (iv) Response According to the Feedback: And this will be the last reply message, the user uses the website to give feedback according to the feedback.

The two asynchronous messages that project are:

1. To know about site / app

2. To know about Admin Contact Info. And the remaining 9 are synchronous message.

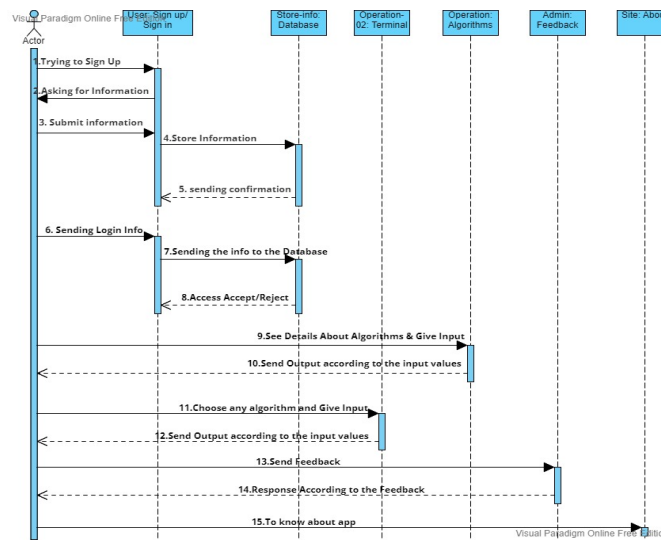


Figure 3.6: Sequence and Communication Diagram

### 3.2.4 UML Class Diagram:

Class diagrams are the main building blocks of every object-oriented method. The class diagram can be used to show the classes, relationships, interface, association and collaboration. UML is standardized in class diagrams. Science classes are the building block of an application that is based on OOPs, so as the class diagram has an appropriate structure to represent the classes, inheritance, relationships and everything that OOPs have in their context. It describes various kinds of objects and the static relationship between them. In this project, total number of algorithms is 13. And the project based on about different types algorithms. There are 13 different OS algorithms. First of all user need to register first and after then they logged in the website. So it has different class for register and user. After than a user direct entry in the algorithms page, so a class is used that name is Algorithms. Then Algorithms class divide into 5 parts, so there are different 5 classes. And each and every class divided into many part. So there are total 13 classes for algorithms.

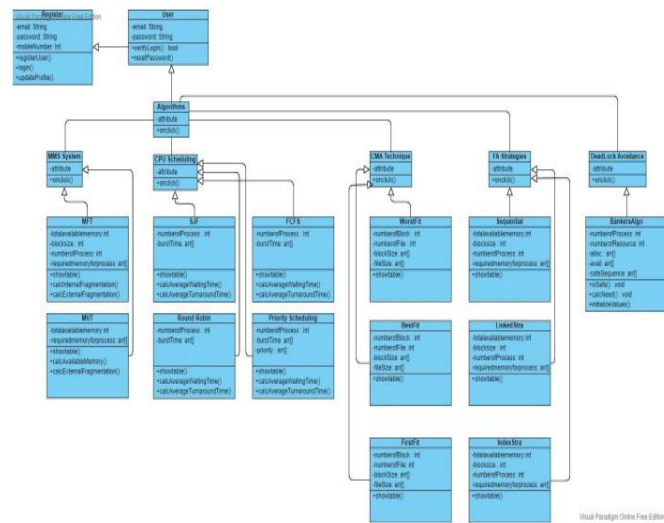


Figure 3.7: UML Class Diagram

### 3.2.5 System Architecture (Low Level and High Level Design):

Architectural design is concerned with understanding how a system should be organized and the overall structure of that system should be designed. In the software development process model, the architectural design is the first step in the software design process. It is an important link between design and requirements engineering, as it identifies the main structural elements of a system and the relationship between them.

The output of the architectural design process is an architectural model that describes how the system is organized as a set of communication elements.

In practice, there is a significant overlap between the required engineering and architectural design processes. Ideally, a system specification should not include any design information. Without a very small system it would be unrealistic. Architectural decomposition is usually necessary to form and organize the specification.

Architectural design is a creative process where one can design a system organization that will meet the functional and non-functional requirements of a system. Since it is a creative process, the activities in the process depend on the type of system development, the background and experience of the system architect and the specific requirements for the system. So it is important to think of architectural design as a series of

decisions rather than a sequence of activities.

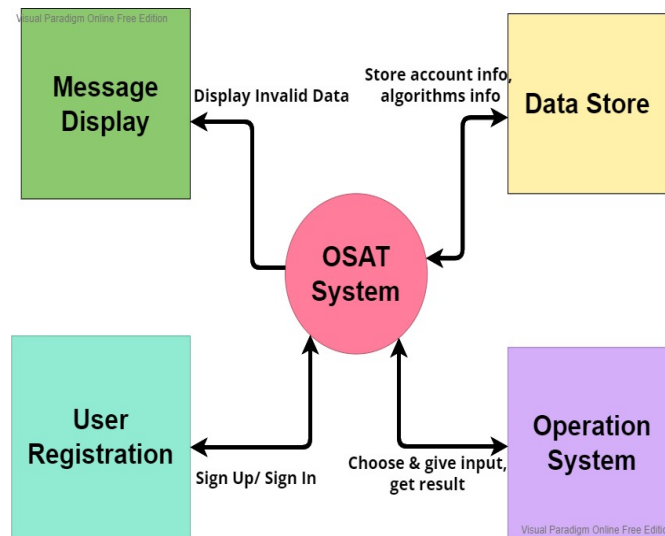


Figure 3.8: High Level Design

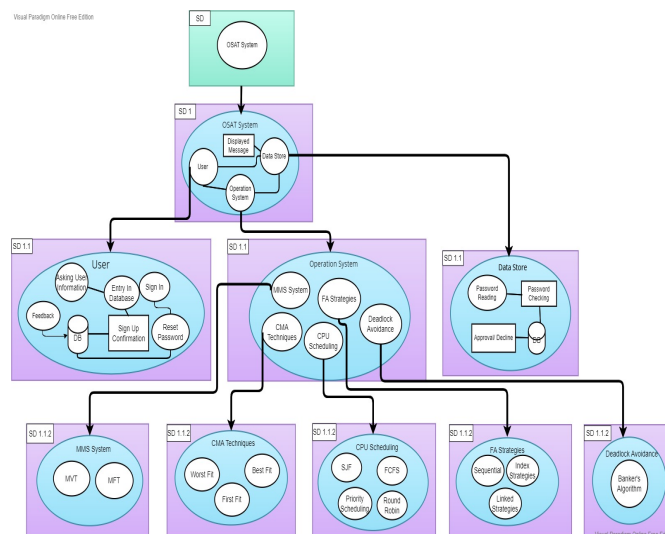


Figure 3.9: Low Level Design



### 3.3 Financial Planning for the Project

Sl No.	Subject	Unit	No. Of Unit	Per Unit Cost	Duration	Total Cost
1	Researcher	Person	3	25000		75000
2	Data Collection	Survey Travel	1	30000		30000
3	Materials	Laptop	1	65000		65000
4	Research Grant Policy	Publication	2	7000		7000
5	Miscellaneous			600	12	7200
					Total	184200

Figure 3.10: Budget Plan

### 3.4 Gantt Chart (Around 4 months)

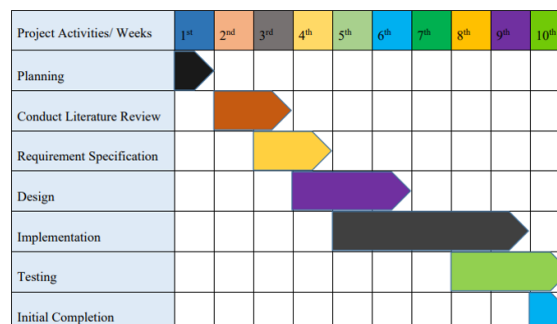


Figure 3.11: Gantt Chart

### 3.5 Conclusion

To complete a project there need so many design, prototype. By using prototype and design it is very easy to understand the project for anyone and also easy to implement. We do here different type of diagram, design that are specify which function will do what.

# Chapter 4

## Evaluation of the Developed System

### 4.1 Introduction

### 4.2 Simulation In Bash Replit:

Software testing is a method of testing whether the actual software product meets the expected requirements and making sure that the software product is flawless. It involves the execution of software / system components using manual or automated tools to evaluate one or more features of interest. The purpose of software testing is to identify errors, gaps, or missing requirements as opposed to actual requirements.

There are various types of testing available in the market, which are used to test the application or the software.

**Manual Testing:** : Manual testing is the process of testing the effectiveness of an application according to the needs of the customer without the help of automation tools. When performing manual testing on any application, we do not need any specific knowledge of any testing tool, but rather have a good idea of the product so that we can easily prepare test documents. Manual testing can be further divided into three types of testing, which are as follows:

1. **White Box Testing:** White box testing is done by developers, where they check every line of code before giving it to the test engineer. Since the code is visible to

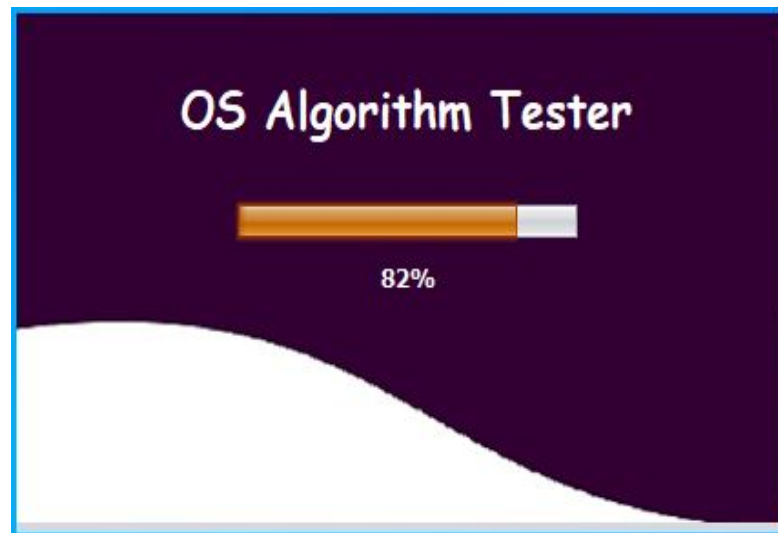


Figure 4.1: Splash Screen

developers during testing, it is also known as white box testing.

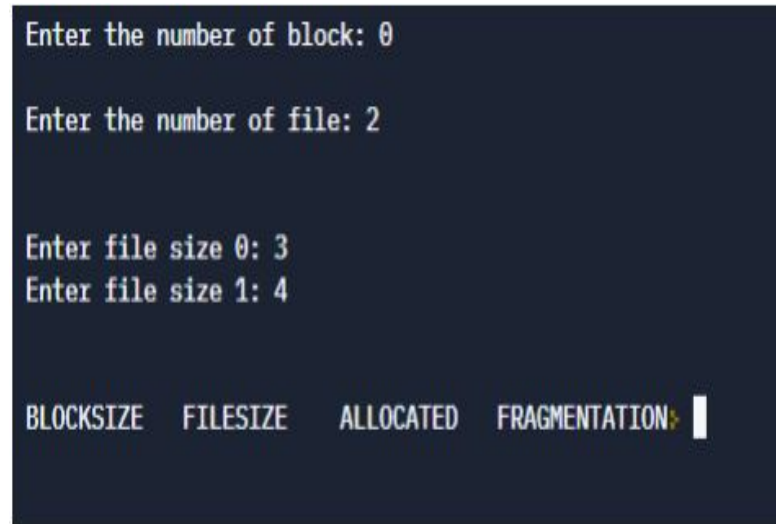
2. **Black Box Testing:** Black box testing is done by test engineers, where they can test the functionality of an application or software according to the needs of the customer / client. In this case, the code is not visible during the test; Hence it is known as black-box testing.

3. **Gray Box Testing:** Gray box testing is a combination of white box and black box testing. This could be done by someone who knows both coding and testing. And if a single person performs the white box, as well as the black-box test for the application, it is known as the gray box test.

**Automation Testing:** Automation testing is the process of converting any manual test case into a test script with the help of automation tools, or any programming language known as automation testing. With the help of automation tests, we can speed up our testing because here, we do not need any human effort. We need to write a test script and run those scripts.

### 4.3 Results and Discussions

We do here manual testing. And add the screen shot. Due to possibility of crush our project we do the same thing in “Bash replit” by using the code in Operating System.



```
Enter the number of block: 0

Enter the number of file: 2

Enter file size 0: 3
Enter file size 1: 4

BLOCKSIZE  FILESIZE  ALLOCATED  FRAGMENTATION
```

Figure 4.2: Test- 1: For number Block is 0

### 4.4 Analysis and Outcome

Testing Performance (%) = (Successful Test Cases / Total Number of Test Cases) × 100%

Testing Performance for Manual Testing =  $(4/5) \times 100 = 80$

```
Enter the number of block: 2

Enter the number of file: 0

Enter block size 0: 2
Enter block size 1: 4


BLOCKSIZE  FILESIZE  ALLOCATED  FRAGMENTATION
```

Figure 4.3: Test-02: For number of File 0

```
Enter the number of block: 2

Enter the number of file: 1

Enter block size 0: 5
Enter block size 1: 4

Enter file size 0: 7


BLOCKSIZE  FILESIZE  ALLOCATED  FRAGMENTATION
5          7        NO          ---
```

Figure 4.4: Test-03: For number of block 2 and number of file 1

```
Enter the number of block: -2

Enter the number of file: -2


BLOCKSIZE  FILESIZE  ALLOCATED  FRAGMENTATION
```

Figure 4.5: Test-04: For number of file and Block is negative.

```
❖ bash main.sh
Enter the number of block: 2

Enter the number of file: 3

Enter block size 0: -8
Enter block size 1: -6

Enter file size 0: -4
Enter file size 1: -3
Enter file size 2: -5


BLOCKSIZE  FILESIZE  ALLOCATED  FRAGMENTATION
-6         -4       NO        ---
-8         -3       NO        ---
0          -5       YES       5
```

Figure 4.6: Test-05: For file and block size negative.

Criteria	Action	Input (Test Case)	Expected Output	Actual Output	Test Result
Worst Fit	1. User enter number of Block equal to 0	0	Not Possible because of number of block is 0	It taken number of file and file size but not take block size. And last it show blank table.	Pass
	2. User enter number of file 0	0	Not possible because of number of file 0	It taken block size and not take file size and show a blank table.	Pass
	3. User enter number of block 2 & number of file 1	2 , 1	Possible, because if block size & file number give 1 , it will must be possible.	It taken block size, file size and give a output table.	Pass
	4. User enter number of block & number of file negative value.	-2, -2	Not possible, it cannot show the table & don't take the file size, block size.	It do not take any file size & block size	Pass
	5. User enter block size value negative.	Block size: - 8, -6 File size: -4, -3,-5	Possible but only two file allocated in block size.	It show table for the block size 0 but that is not given.	Not Pass

Figure 4.7: Manual Testing.

# Chapter 5

## Conclusion

From this project we will try to summarize the whole idea and algorithms that we practiced and learned on our whole operating system course and describe what the algorithm is. A person can demonstrate the basic concept, algorithms and their proper usages.

### 5.1 Limitation of the research

1. We cannot add SQL Database in our project though we will add this very soon.
2. For Neatbeans version issue, our main interface shows invisible. To solve this problem we need to again design the whole project which is now not possible.

### 5.2 Practical Implications

Our project can help a beginner to learn the basic of operating system course algorithms.

1. This project helps the students and teachers. Students can find all the algorithms in one place. And teachers also can explain this algorithm easily, find all the algorithms in one place.
2. It helps someone to enrich their knowledge with algorithms
3. It helps someone to find out the better algorithm with the comparison



## **5.3 Future Works**

In this project we will create a algorithm tester that work as converter which convert the all algorithm of operating system which is taught in our operating system course. So, if we create the converter based on operating system algorithm then we can also create converter based on microprocessor and micro-controllers course algorithm, algorithms course, data structures course algorithm etc.

This project specially focuses on the simple, user friendly and accurate results for operating system algorithms. By this project, the students, teachers and other users can easily test the algorithms, use it into their individual purpose or own field of work. Also, it can be used in a program for a company or industry for the management of their data or resources so that the process can be easier.

# References

- [1] Albert S. Woodhull Andrew S. Tanenbaum. Operating systems: Design and implementation. *Prentice Hall*, 70:263–286, 2007.