

Project: Routine Generator

Name: MD Omar Faruk Siam

Id: 0242220005341148

**Project Documentation**

Fall – 2023

**DEPARTMENT OF SOFTWARE ENGINEERING**

**TABLE OF CONTENTS:**

|  |  |
| --- | --- |
| 1. **Project Overview** | **3** |
|  |  |
| 1. **Obstacles** | **5** |
|  |  |
| 1. **Technology Requirement** | **5** |
|  |  |
| 1. **Milestone and reporting** | **6** |
|  |  |
| 1. **PROJECT METHADOLGY AND USER MANUAL** | **7** |
|  |  |
| 1. **PRIORITIZATION** | **12** |
|  |  |
| 1. **USE CASE** | **26** |
|  |  |
| 1. **Use Case Diagram** | **30** |

**Project Overview: Class Schedule Management System**

The "Class Schedule Management System" is a C-based console application designed to automate the process of creating and managing class schedules for educational institutions. The system aims to simplify the tasks related to teacher subject allocation, batch-wise subject distribution, and generation of daily class routines.

The primary goal of the project is to streamline the scheduling process within an educational institution, offering the following key features:

Efficient allocation of teachers to subjects based on their availability and maximum class limits.

Distribution of subjects to batches and sections in an organized manner.

Generation of comprehensive class routines for different batches and sections.

User-friendly interface for administrators to interact with the system and make necessary adjustments.

**Key Features:**

       *Header Files:*

1.            #include <stdio.h>

2.            #include <stdlib.h>

3.            #include <string.h>

       *Functions*:

1.            int fileCheck(FILE \*file);

2.             int rewrite(const char \*subject, char \*teacher);

3.             int fileRegenaret();

4.             int distribute(int loop);

5.             int afterdestributeTeachers();

6.             int BatchWiseSubRe();

7.             int destributeTeachers();

8.             int makeClasses();

9.             int makeRoutine();

10.                       void appendFile(char \*filename);

11.                       int find();

12.                       int FullRoutine();

       *Files*:

1.            char TeacherSubjects[30] = "TeacherSubjects.txt";

2.            char BatchWiseSub[30] = "BatchWiseSub.txt";

3.            char SubjectDestribution[30] = "SubjectDestribution.txt";

4.            char SubjectDestribution2[30] = "SubjectDestribution2.txt";

5.            char Temp[30] = "Temp.txt";

6.            char PreSchedule[30] = "PreSchedule.txt";

7.            char Shedule[30] = "Shedule.txt";

**Conclusion:**

The "Class Schedule Management System" is a comprehensive tool for educational institutions to automate and optimize the scheduling process. By providing efficient subject and teacher allocation, the system contributes to the smooth operation of classes and ensures a well-organized educational environment.

**Obstacles**

While crafting the project code, I encountered several challenges. The sheer scale of this project made it apparent that collaboration is essential, yet I chose to embrace this challenge independently. I aimed to streamline the system by employing clear and concise code. Among all the components, the creation mechanism for quizzes posed the greatest difficulty. To implement this section, I had to devise a method for generating random file names as addresses for each quiz file.

**Technology Requirement (Software and hardware)**

       Visual Studio Code

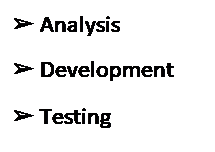
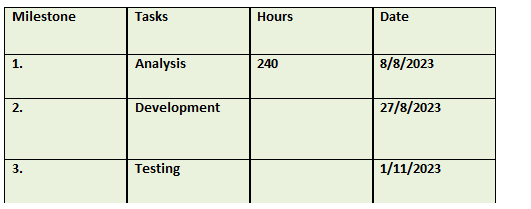
       Code Blocks

       Dev-C++

**Support**

Provide the full support for maintenance. Give proper support so that the project will run smoothly. And every time when ever you want to make routine, you will need to press “0” zero to refresh the files.

**Milestone and reporting**

**Milestone and reporting**

**Contact me**

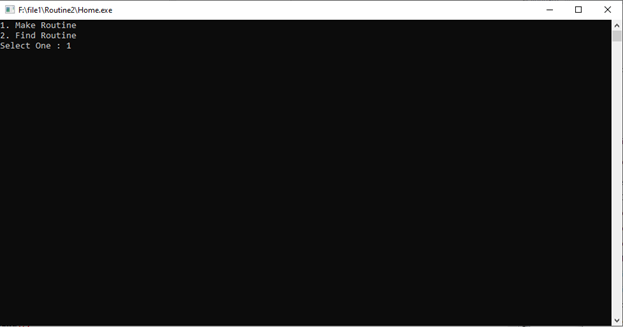
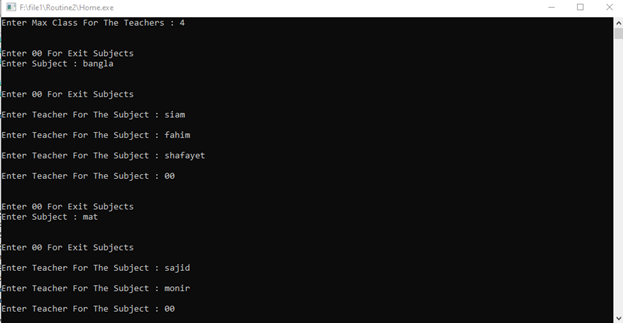
Email: omarfaruk122019@gmail.com

Mobile: +8801970608472

Facebook: <https://www.facebook.com/m.omarfaruksiam/>

**PROJECT METHADOLGY AND USER MANUAL**

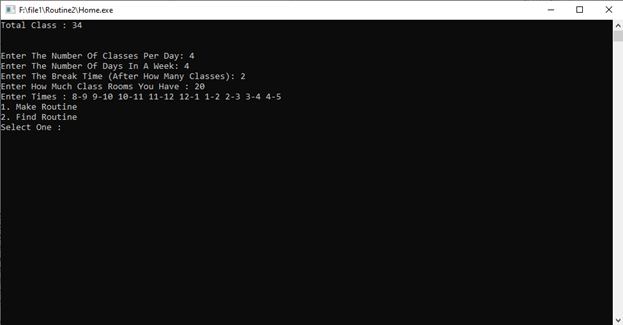
**Press “1” :**

**Enter Subjects information:**

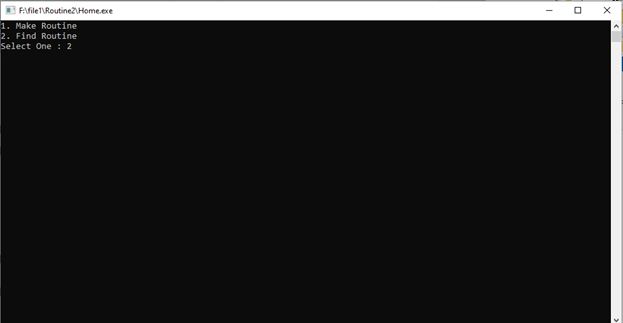
**Enter Students Classes Information:**

****

**Enter Your Class times As Your Wish:**

****

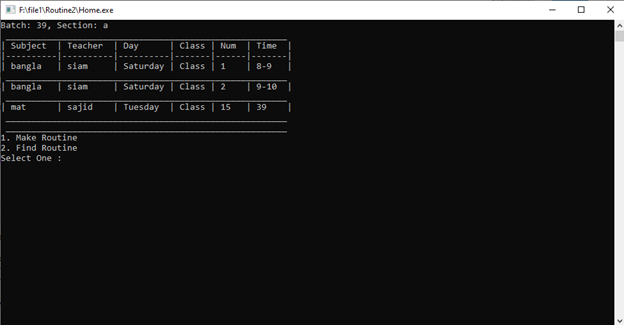
**Now Press “2”:**

****

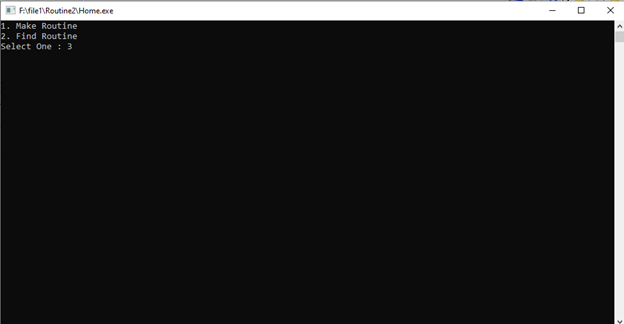
**Enter Your Batch And Section:**

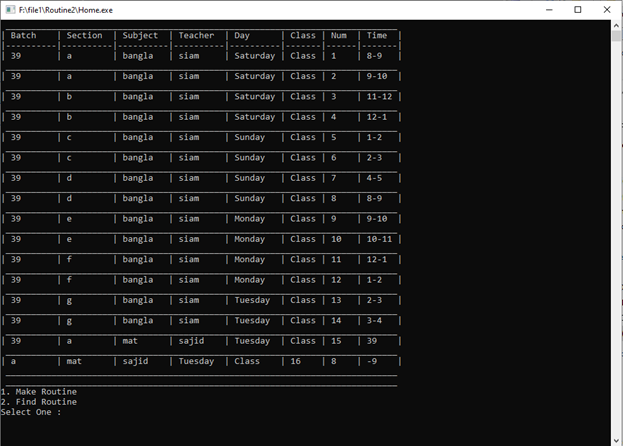
****

**See Your Routine:**

****

**Press “3” To See The Full Routine:**



**See Routine For All Sections:** 

**Notes**: If you press “0” then the data will be deleted. That means you will be not able to get the information again. And if you want to make another routine than save the “schedule.txt” file with another name. Than make a new routine.

**PRIORITIZATION:**

|  |  |  |
| --- | --- | --- |
| **Sl.No** | **Requirement** | **Dollar** |
| 1 | Add Subject | 5$ |
| 2 | Add Teacher | 10$ |
| 3 | Add Batch | 5$ |
| 4 | Add Section | 5$ |
| 5 | Add Class Per Day | 15$ |
| 6 | Add Classer Per Week | 15$ |
| 7 | Break Time | 5$ |
| 8 | Class Times | 10$ |
| 9 | Find Routine | 20$ |
| 10 | Find Full Routine | 10$ |
|  | **Total** | **100$** |

**USE CASE:**

|  |  |
| --- | --- |
| **Use case** | Add Subject |
| **Goal** | To add make the routine |
| **Preconditions** | N\A |
| **Success End Condition** | Subject Added |
| **Failed End Condition** | Error |
| **Primary Actors;** | Teacher, Student |
| **Secondary Actors** | Admin |
| **Description / Main Success Scenario** | Action |

|  |  |
| --- | --- |
| **Use case** | Add Section |
| **Goal** | To  make routine. |
| **Preconditions** | N\A |
| **Success End Condition** | Section Added |
| **Failed End Condition** | Error |
| **Primary Actors;** | Teacher, Student |
| **Secondary Actors** | Admin |
| **Description / Main Success Scenario** | Action |

|  |  |
| --- | --- |
| **Use case** | Add Class Per Day |
| **Goal** | To make routine. |
| **Preconditions** | N\A |
| **Success End Condition** | Added |
| **Failed End Condition** | Error |
| **Primary Actors;** | Teacher, Student |
| **Secondary Actors** | Admin |
| **Description / Main Success Scenario** | Action |

|  |  |
| --- | --- |
| **Use case** | Add Class Per Week |
| **Goal** | To  make routine. |
| **Preconditions** | N\A |
| **Success End Condition** | Added |
| **Failed End Condition** | Error |
| **Primary Actors;** | Teacher, Student |
| **Secondary Actors** | Admin |
| **Description / Main Success Scenario** | Action |

|  |  |
| --- | --- |
| **Use case** | Break Time |
| **Goal** | To  make routine. |
| **Preconditions** | N\A |
| **Success End Condition** | Added |
| **Failed End Condition** | Error |
| **Primary Actors;** | Teacher, Student |
| **Secondary Actors** | Admin |
| **Description / Main Success Scenario** | Action |

|  |  |
| --- | --- |
| **Use case** | Class Times |
| **Goal** | To  make routine. |
| **Preconditions** | N\A |
| **Success End Condition** | Added |
| **Failed End Condition** | Error |
| **Primary Actors;** | Teacher, Student |
| **Secondary Actors** | Admin |
| **Description / Main Success Scenario** | Action |

|  |  |
| --- | --- |
| **Use case** | Find Routine |
| **Goal** | To  make routine. |
| **Preconditions** | N\A |
| **Success End Condition** | Log in successfully, access the log in portal |
| **Failed End Condition** | Error |
| **Primary Actors;** | Teacher, Student |
| **Secondary Actors** | Admin |
| **Description / Main Success Scenario** | Action |

|  |  |
| --- | --- |
| **Use case** | Find Full Routine |
| **Goal** | To  make routine. |
| **Preconditions** | N\A |
| **Success End Condition** | Added |
| **Failed End Condition** | Error |
| **Primary Actors;** | Teacher, Student |
| **Secondary Actors** | Admin |
| **Description / Main Success Scenario** | Action |

**Use Case Diagram:**

Student

Teacher