

Scheduler : A Mobile Application

BY

NILOY KUMAR SARKER

ID: 191-15-12826

AND

SURAIYA TANZEEN

ID: 191-15-12458

This Report Presented in Partial Fulfillment of the Requirements for the
Degree of Bachelor of Science in Computer Science and Engineering

Supervised By

Saiful Islam

Assistant Professor

Department of CSE

Daffodil International University

Co-Supervised By

Md. Jueal Mia

Assistant Professor

Department of CSE

Daffodil International University



DAFFODIL INTERNATIONAL UNIVERSITY

DHAKA, BANGLADESH

JANUARY 2023

APPROVAL

This Project/internship titled “**Scheduler: A Mobile Application**”, submitted by Niloy Kumar Sarker, ID No: 191-15-12826 and Suraiya Tanzeen, ID No: 191-15-12458 to the Department of Computer Science and Engineering, Daffodil International University has been accepted as satisfactory for the partial fulfilment of the requirements for the degree of B.Sc. in Computer Science and Engineering and approved as to its style and contents. The presentation has been held on 28 January 2023.

BOARD OF EXAMINERS

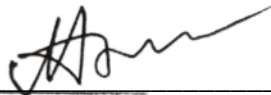


Chairman

Dr. Touhid Bhuiyan

Professor and Head

Department of Computer Science and Engineering
Faculty of Science & Information Technology
Daffodil International University



Nazmun Nessa Moon

Associate Professor

Department of Computer Science and Engineering
Faculty of Science & Information Technology
Daffodil International University



Internal Examiner

Zakia Sultana

Senior Lecturer

Department of Computer Science and Engineering
Faculty of Science & Information Technology
Daffodil International University



External Examiner

Dr. Shamim H Ripon


Professor

Department of Computer Science and Engineering
East West University

DECLARATION

We hereby declare that, this project has been done by us under the supervision of **Saiful Islam, Assistant Professor, Department of CSE** Daffodil International University. We also declare that neither this project nor any part of this project has been submitted elsewhere for award of any degree or diploma.

Supervised by:



Saiful Islam

Assistant Professor
Department of CSE
Daffodil International University

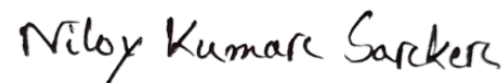
Co-Supervised by:



Md Jueal Mia

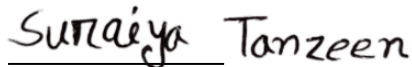
Assistant Professor
Department of CSE
Daffodil International University

Submitted by:



Niloy Kumar Sarker

ID: 191-15-12826
Department of CSE
Daffodil International University



Suraiya Tanzeen

ID: 191-15-12458
Department of CSE
Daffodil International University

ACKNOWLEDGEMENT

First we express our heartiest thanks and gratefulness to almighty God for His divine blessing makes us possible to complete the final year project/internship successfully.

We really grateful and wish our profound our indebtedness to **Saiful Islam, Assistant Professor**, Department of CSE Daffodil International University, Dhaka. Deep Knowledge & keen interest of our supervisor in the field of **Android Application Development** to carry out this project. His endless patience ,scholarly guidance ,continual encouragement , constant and energetic supervision, constructive criticism , valuable advice ,reading many inferior draft and correcting them at all stage have made it possible to complete this project.

We would like to express our heartiest gratitude to **Professor Dr. Touhid Bhuiyan, Professor** and Head, Department of CSE, for his kind help to finish our project and also to other faculty member and the staff of CSE department of Daffodil International University.

We would like to thank our entire course mate in Daffodil International University, who took part in this discuss while completing the course work.

Finally, we must acknowledge with due respect the constant support and patients of our parents.

ABSTRACT

The daily routine of a class eventually determines the overall potential of an academic institute like schools, colleges, and universities. The wide range of techniques and methods teachers use to keep their students on track and academically productive throughout a lesson or session are referred to as "classroom management." The majority of educational institutions still use a manual procedure to generate class schedules. As a result, there is a great chance of making mistakes during an overlapped timetable period or slot, and a lot of time is lost because there are so many things to keep in mind. We have analyzed and implemented **Scheduler** to manage class routines for any university or department. This project intends to provide a better and more dynamic schedule management service as well as reduce the admin panel's manual work that involves manually adding each section's class schedule one at a time. One Android mobile application will be able to control the class schedule for any department. According to Business of Apps 2022 figures, Android is a widely used platform with over 2.5 billion active users in more than 190 countries. It is not surprising that there is a constant demand for Android app development given the rapid advancement of new technology. The system will set itself up after the user or routine management team creates a batch schedule. By using a special key, users can share their routine using an inbuilt sharing feature.

TABLE OF CONTENTS

CONTENTS	PAGE
Board of examiners	ii
Declaration	iii
Acknowledgements	iv
Abstract	v
Table of Contents	vi
List of Figures	viii
List of Tables	ix
CHAPTER	
CHAPTER 1: INTRODUCTION	1-2
1.1 Introduction	1
1.2 Motivation	1
1.3 Objectives	1
1.4 Expected Outcomes	1
1.5 Project Management and Finance	2
1.6 Report Layout	2
CHAPTER 2: BACKGROUND	3-4
2.1 Preliminaries/Terminologies	3
2.2 Related Works	3
2.3 Comparative Analysis	3
2.4 Scope of the Problem	4
2.5 Challenges	4

CHAPTER 3: REQUIREMENTS SPECIFICATION	5-10
3.1 Business Process Modeling	5
3.2 Requirements Collection and Analysis	6
3.3 Use Case Modeling and Description	6
3.4 Logical Data Model	9
3.5 Design Requirement	10
CHAPTER 4: DESIGN SPECIFICATION	11-12
4.1 Front-end Design	11
4.2 Back-end Design	11
4.3 Interaction Design and User Experience (UX)	11
4.4 Implementation Requirements	11
CHAPTER 5: IMPLEMENTATION AND TESTING	13-22
5.1 Implementation of Database	13
5.2 Implementation of Front-end Design	15
5.3 Testing Implementation	20
5.4 Test Result and Reports	20
CHAPTER 6: CONCLUSION AND FUTURE SCOPE	23
6.1 Discussion and Conclusion	23
6.2 Scope for Future Developments	23
APPENDIX	24-25
REFERENCES	26
PLAGIARISM REPORT	27

LIST OF FIGURES

FIGURES	PAGE NO
Figure 3.1.1: BPM of Student Activity	5
Figure 3.3.1: Use Case of Scheduler Mobile App	6
Figure 3.4.1: ERD for Scheduler Mobile App	9
Figure 5.1.1: Firebase Authentication Method	13
Figure 5.1.2: List of registered users	14
Figure 5.1.3: Firebase's database architecture	14
Figure 5.1.4: User Information on Firebase	15
Figure 5.2.1: Splash Screen	15
Figure 5.2.2: Login and Signup Screen	16
Figure 5.2.3: Routine Screen	16
Figure 5.2.4: Task Screen	17
Figure 5.2.5: More Screen	17
Figure 5.2.6: Add Routine Screen	18
Figure 5.2.7: Manage Routine Screen	18
Figure 5.2.8: Routine Key Screen	19
Figure 5.2.9: Import Routine Screen	19

LIST OF TABLES

TABLES	PAGE NO
Table 3.3.1: Description of Login Use Case	7
Table 3.3.2: Description of Add Routine Use Case	7
Table 3.3.3: Description of View Routine Use Case	7
Table 3.3.4: Description of Manage Routine Use Case	8
Table 3.3.5: Description of Add Daily Task Use Case	8
Table 3.3.6: Description of View Task Use Case	8
Table 3.3.7: Description of Import Routine Use Case	9
Table 5.1: Test for Compatibility	21

CHAPTER 1

INTRODUCTION

1.1 Introduction

The "Scheduler" is indeed a routine management tool available to students at any college. With an in-built sharing option, users can share their routine through the usage of a unique key. A notification is issued to all subscribers each time the host modifies the routine. Without changing the routine, users can adjust it to their requirements and preferences. It can be used for daily task reminders and adding future tasks such as quizzes, assignments, and presentation dates. The application will remind you easily through notifications.

1.2 Motivation

To prepare ourselves for unexpected tasks such as extra classes, quizzes, and assignments along with other tasks, we need to predict the sudden tasks that may pop up. Thus, it's much easier to craft contingencies when you're well organized. We have faced such problems during our whole life so we can relate to them properly and that's why we came up with the solution which is our application "Scheduler".

1.3 Objectives

We aim to build an application program that would help a student to manage their class routine where they can add their class routine as well as daily tasks like homework, upcoming quiz, and assignments. This project also aims to reduce the manual work for the admin panel that adds every section's class routine one by one.

1.4 Expected Outcomes

We have told that we want to make a class routine manager hassle free. It's not only managed the class schedule but also remind you your upcoming quiz, assignment and presentation. The main purposes of this project are:

- Provide student a well-maintained class schedule
- Remind about classes before five minutes to get prepare for that class
- Add upcoming task such as Quiz, Assignment or Presentation
- Show monthly progress of individual according their task status
- Reduce manual work for admin panels. User will play the admin role

1.5 Project Management and Finance

For a system to function effectively, maintenance is required. And it costs money to cover those maintenance expenses. The maintenance and other costs are covered by user donations since the app is non-profit. To keep our software running, we hope that people will be inspired to make donations to us.

1.6 Report Layout

- The background, motivation, aim, anticipated outcomes, management, and funding of the project are briefly summarized in Chapter 1.
- Background, related activities, and issues are covered in Chapter 2.
- Chapter 3: requirements analysis and requirements definition for this project, with use case models, business case modelling, and logical data models.
- Chapter 4: The development and implementation of this project's components.
- The implementation of the project's front end, back end, and testing strategy are all covered in Chapter 5.
- Chapter 6: This chapter includes suggestions for future research breakthroughs and finishes the study.

CHAPTER 2 BACKGROUND

2.1 Preliminaries/Terminologies

The app "Scheduler" manages class routines. First of all, our app is not department dependent. Any department student even any varsity student can use our app to manage their routine. Also, our app provides a routine share option. With a unique routine key, any student can share their routine with each other. Suppose you are a retake student and want to add your retake class routine, it is possible in our app and regular routine won't be replaced. Then any student can add their daily task as well as the upcoming quiz, assignment, and presentation date easily. and can get to remember with a notification as well. We want to make this app available to everyone with a great user experience.

2.2 Related Works

We are not claiming to be the only ones who make this concept a reality. A similar type of mobile application is also available. We researched some of them like **Class Organizer, Class Manager, Task Planer**. But there are some limitations on those applications. Such applications perform a single task only. Additionally, some of them are not the most convenient or user-friendly. To make the process of routine management incredibly simple and practical, we have merged a variety of functions and improved its usability and effectiveness.

2.3 Comparative Works

One of the popular routine management apps is called "Class Manager." It does, however, have some limitations, such as being used by only one department. "Scheduler" is different from this app. We eliminate department dependency and make it available for any department at any university. It is designed to manage class routines and daily tasks efficiently. One of the best features of our app is that it provides a routine sharing option. With a unique routine key, any student can share their routine with each other. That's why it doesn't require an admin panel to maintain the routine of every section. In an effort to increase user engagement, we put a lot of effort into improving its usability.

2.4 Scope of the Problem

The goal of "Scheduler" is to create new and inventive ways to manage classes. We concentrated on making the routine system easier to use and more efficient for users. Both students without a science background and those with a background in computer science will benefit from this initiative. We propose a very user-friendly app, the user can easily organize their daily class schedule or daily tasks. As there is no specific admin panel, users play the admin role as well as viewers. Users can share their class routines with their friends by using a special key.

2.5 Challenges

Making an idea into reality is not easy. There have to be a lot of difficulties. Also, we have faced some difficulties in solving this real-life problem while making the application. We collect data from the users of similar applications and find out their limitations. Then we are working on that. Some limitations are:

- System Design
- For smooth performance we switch to Kotlin programming language
- Keep user data secured with firebase authentication
- Make only for android user

CHAPTER 3 REQUIREMENT SPECIFICATION

3.1 About Business Process Modeling

Business process modeling is a tool for charting our regular business processes and determining how to improve them. It is a component of business process management. It indicates how the process exactly works. BPM gives you a comprehensive understanding of your concept. BPM also helps your application maintain consistency. It reduces the capacity for waste. BPM aids the project in reducing costs, increasing speed, and improving the process's nature.

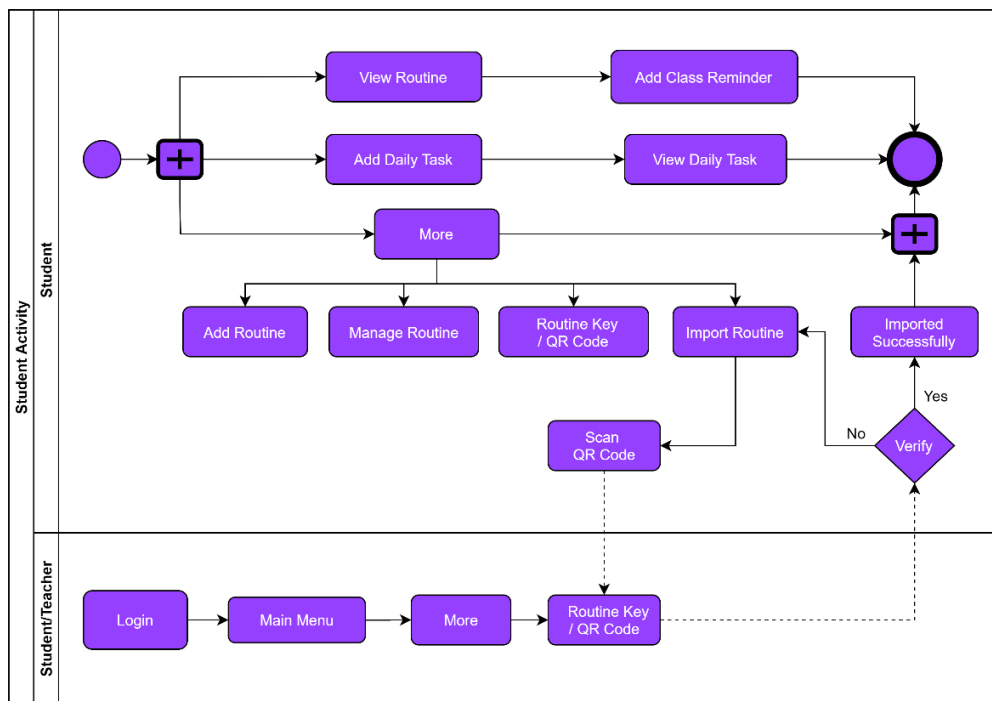


Figure 3.1.1: BPM of Student Activity

We displayed the user activity scenario in figure 3.1.1. Here the user can add their routine as well as daily or upcoming tasks such as quizzes, assignments, and presentations. Also, users can add class reminders. A user can share their routine with others, and others can import a user’s routine easily by entering or scanning a routine key. If someone enters an invalid key, then they can’t import a routine and move to the import routine page.

3.2 Requirements Collection and Analysis

We now have a general understanding of the requirements depending on the discussion up to this point. Here are some of them:

We require a solution that enables users to effectively manage their class schedules and upcoming tasks. We also added a feature where users can see their monthly progress on their upcoming tasks. Obviously, the user has access to these capabilities after logging in.

As we mentioned previously, we don't have an admin panel. Every user can take on the role of administrator here. After adding a class schedule, users can easily share this routine with their friends or others.

3.3 Use Case Modeling and Description

Use case model are one sort of scenario that links your framework or system to customers, your business, and external systems. Users and applications typically have the ability to easily condense the entire project for the user in a variety of use situations.

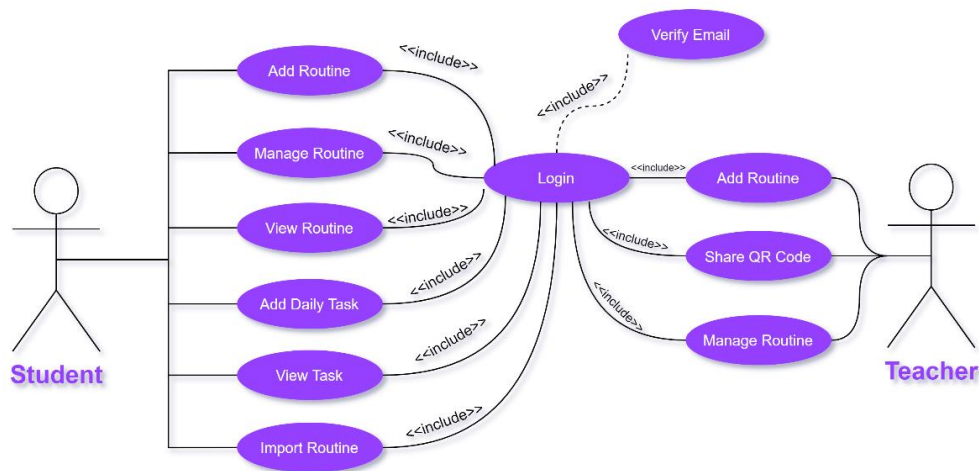


Figure 3.3.1: Use Case of "Scheduler Mobile App"

The following use case description is –

Table 3.3.1 describes the user login methodology. Here, the primary actor is a student. To login, students have to be registered first; otherwise, they can't login. A verified account is required for student login.

Table 3.3.1: Description of Login Use Case

Use case	Login
Primary Actor	Student
Secondary Actor	None
Pre-Condition	User have to be Registered
Primary Path	Verified Account
Exceptional Path	Can't log in without authentication

Table 3.3.2 describes how a student can add routines. Here, the primary actor is a student. To add a routine student, you must first log in. To add a routine student, you must fill out a form with accurate information. The field cannot be empty.

Table 3.3.2: Description of Add Routine Use Case

Use case	Add Routine
Primary Actor	User
Secondary Actor	None
Pre-Condition	User have to be Logged in
Primary Path	Add Class Routine according to day of a week
Exceptional Path	All field must be filled

Table 3.3.3 describes how a student can view a routine. To view routine students, you must first add routine. For the first time, an internet connection is required.

Table 3.3.3: Description of View Routine Use Case

Use case	View Routine
Primary Actor	User
Secondary Actor	None
Pre-Condition	User have to add routine
Primary Path	View added routine by day of a week
Exceptional Path	No routine will show if no routine was added or no internet connection

Table 3.3.4 describes how a student can edit or delete their routine. If there is no routine, then students don't need to edit or delete it.

Table 3.3.4: Description of Manage Routine Use Case

Use case	Manage Routine
Primary Actor	User
Secondary Actor	None
Pre-Condition	User have to add routine
Primary Path	User can edit or delete routine data
Exceptional Path	No routine will show if no routine was added or no internet connection

Table 3.3.5 explains how a student can add upcoming tasks, such as quizzes, assignments, and presentations, after logging in.

Table 3.3.5: Description of Add Daily Task Use Case

Use case	Add Daily Task
Primary Actor	User
Secondary Actor	None
Pre-Condition	User have to be Logged in
Primary Path	Add upcoming quiz, assignment or any task
Exceptional Path	All field must be filled

Table 3.3.6 describes how a student can view their daily tasks. In the task section, students can view their added tasks automatically. First-time viewers need an internet connection.

Table 3.3.6: Description of View Task Use Case

Use case	View Task
Primary Actor	User
Secondary Actor	None
Pre-Condition	User have to add task
Primary Path	View daily task

Exceptional Path	No task will show if no task was added or no internet connection
------------------	--

Table 3.3.7 describes how a student can import a routine from others. There are two ways to import routine students. Either use a routine key or scan a QR code, which includes a routine key.

Table 3.3.7: Description of Import Routine Use Case

Use case	Import Routine
Primary Actor	User 1
Secondary Actor	User 2
Pre-Condition	User have to be a routine key
Primary Path	After scanning or entering routine key user able to import others routine successfully
Exceptional Path	Invalid routine key

3.4 Logical Data Model

An E-R diagram allows a developer to see the entire database. It indicates a connection to a group of entities maintained in a database. In other words, ER graphs are used to explain database logic. Entities, attributes, and linkages are the three core principles that define ER diagrams. The E-R model included into our project:

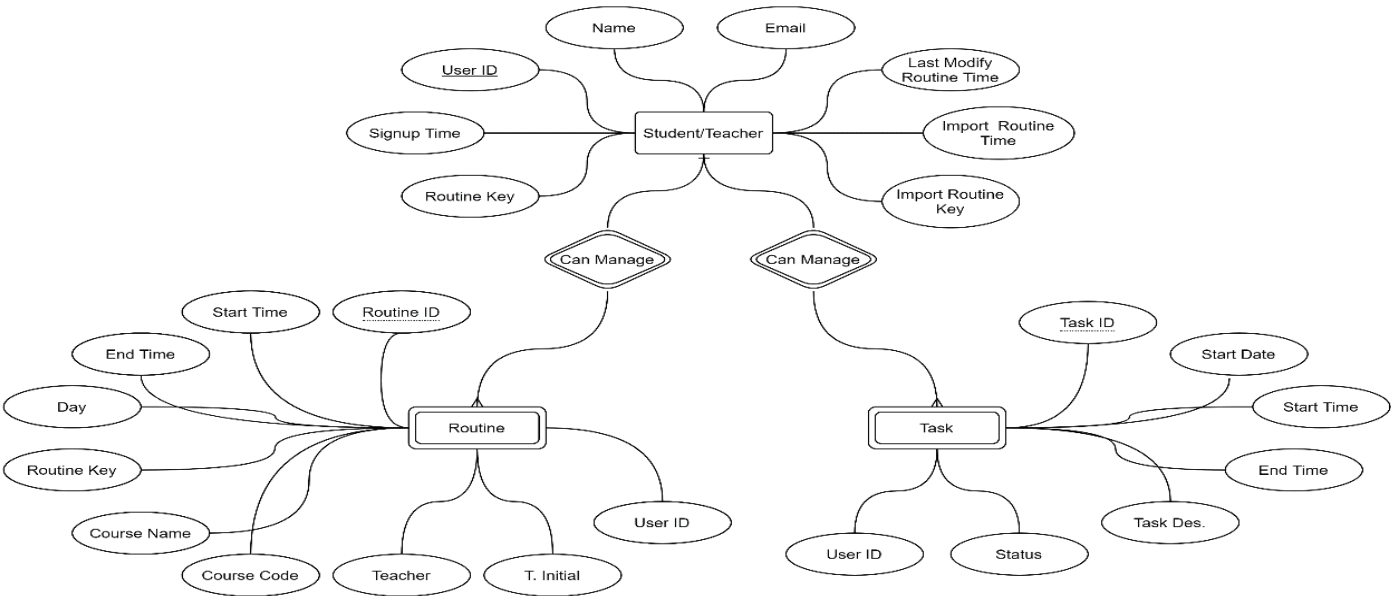


Figure 3.4.1: ERD for Scheduler Mobile App

In figure 3.4.1, a student/teacher can manage routine and task both. 'Routine' & 'Task' are the weak entity. 'Can Manage' is the identifying relationship and 'Routine ID' & 'Task ID' are the partial key.

3.5 Design Requirement

The development of the "Scheduler" application has various prerequisites. They are specifications for both the hardware and software.

Hardware Requirements:

- An Intel Core i5 processor or above.
- 8GB or more of RAM.
- A 500GB or larger hard drive.
- A screen with a resolution of 1080*720 or greater is necessary.
- Use of the Internet is necessary.

Software Requirements:

- Windows as operating system.
- Database as Firebase.
- For editing, use Sublime Text.
- View Android Emulator.

Language

- Kotlin

Tools

- Android Studio

CHAPTER 4 DESIGN SPECIFICATION

4.1 Front-end Design

A mobile application's graphical user interface is considered its front-end. We do adhere to several design principles in order to maintain the app user-friendly, eye catching, and relatively simple. The application was kept as simple, attractive, and user-friendly as possible. In light of this, we concentrated on the application's usability and effectiveness from the user's point of view.

A splash screen appears when the app first opens. The user will then be required to log in or register after that. To access the application for the first time, a user must first register. Once logged in, they can use the credential to keep using the app. The registration and login procedures have both been kept simple.

4.2 Back-end Design

A mobile app's database serves as its major backend. It requires frequent maintenance and observation. The back-end procedure of our project must store all data. We processed and stored data for our project using the "Google Firebase Database." It offers authentication, real-time data, a firestore database, cloud service, and a few extra features. Also, we follow a design pattern for the backend called MVVM. MVVM stands for "Model, View, and View Model." Maintaining progress is simple with this pattern.

4.3 Interaction Design and User Experience (UX)

A user-friendly application would be engaging for users. To improve its usability, we have given the UI additional thought. Security is one of the key problems with every software. Security for users has been ensured. We have secured the system and designed the procedure as simple as possible to avoid users being discouraged by providing an email account and password.

4.4 Implementation Requirements

We have used a variety of tools to complete the project. Some of them are listed below:

4.4.1 Adobe XD

Adobe XD is a vector design tool for web and mobile applications, developed and published by Adobe Inc. It is available for macOS and Windows, and there are versions for iOS and Android to help preview the result of work directly on mobile devices. Screen design, illustration, and prototyping are all done with Adobe XD. Prior to implementation, it is beneficial to understand an application's output.

4.4.2 Android Studio

The official Integrated Development Environment (IDE) for Google's Android operating system, built on IntelliJ IDEA from JetBrains, is known as Android Studio. On Windows, macOS, and Linux-based operating systems, it can be downloaded. Additionally, there are numerous different IDEs for developing Android applications; however, some fantastic features are offered by Android Studio.

4.4.3 SDK

An installable package of software development tools is known as a software development kit. Compiler, debugger, and sometimes a software framework are included to simplify the development of applications. A collection of development tools is included in the Android SDK for creating Android apps. Along with other APIs, it has a number of libraries.

4.4.4 Firebase Database

Firebase provides a range of hosting options for various applications. Additionally, it provides other services, such as real-time communication servers from Google Inc., as well as NoSQL and real-time databases, contents, login authentication, and notification hosting.

CHAPTER 5 IMPLEMENTATION AND TESTING

5.1 Implementation of Database

Now talk about how we're going to integrate a database into our project. Our primary database, Firebase, was stated earlier. For safe database access, Firebase supports a number of authentication mechanisms. With a validated email address and password, it offers a safe and secure method of authentication. Data modifications are easy with Firebase's Firestore database capability. A hierarchical JSON-like structure is used to save data in Firebase. To guarantee secure access, it comes with a few security guidelines.

Additionally, Firebase provides a few other methods of authentication. In Figure: 5.1.1, For our project, however, we'll be using Gmail and email passwords for user-provided authentication, and the verification is being handled by Firebase.

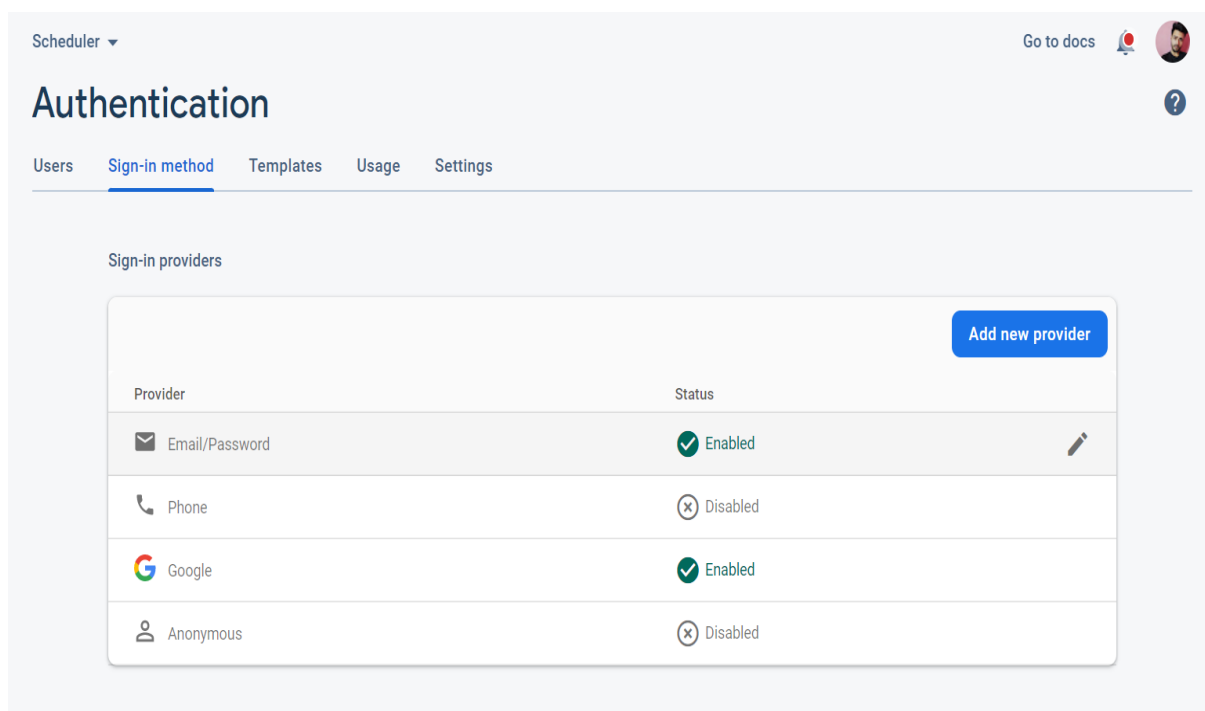


Figure 5.1.1: Firebase Authentication Method

Based on the distinct user ID and registration date, Firebase may show the details of a registered user. Additionally, in figure 5.1.2, details about the earlier sign-in will be shown.

Identifier	Providers	Created ↓	Signed in	User UID
kurtwilkinson.72383@gma...		Nov 26, 2022	Nov 26, 2022	4lvrK7imLvhdFngVKnu771zqZak1
masud@gmail.com		Nov 2, 2022	Nov 2, 2022	npPipmh2qSMl7iugHmvY0ws8J9t1
test2@gmail.com		Sep 12, 2022	Sep 13, 2022	vu6s6DhbsZXT6vTrJIGsqhCu3F02
test@gmail.com		Aug 14, 2022	Dec 27, 2022	iYpulM3z57PILGoKOWmOCJWe54...
niloy64529@gmail.com		Aug 13, 2022	Dec 1, 2022	eOYGUIT5CwN7umcWe8LvTFIvJN...

Figure 5.1.2: List of registered users

To be authorized, users must enter their email address and password. They'll receive verification instructions from our system. Users can use a validated email and password to log in to our system the following time. In Figure: 5.1.3, Our application data is kept in a cloud-based database. In the database, the data will be organized in a JSON structure.

Collection	Document	JSON Structure
Task	h3yUPoCMMpTG1uZy1022	<pre> { "endTime": 1669825800000, "startDate": 1669744800000, "startTime": 1669809600000, "status": "pending", "taskDes": "Make Pre-defense Slide", "taskId": "h3yUPoCMMpTG1uZy1022", "userId": "eOYGUIT5CwN7umcWe8LvTFIvJNm2" } </pre>

Figure 5.1.3: Firebase's database architecture

In Figure: 5.1.4, The rest of the user's information is preserved with the user's unique user id.

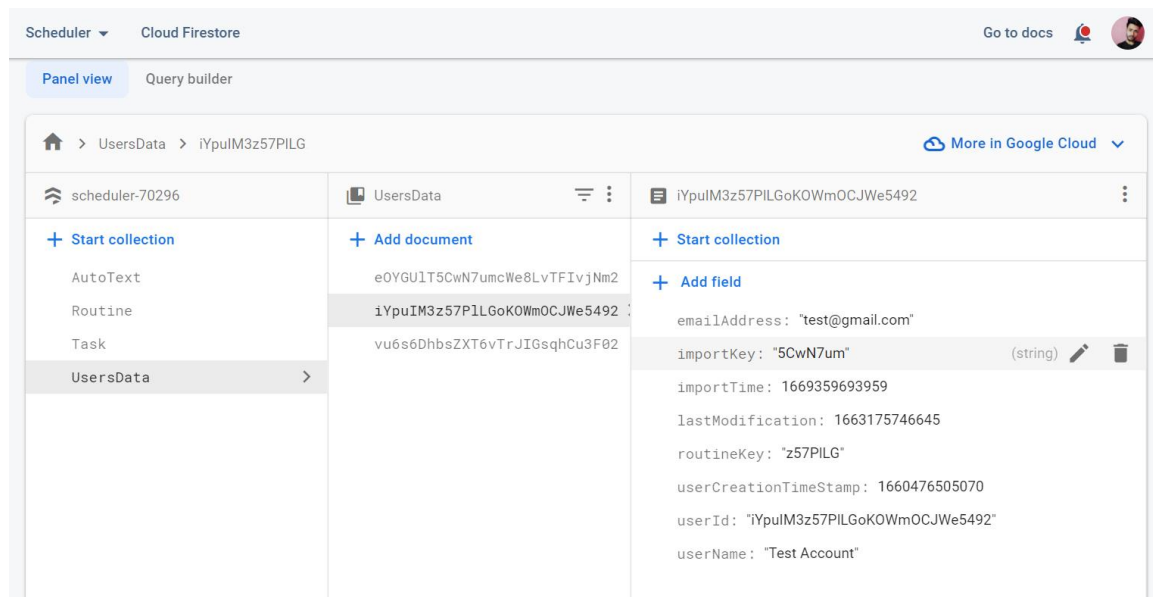


Figure 5.1.4: User Information on Firebase

5.2 Implementation of Front-end Design

A mobile app's front-end design is always essential. To continue using the software, users must like the user interface (UI). The following describes the proposed application's front-end design-

In Figure: 5.2.1, When users open the app, a splash screen with the title "Scheduler" and the app's logo will appear.

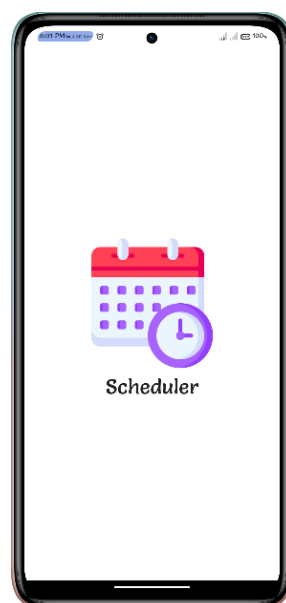


Figure 5.2.1: Splash Screen

In Figure: 5.2.2, Then user will get the option to log in. If the user has not previously created an account, they must do so before logging in.

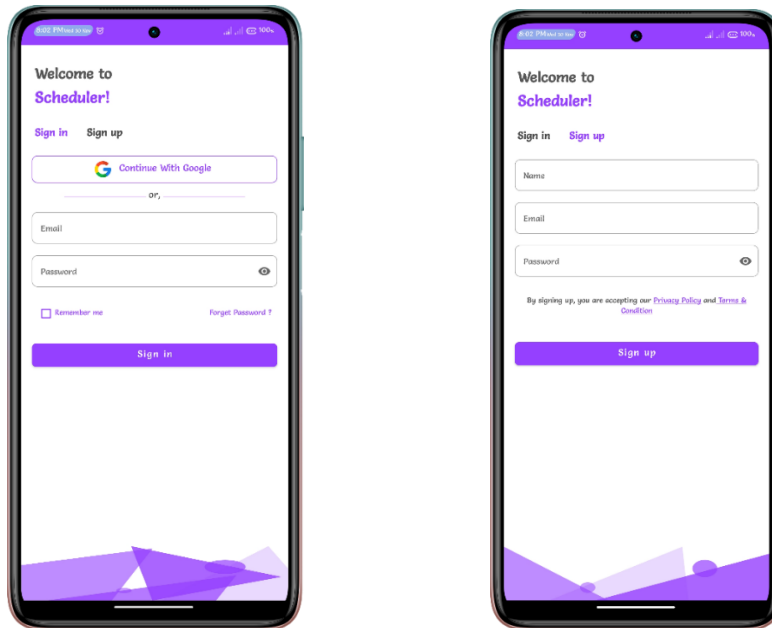


Figure 5.2.2: Login and Signup Screen

In Figure: 5.2.3, It will take the user to the routine page after a successful login.

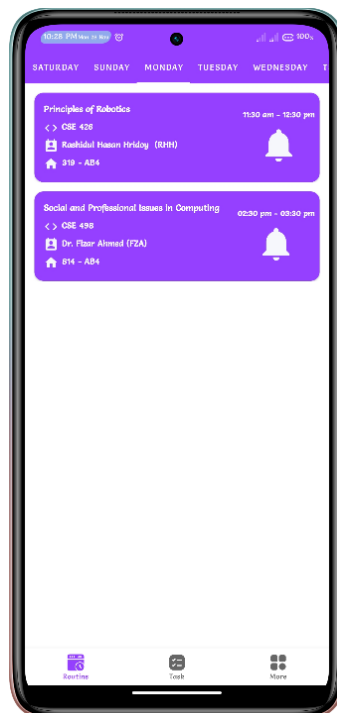


Figure 5.2.3: Routine Screen

In Figure: 5.2.4, The user will land at the Task page after clicking the Task button. Here they can add their daily tasks like assignments, quizzes, presentations, etc. and get a reminder of those tasks. They can view all of the tasks they added after adding them. They can also see the monthly progress. They can also change the status of a task by swiping it left or right.

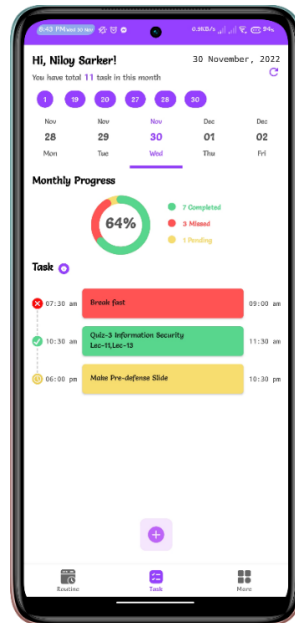


Figure 5.2.4: Task Screen

In Figure: 5.2.5, The user will be taken to the More page after clicking the "more" button. Here they can see multiple option to do.

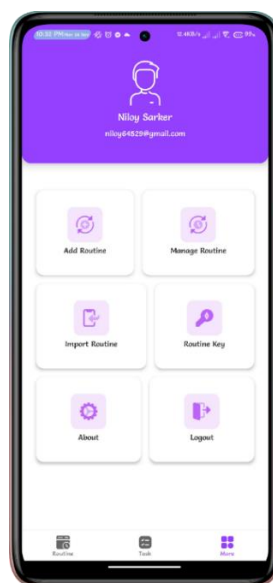


Figure 5.2.5: More Screen

In Figure: 5.2.6, Once the Add Routine button has been clicked, the user will be taken towards the Add Routine page. Here they can add their routine according to the day of the week.

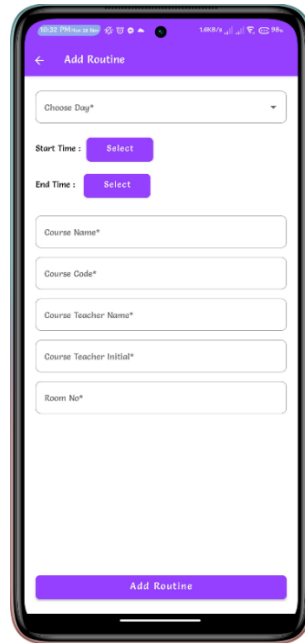


Figure 5.2.6: Add Routine Screen

In Figure: 5.2.7, A user can access the Manage Routine page by clicking the Manage Routine button. Here they can Update or Delete their routine according to the day of the week.

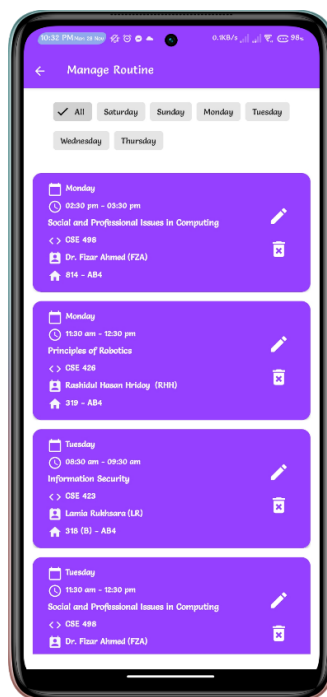


Figure 5.2.7: Manage Routine Screen

In Figure: 5.2.8, Clicking on the Routine Key button will show the user a unique routine key with a QR code. They are free to distribute this QR code or key to anyone.

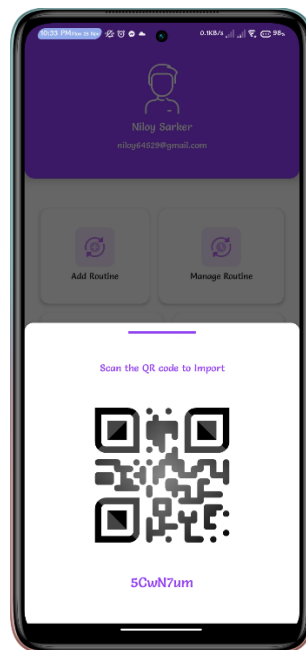


Figure 5.2.8: Routine Key Screen

In Figure: 5.2.9, Clicking on the Import Routine button will show the user two option. They can import routine either scan QR code or enter routine key.

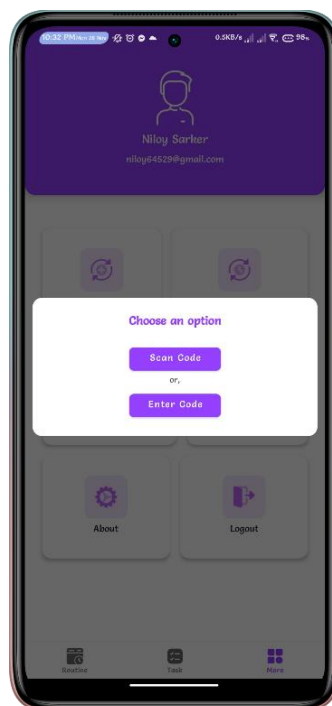


Figure 5.2.9: Import Routine Screen

5.3 Testing Implementation

The project's success was measured using the following criteria:

Requirement Specifications That Are Met: A project is considered successful if it satisfies both functional and non-functional needs. It must be able to ensure that the requirements are met, in other words. Accuracy is one of the most important requirements for software development. For service-oriented software to work, perfectionism is necessary. Each part of the software must operate accurately and properly.

Compatibility and Integrity: These are the two main factors that influence whether or not a project will succeed. Every Android version can use the sophisticated class project. Additionally, how well the application was integrated into the larger system was a factor in how well the system was evaluated.

Management of reliability and security: Any service-oriented system needs security as one of its most crucial components. As a result, the design of the system takes the evaluation criteria for the security features into consideration.

User-friendliness: Every application has a different criterion for evaluating systems, and user-friendliness is one of them. For instance, users must be satisfied with the result when using this solution. Effectiveness, portability, reusability, adaptability, integrity, and loose connection among the numerous developed software components are just a few qualities that a system should generally possess.

5.4 Test Result and Reports

System weaknesses are regularly discovered via software tests. By looking at the program's coding, design, and execution as a whole, a software test can be created. It's imperative to test systems to improve system quality. Another facet of software engineering that's also occasionally disregarded while a project is being developed is the fundamental procedures of the review process and testing. a critical stage in the creation of the software testing methodology.

5.4.1 Testing of Functionality

- A tester must ensure that the app satisfies every need stated by the user and detailed in the additional constraint system when conducting functionality testing.
- Two types of functional testing exist.
- Positive functional testing involves two things: checking the accuracy of the outputs and validating the application's operation.
- A program is put through a range of faulty inputs, unanticipated operational scenarios, and other "out-of-bounds" circumstances during negative functional testing.

5.4.2 Unit Test

During the phases of extensive design and implementation, unit testing is frequently used. Unit testing served the objective of finding the project's problems.

5.4.3 Test for Compatibility

A non-functional software test called compatibility testing determines how well an application works in a particular computing environment. The setting for assessing software compatibility seems to be more suited to the consumer experience. Testing on various Android phones has confirmed the following:

Table 5.1: Compatibility Testing

Device	Display Size	Run The Test	Test Result
Xiaomi Redmi Note 10 Pro Max	6.67"	Yes	Successful
Xiaomi Redmi Note 11 Pro Max	6.67"	Yes	Successful
Pixel 6	6.4"	Yes	Successful
Xiaomi Redmi Note 8	6.3"	Yes	Successful

Samsung Galaxy J6	5.6"	Yes	Successful
Samsung Galaxy S8	5.8"	Yes	Successful

Table 5.1 describes the number of tests on different mobiles. The app runs smoothly on all mobile devices. The app easily interacts with the different screen sizes.

CHAPTER 6

CONCLUSION AND FUTURE SCOPE

6.1 Discussion and Conclusion

Our application's final objective is to reach a certain level which will be the perfect counter for unexpected tasks such as extra classes, quizzes, and assignments, along with other tasks. None can assure that our project will have the highest chance of success, neither can say that it might fell flat. So, we think our project depends on the steps we will take, the decisions we will make, the hard work we will do, and most important, our own determination.

6.2 Scope for Future Developments

It looks positive that "Scheduler" will continue to be a useful and well-liked tool.

Several potential directions for future development are as follows:

- Release on both Android & iOS
- View routine without user login
- Host user make reminder for all
- Extra panel only for teacher
- OCR for Add Routine

APPENDEX

Appendices A

Projection Reflection: We started our journey to create a routine management as well as task management app for student in the Summer 2022 semester. We try to keep the app very simple and user friendly, and we were successful in achieving our goal.

Summary of Our Project

1. Motivation: To prepare ourselves for unexpected tasks such as extra classes, quizzes, and assignments, along with other tasks, we need to predict the sudden tasks that may pop up.
2. Objectives: Our aim is to build an application program that would help a student manage their class routine, where they can add their class routine as well as daily tasks like homework, an upcoming quiz, and assignments. This project also aims to reduce the manual work for the admin panel that adds every section's class routine one by one.
3. Challenges: We face a little problem while implementing the routine sharing option. also face difficulties in report writing.
4. Related works: There are already apps of this type. But they have some limitations. We will remove those limitations and improve our app.
5. Requirement specification: In this phase, we first draw a business process model. Following analysis, we create a use-case diagram and an ERD. It enhances the speed of our work.

Appendices B

DESIGN

For design we use adobe XD. Here use initially design the UI of our app for better understanding.

DATABASE

We use firebase for store our data. Firebase provides a few other methods of authentication. We use email & password and gmail authentication system.

Android Studio

It is a tools for developing android app. On Windows, macOS, and Linux-based operating systems, it can be downloaded.

Kotlin

We use Kotlin as a programming language. What is the Kotlin used for?

Kotlin is a modern statically typed programming language used by over 60% of professional Android developers that helps boost productivity, developer satisfaction, and code safety.

REFERENCES

- [1] Class Manager DIU CSE, available at << <https://play.google.com/store/apps/details?id=bd.edu.daffodilvarsity.classmanager>>>, last accessed on 17-10- 2022 at 07:53 PM.
- [2] Google Keep, available at <<<https://play.google.com/store/apps/details?id=com.google.android.keep>>>, last accessed on 17-10- 2022 at 08:04 PM.
- [3] Habito, available at <<<https://play.google.com/store/apps/details?id=com.sarker.habitbreaker>>>, last accessed on 17-10- 2022 at 08:04 PM.
- [4] Feasibility Study, available at << <https://www.projectmanager.com/training/how-to-conduct-a-feasibility-study>>>, last accessed on 22-10- 2022 at 09:54 PM.
- [5] Learn Android for Developers, available at: <<<https://developer.android.com>>>, last accessed on 15-11-2022 at 10:48 AM.
- [6] Learn Firebase Database, available at <<<https://firebase.google.com>>>, last accessed on 17-11- 2022 at 08:29 AM.
- [7] Learn Google Play, available at <<<https://play.google.com>>>, last accessed on 26-11-2022 at 08:12 PM.
- [8] Learn Adobe XD, available at <<<https://www.adobe.com>>>, last accessed on 28-11-2022 at 11:49 PM.
- [9] Android Studio, available at <<https://en.wikipedia.org/wiki/Android_Studio>>, last accessed on 05-12- 2022 at 10:11PM.
- [10] Use Case Diagram, available at <<https://en.wikipedia.org/wiki/Use_case_diagram>> last accessed on 10-12- 2022 at 08:21 PM.
- [11] Entity Relationship Model, available at <<<https://www.javatpoint.com/dbms-er-model-concept>>> last accessed on 12-12- 2022 at 08:25PM.
- [12] Flowchart in Programming, available at << <https://www.programiz.com/article/flowchart-programming>>>, last accessed on 15-12- 2022 at 09:30PM

Turnitin Originality Report

Processed on: 07-Jan-2023 10:33 +06

ID: 1989425326

Word Count: 5768

Submitted: 1

Scheduler : A Mobile Application By Niloy
Kumar Sarker

Similarity Index

26%

Similarity by Source

Internet Sources: 24%
Publications: 2%
Student Papers: 20%

3% match (Internet from 21-Nov-2022)

[http://dspace.daffodilvarsity.edu.bd:8080/bitstream/handle/20.500.11948/3449/P10957.pdf?
isAllowed=y&sequence=2](http://dspace.daffodilvarsity.edu.bd:8080/bitstream/handle/20.500.11948/3449/P10957.pdf?isAllowed=y&sequence=2)

2% match (student papers from 07-Apr-2018)

[Submitted to Daffodil International University on 2018-04-07](#)

2% match (student papers from 19-May-2022)

[Submitted to Daffodil International University on 2022-05-19](#)

1% match (Internet from 14-Feb-2022)

[http://dspace.daffodilvarsity.edu.bd:8080/bitstream/handle/123456789/7129/171-15-9197%20%2817%25%29.pdf?
isAllowed=y&sequence=1](http://dspace.daffodilvarsity.edu.bd:8080/bitstream/handle/123456789/7129/171-15-9197%20%2817%25%29.pdf?isAllowed=y&sequence=1)

1% match (Internet from 21-Nov-2022)

[http://dspace.daffodilvarsity.edu.bd:8080/bitstream/handle/123456789/8629/181-15-11165.pdf?
isAllowed=y&sequence=1](http://dspace.daffodilvarsity.edu.bd:8080/bitstream/handle/123456789/8629/181-15-11165.pdf?isAllowed=y&sequence=1)

1% match (Internet from 23-Feb-2022)

[http://dspace.daffodilvarsity.edu.bd:8080/bitstream/handle/123456789/7123/171-15-9341%20%286%25%29.pdf?
isAllowed=y&sequence=1](http://dspace.daffodilvarsity.edu.bd:8080/bitstream/handle/123456789/7123/171-15-9341%20%286%25%29.pdf?isAllowed=y&sequence=1)

1% match (student papers from 30-Mar-2022)

[Submitted to Daffodil International University on 2022-03-30](#)

1% match (student papers from 12-May-2022)