DWIT COLLEGE

DEERWALK INSTITUTE OF TECHNOLOGY

Tribhuvan University

Institute of Science and Technology

ROUTINE MANAGEMENT SYSTEM

A PROJECT REPORT

Submitted to

Department of Computer Science and Information Technology

DWIT College

In partial fulfillment of the requirements for the Bachelor's Degree in Computer Science and Information Technology

Submitted by

Anil Lama

August, 2016

DWIT College

DEERWALK INSTITUTE OF TECHNOLOGY

Tribhuvan University

SUPERVISOR'S RECOMENDATION

I hereby recommend that this project prepared under my supervision by ANIL LAMA entitled **"ROUTINE MANAGEMENT SYSTEM"** in partial fulfillment of the requirements for the degree of B.Sc. in Computer Science and Information Technology be processed for the evaluation.

……

Sarbin Sayami

Assistant Professor

Deerwalk Institute of Technology

DWIT College

DWIT College

DEERWALK INSTITUTE OF TECHNOLOGY

Tribhuvan University

LETTER OF APPROVAL

This is to certify that this project prepared by ANIL LAMA entitled **"ROUTINE MANAGEMENT SYSTEM"** in partial fulfillment of the requirements for the degree of B.Sc. in Computer Science and Information Technology has been well studied. In our opinion it is satisfactory in the scope and quality as a project for the required degree.

…… Sarbin Sayami [Supervisor] Assistant Professor IOST, Tribhuan University	…… Hitesh Karki Chief Academic Officer DWIT College
…….. Jagdish Bhatta [External Examiner] IOST, Tribhuvan University	…….. Rituraj Lamsal [Internal Examiner] Lecturer IOST, Tribhuvan University

ACKNOWLEDGEMENT

I would like to express my deepest appreciation to all those who provided me the possibility to complete this report. I give a special gratitude to our Asst. Prof. Sarbin Sayami, whose contribution in stimulating suggestions and encouragement, helped me to coordinate my project.

Furthermore I would also like to acknowledge with much appreciation the crucial role of Deerwalk Institute of Technology, who gave the permission to use all required equipment and the necessary materials to complete the task. Last but not least, many thanks to my friends who invested their full effort in guiding me in achieving the goal. I have to appreciate the guidance given by other supervisor as well as the panels especially in my project, thanks to their comments and advices.

Tribhuvan University

Institute of Science and Technology

STUDENT'S DECLARATION

I hereby declare that I am the only author of this work and that no sources other than the listed here have been used in this work.

... ...

Anil Lama

Date: August, 2016

ABSTRACT

Routine Management System is a web based application as well as a mobile device application. The system is used to create and manipulate the class routine of an educational institution.

Routines of different educational institutions are created and maintained using the web application. Also the routine can be viewed from the web application. The primary purpose of developing a mobile application is to have an instant access of the routine from anywhere. The mobile application is primarily able to view the routine while the maintenance and creation is done through web application.

Sample routine of Deerwalk Institute of Technology was created and used and was uploaded to the server and the users were easily able to view the routine from both the systems.

Keywords: Routine Management System, Mobile Application, Web Application, Web Server, PHP.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATION

HTML:	Hyper Text Markup Language
CSS:	Cascading Style Sheet
OS:	Operating System
DBMS:	Database Management System
Md5:	Message-Digest algorithm 5
JSON:	JavaScript Object Notation
PHP:	Hypertext Preprocessor

CHAPTER 1: INTRODUCTION

1.1 Background

Every educational institution in Nepal have routines for the management of classes and time allocated to them. Currently these routines are manually created by a particular department in the institution. The parameters like teachers' time, number of periods, class time, etc. for creating routines are also manually kept in mind while creating these routines. Also the routines keep changing with time. Hence making their management even more difficult due to change in parameters value/state. If the above harsh process of creating and maintaining the routine could be done using an automated system then it would help the department of an institution to loosen the burden and increase the efficiency.

Due to the absence of such automated system for creating and maintaining the class routines, Routine Management System is required to be invented. Routine Management System is an automated management system comprising of both mobile and web applications used by educational institutions for effective management of creation and maintenance of routines.

1.2 Problem Statement

Most educational institutions manually create their routines by using a human resource to work on it. Some institutions create such management system for their personal requirement and use only while others do it manually. Unfortunately there is no such generic system at all for creating and maintaining class routines which can be used by all the educational institutions.

Routine Management System will help educational institutions to create and maintain the routines in a more effective and efficient way.

1.3 Objectives

1. To develop a web based system that allows users to create and maintain the class routines.
2. To develop an android system to view the created class routines.

1.4 Scope

Routine Management System benefits all the educational institutions that currently uses manual creation and maintenance of the class routines. Users can easily create and maintain class routines using the system's simple user interface. It also resolves the issue of manual staffing process into a controlled and closely monitored system. It can be used by every educational institutions interested to create class routines using a computerized system.

1.5 Limitation

1. The system cannot be used for making routines of activities other than class routine.
2. The system cannot automatically create routines based on given parameters.

1.6 Outline of Document

The remaining report is organized as follows.

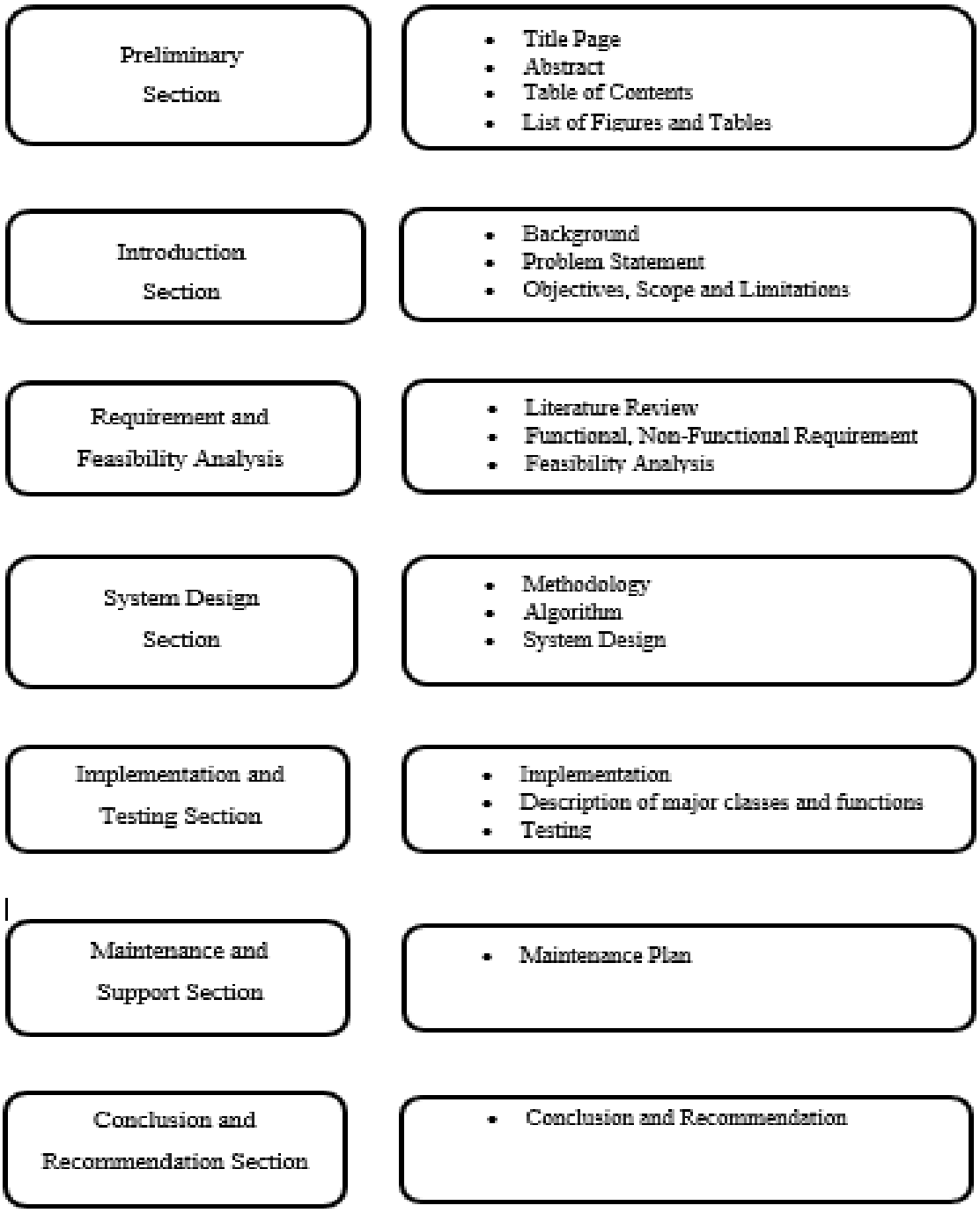

Figure 1 - Project Block Diagram

CHAPTER 2: REQUIREMENT ANALYSIS AND FEASIBILITY ANALYSIS

2.1 Literature Review

2.1.1 Comparison between Native and Hybrid App Development

According to a comparison done between Native and Hybrid / PhoneGap App Development by Bernard Kohan and Joseph Montanez, there are benefits and drawbacks to using either technology. Native app development uses the native programming languages of the devices to build the app. For iPhone, the native programming language is Objective C and the new Swift. For Android, the native programming language is Java. Hybrid apps use web technologies: HTML5, CSS and JavaScript, then put inside a native container such as Adobe PhoneGap. These native containers run the web application code and package it into an app. According to his experiment, there is more flexibility in designing the interfaces using hybrid app development technology. However, it is tedious and time consuming. Also, it is more cost effective to build mobile apps using hybrid app development technology stack as it has diverse sets of libraries providing the tools required to reduce the development time as well as saves money by not having to build the app using native programming language of each platform as it build the app once and submits it to all the platform using PhoneGap technology. Hybrid development also provide high maintainability. On the contrary, native app provide better performance, responsive and fluid experience and also provides better security environment. It also provides more support and lots of resources to develop mobile apps.

In addition to that, if the requirement is to create the best user experience, native development is a better choice. But for rapid development, budgetary limitation and if the app is simple, hybrid / PhoneGap is preferred.

(Bernard Kohan and Joseph Montanez, 2015)

According to the paper, Native versus Cross-platform frameworks for mobile application development by Rosario Madaudo and Patrizia Scandurra, the application development approach developers have to choose really depends on the application requirements itself. It depends on if the application requires capabilities native to the operating system, require security features or support capabilities. However, it is concluded that both approaches have advantages and disadvantages shortly summarized as Table 1 below.

Table 1 - Comparison of native and cross-platform development

(Source: Rosario Madaudo and Patrizia Scandurra, 2013)

Features	**Native**	**Cross-Platform**
UI User Experience	high	low
Performance	high	low
Device-specific features	high	low
Distribution via app-store	high	low
Multiple platforms deployment cost	high	low
Developers support	high	low
Security	high	low
Timely access to new OS innovations	high	low
Code reusability	low	high
Design challenges	low	high
Availability of programming expertise	low	high

According to the chosen criteria, native and cross-platform development approaches are complementary. These two approaches can be combined showing how the advantages of one can be exploited to cover or weaken the disadvantages of the other. In order to combine in a tight way preciseness and efficiency of native apps with flexibility and automation of cross-platform apps, conventional software architectural design patterns may be adopted and revised to adopt a hybrid development approach.

(Rosario Madaudo and Patrizia Scandurra, 2013)

According to a comparative study done by Paulo R. M. de Andrade, Adriano B. Albuquerque, the development of native applications require a high level of specialized knowledge in programming. A multiplatform architecture would be a solution to make this difficult task into something much more affordable, and with the possibility of the development using Web methods or hybrid application development. The development using HTML5, CSS3 and JavaScript allows a single application for smart phones to work on multiple operating systems using the same markup language of websites and requires a minimal level of investment in technical knowledge and time. The frameworks minimize the need for specialized programming language and increases the power of use of native application APIs. Native applications can provide a good user experience, but lack of money or expertise to develop natively can cause hindrance. A hybrid approach offers a simple solution for developing applications for smart phones and tablets. The code is written once and is then deployed to different operational systems will help companies to quickly launch their mobile applications and reduce maintenance costs. Hybrid structures are suitable options for the real benefits in the use of applications for business or for education.

(Paulo R. M. de Andrade, Adriano B. Albuquerque, 2015)

2.1.2 Upgrading android applications

From the experiment done by Luyi Xing, Xiaorui Pan, Rui Wang, Kan Yuan and XiaoFeng Wang, it can be deduced that when an android devices are upgraded, replacing and adding tens of thousands of files on a live system in the presence of a large amount of user data and existing apps, the apps installed can be crashed or vulnerable. To ensure that this process goes smoothly without endangering such user assets, the Android update mechanism involves complicated program logic and inevitably becomes error-prone. Their research reveals Pileup, a new type of privilege escalation vulnerabilities within the updating logic. Exploiting Pileup flaws, a malicious app can use what it declares on a low-version system to gain system capabilities on the new OS after an upgrade, involving gaining system and signature level permissions, substituting system apps, contaminating browser data and blocking the installation of new system apps. A large-scale measurement study was performed to confirm the presence of such flaws in all Android versions, official or customized. To mitigate the threat they pose, SecUP,

a new service that detects Pileup vulnerabilities from released system code, was developed which automatically gathers attack opportunities and leverages such information to support a scanner app running on the user's device to identify the malicious code attempting to exploit Pileup flaws.

(Luyi Xing, Xiaorui Pan, Rui Wang, Kan Yuan and XiaoFeng Wang, 2014)

2.2 Requirement Analysis

The requirement analysis for this project is broken down into functional and non-functional requirements and each is discussed below.

2.2.1 Functional Requirements

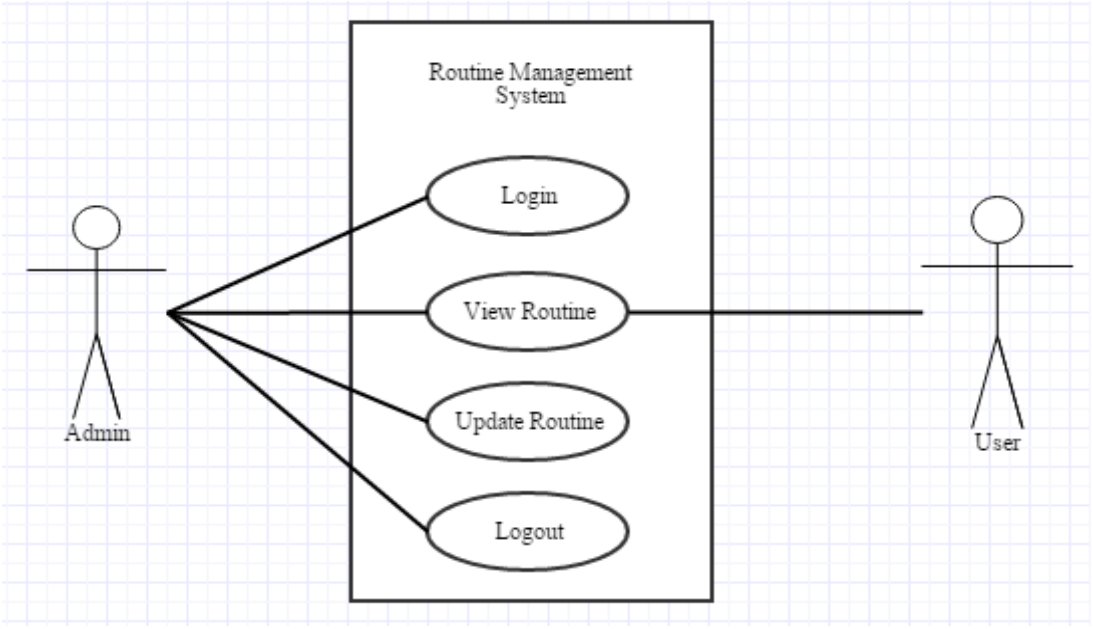

Figure 2 - Use Case Diagram of Routine Management System

2.2.2 Non-Functional Requirements

The non-functional requirements are:

1. Secure management of users is done by the use of session.
2. Web interface is developed to view and edit the routine while mobile application is developed to view the routine.
3. Data of routines are saved in fast and reliable database for fast data exchange.

2.3 Feasibility Analysis

2.3.1 Technical Feasibility

Routine Management System is a web interface and a mobile application that is developed in PHP Framework and Java (Android) respectively. It uses PHP pages, JavaScript for front end and backend of the web application while it completely uses Java (Android) for both backend and front end for the mobile application.

A server is required which is connected with an internet/intranet connection. Clients are able to use the system through personal computer or a mobile device.

It supports both Windows and Linux platform for its operation. All of the technology required by Routine Management System are available and can be accessed freely, hence it is determined technically feasible.

2.3.2 Operational Feasibility

Routine Management System is designed to easily manipulate class routine. It follows two tier architecture for the client request and server response operations. The system can be easily accessed via internet or intranet connection using a desktop or a mobile device. It can be used to view as well as manage routines of different class run in the institution. Moreover it can be accessed or updated from anywhere if the internet or intranet connection is available. Hence, Routine Management System was determined operationally feasible.

2.3.3 Schedule Feasibility

Schedule of Routine Management System is as follows:

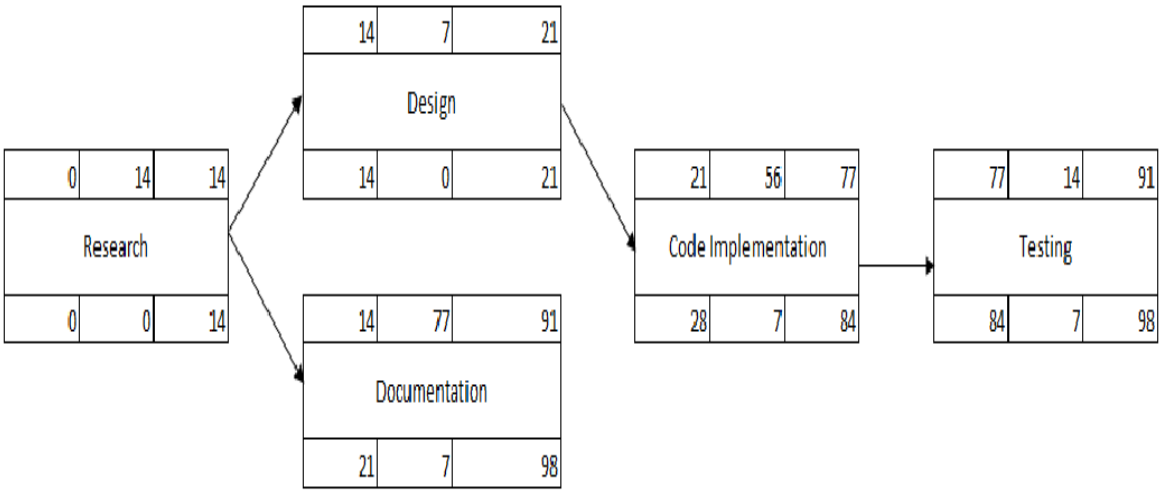

Figure 3 - Activity Network Diagram of Routine Management System

From the Figure 3 about the Activity Network Diagram, Routine Management System was completed in 91 days (7 weeks) which is within 15 weeks of the semester. Hence, Routine Management System was determined to be feasible in terms of schedule.

CHAPTER 3: SYSTEM DESIGN

3.1 Methodology

The methodology implemented while doing this project are discussed below.

3.1.1 Data Collection

Data for Routine Management System was collected by taking class routine of Deerwalk Institute of Technology. The administration was requested for getting the class routines data.

Deerwalk Institute of Technology

SENIOR YEAR_SEMESTER VII_CLASS OF 2016

WEEKLY CLASS SCHEDULE

DAY	SEC	8:10-9:00	9:0-9:50	10:10-11:00	11:00-11:50	11:50-12:40	12:40-13:30	13:30-14:20	14:20-15:10
MONDAY	A	Advanced Java Programming A	Advanced Java Programming A	Lunch	Database Administration A	Database Administration A	Internet Technology A		
	B	Advanced Database Management System B	Advanced Database Management System B	Lunch	Internet Technology B	Internet Technology B	Advanced Java Programming B		
TUESDAY	A	Internet Technology A	Internet Technology A	Lunch	Advanced Java Programming A	Advanced Java Programming A	Database Administration A	Advanced Database Management System A	
	B	Advanced Database Management System B	Advanced Database Management System B	Lunch	Database Administration B	Database Administration B	Advanced Java Programming B		
WEDNESDAY	A	Advanced Java Programming A	Advanced Java Programming A	Lunch	Database Administration A	Database Administration A	Advanced Database Management System A		
	B	Advanced Database Management System B	Advanced Database Management System B	Lunch	Internet Technology A	Internet Technology A	Database Administration B		
THURSDAY	A	Advanced Database Management System A	Advanced Database Management System A	Lunch	Internet Technology B	Internet Technology B	Database Administration A		
	B	Advanced Java Programming B	Advanced Java Programming B	Lunch	Database Administration B	Database Administration B	Self Study		
FRIDAY	A	Advanced Database Management System A	Advanced Database Management System A	Lunch	DeerTalk/ DeerExpress		Internet Technology A		
	B	Internet Technology B	Internet Technology B	Lunch			Advanced Java Programming B	Advanced Java Programming B	Database Administration B

Figure 4 - Sample Data Set of Routine Management System

Sample routine data set of seventh semester is shown in the Figure 4.

3.1.2 Data Preprocessing

The routine data collected from Deerwalk Institute of Technology were edited as per the systems requirement. The unwanted and irregular parts of the routine data were removed.

3.1.3 Data Storage

The routine data are stored in the files of JSON format. The routine data are also retrieved from the same files. The edit to the routine data after the update is also done in the same files.

3.1.4 Data Validation

The routine data were validated by the testing them. Testing was performed properly using test cases which are presented in section 4.3.

3.2 Algorithm

3.2.1 Algorithm for Operations

Algorithms for the operations in Routine Management System are further described in sections below.

3.2.1.1 Routine Manager

Step 1: Start.

Step 2: Routine Manager logs in to the system.

Step 3: Routine Manager chooses the batch whose routine needs to be changed.

Step 4: Routine Manager updates the routine as per the requirement.

Step 5: End.

3.2.1.2 Routine Viewer

Step 1: Start.

Step 2: Routine Viewer chooses the batch whose routine is to be viewed.

Step 3: In the mobile application if there is no routine data, the system first downloads the JSON routine data.

Step 4: The JSON routine data is then parsed to get data in simpler form.

Step 5: Acquired simple data are then stored in the database.

Step 6: Routine data from database are then displayed to the user.

Step 7: If the routine is outdated then the user updates the routine explicitly by pressing update option.

Step 8: End

3.3 System Design

3.3.1 Class Diagram

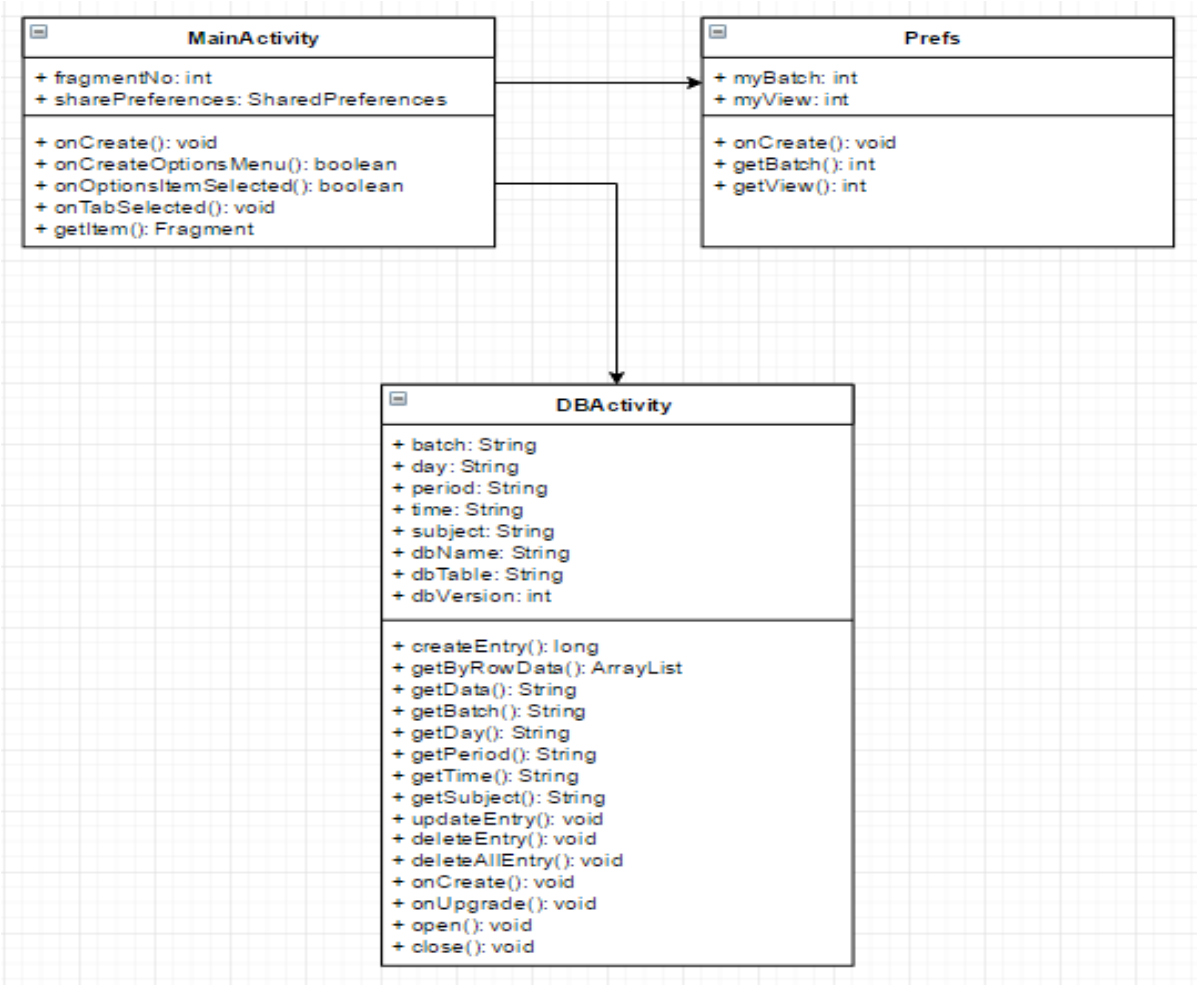

Figure 5 - Class Diagram of Mobile Application

Figure 5 explain the classes used in the mobile application of Routine Management System. There are three classes used in total, MainActivity, Prefs and DBActivity. MainActivity is the main class in the application. All the initializing processes are done in MainActivity and processes related to preferences of users are done in Prefs while all the database activities are organized by DBActivity.

3.3.2 State Diagram

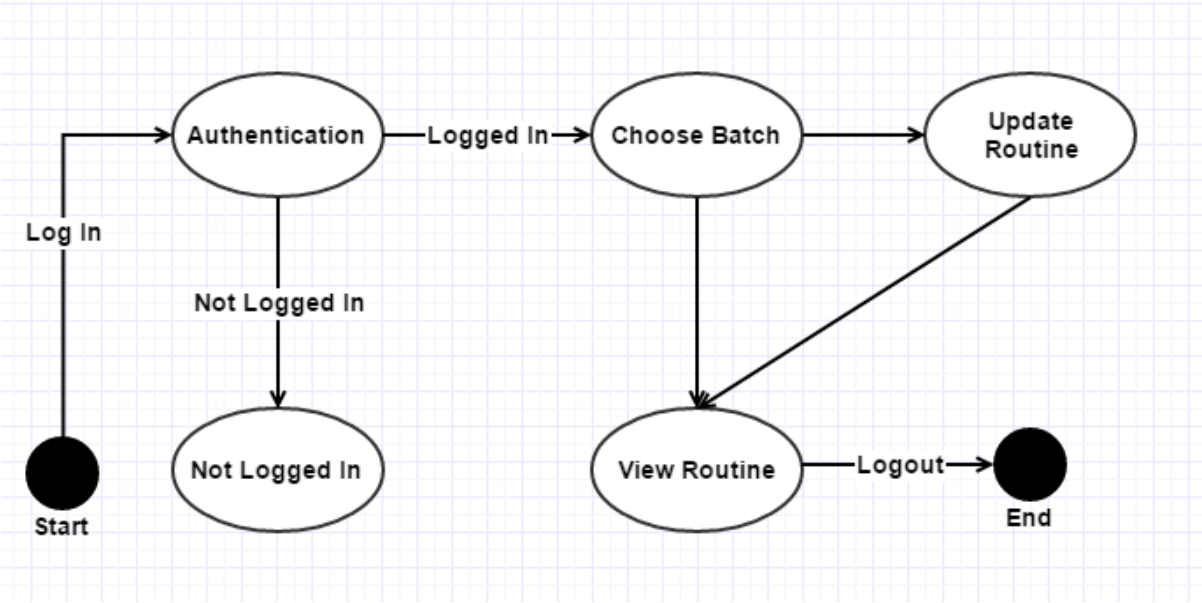

Figure 6 - State Diagram of Web Application

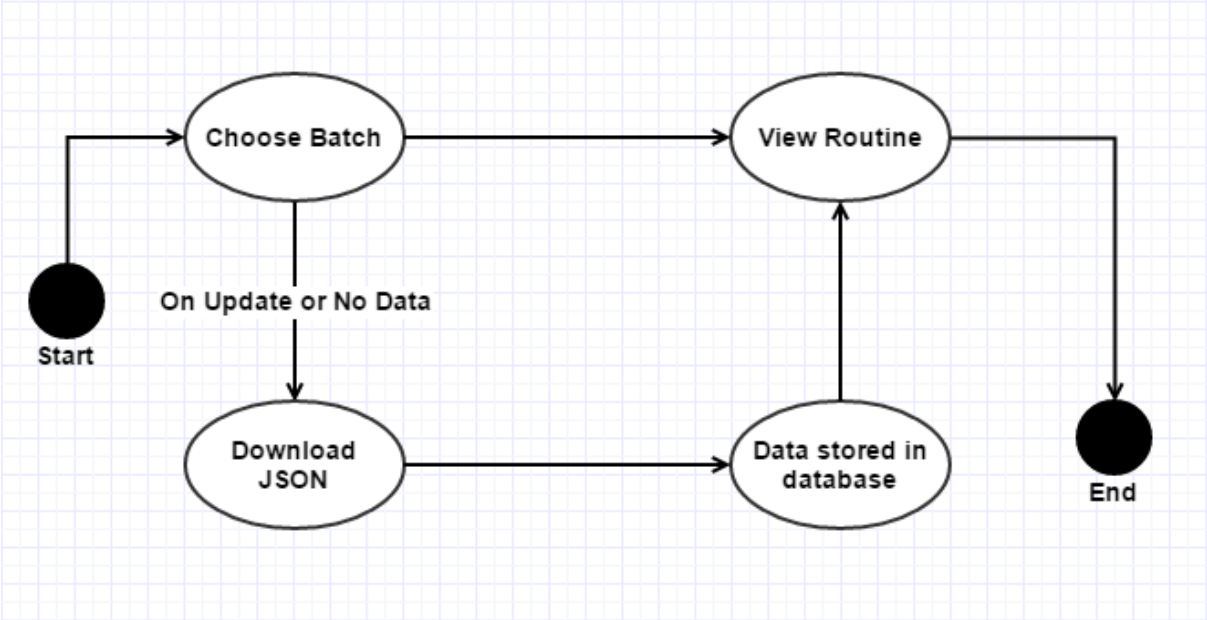

Figure 7 - State Diagram of Mobile Application

Figure 6 presents the state diagram of the web application of Routine Management System. First the user logs in to the system. Here the system authenticates the user. If the user is a valid user, the user enters the system where it gets to choose the batch. Batch is chosen to either view the routine or to update the routine of a particular batch. And finally after all the operations, user logs out of the system.

Similarly, Figure 7 presents the state diagram of the mobile application of Routine Management System. Here the user first opens the mobile application. If the system is opened for the first time, the user is prompted to choose the batch whose routine is to be displayed as default. Once chosen, the system then downloads the current routine data for all the batches and store them into the database. Once the data is kept in the database, the system displays the routine of the batch which was chosen firstly.

3.3.2 Sequence Diagram

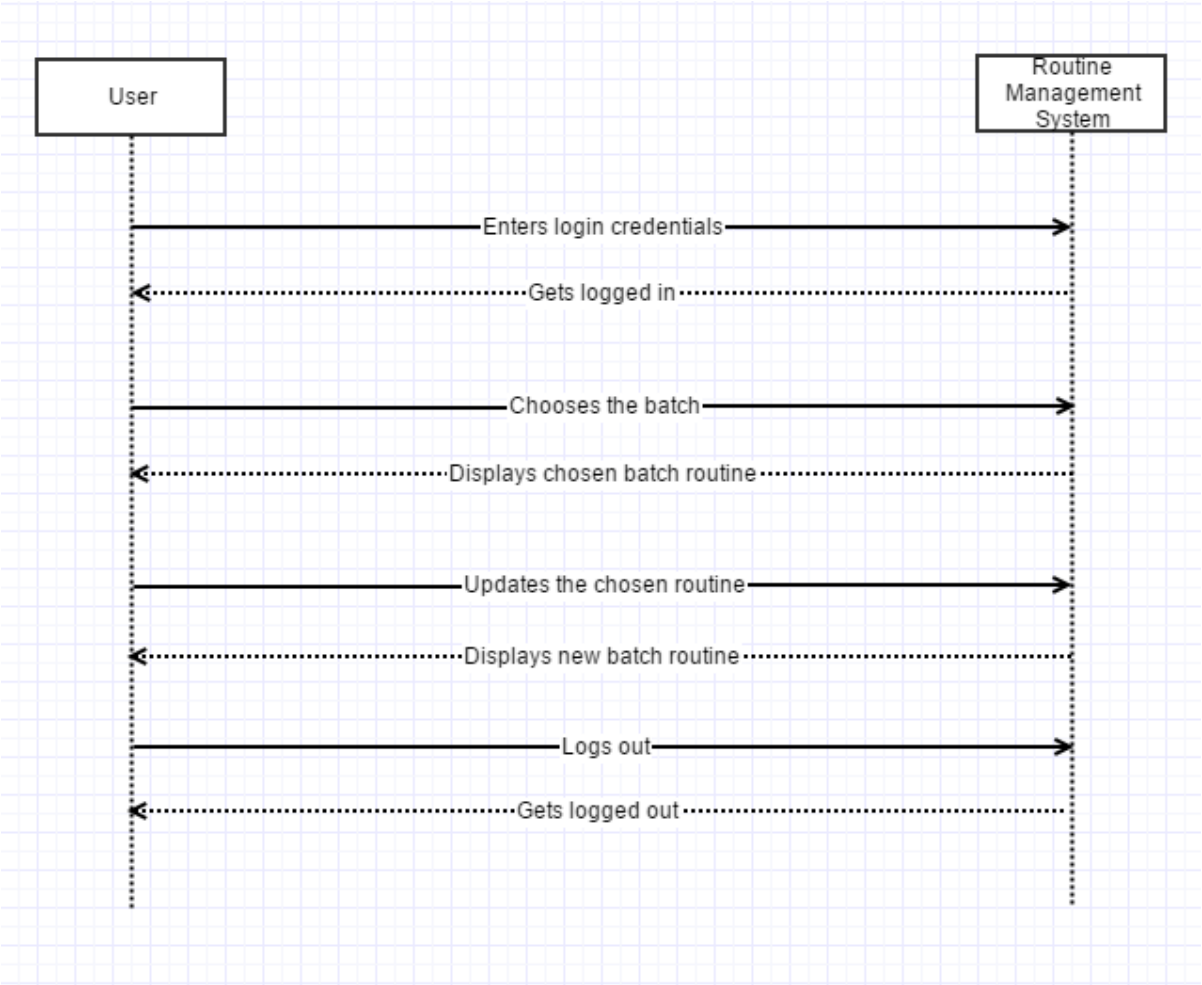

Figure 8 - Sequence Diagram of Web Application

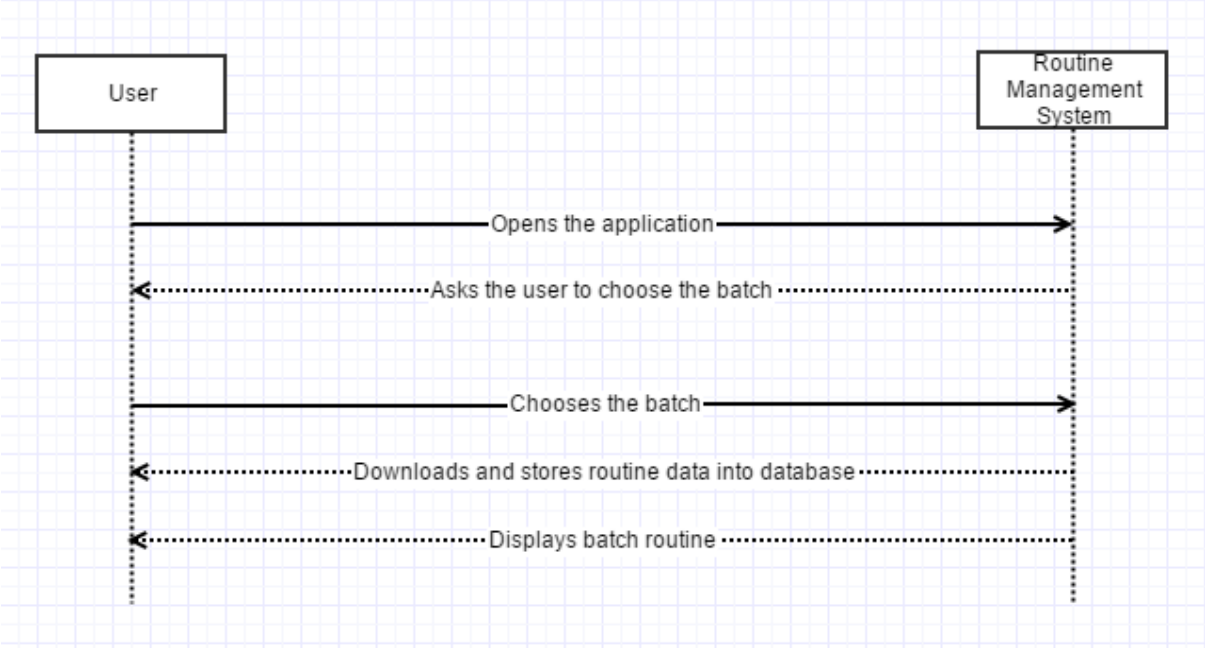

Figure 9 - Sequence Diagram of Mobile Application

Figure 8 explains the sequence of the web application of Routine Management System. Initially the user browses the system using a web browser via internet connection. User then logs in to the system. The user then chooses the batch and views the routine. If the routine is outdated, the user updates the routine. User then logs out of the system.

Similarly, Figure 9 explains the sequence of the mobile application of Routine Management System. User first opens the application. System then asks the user to choose the default batch for the routine. User chooses the batch and then the system downloads the data and stores them into the database. The system finally displays routine of the chosen batch.

CHAPTER 4: IMPLEMENTATION AND TESTING

4.1 Implementation

Routine Management System is a combination of web application as well as a mobile application. Hence the web application is accessed using a browser via internet connection. User can view the routine provided the web interface of the system. The user can also update the routine as per the requirement. This section describes about the technologies used in Routine Management System.

4.1.1 Tools Used

CASE Tools:

1. Android Studio

Android Studio was used to develop the mobile application of Routine Management System. It was used to develop both front end and backend side of the mobile application.

2. PhpStorm

PhpStorm was used to develop the web application of Routine Management System. It was used to develop both front end and back end of the web application.

Client Side:

1. HTML

HTML was used for the front end side of the web application of Routine Management System. It was used in designing the webpages of the system.

2. CSS

CSS was used to style the webpages of the web application of Routine Management System.

3. JavaScript

JavaScript was used for client side validation of the web application of Routine Management System.

Server Side:

1. PHP

PHP was used to develop the back end side of the web application of Routine Management System.

2. Java (Android)

Java was used in development of both back end and front end of the mobile application of Routine Management System.

DBMS

1. SQLite

SQLite was used as the database management system for the mobile application of Routine Management System.

2. Md5

Md5encryption is used for encrypting the data as per the requirement. For the web application, session tracking is implemented to track and validate the user throughout the use of Routine Management System. Data exchange and synchronization between the web and mobile system is done via JSON file.

4.2 Description of major classes and functions

The major classes in the web application are listed and explained below.

4.2.1 Login

This is the class used to login the system.

Input: It takes the username and password of a user.

Process: It validates whether the user is valid or not.

Output: It provides access to the system if the user is valid, else not.

4.2.2 Common

This is the class used in parsing the JSON file to get simpler data.

Input: It takes the JSON filename with batch parameter.

Process: It parses the JSON file to return simpler data of the batch provided.

Output: It outputs parsed routine data from the JSON file.

The major classes in the mobile application are listed and explained below.

4.2.3 MainActivity

This is the landing class which is first run when the system is opened.

Input: It takes the batch whose routine is to be shown.

Process: It determines the batch and processes to give routine.

Output: It provides routine according to the batch.

4.2.4 Prefs

This is the class used in storing the preferences of the user.

Input: It takes the batch to display as default.

Process: It reads and stores the value given in the input.

Output: It displays stored value in the preferences.

4.2.5 DBActivity

This is the class used to handle all the database activities of the system.

Input: It is triggered by a call to its function.

Process: It reads and process the request according to the function name and parameters.

Output: It returns the result after the query is performed.

4.3 Testing

The system was provided a sample routine of Deerwalk Institute of Technology through its web application. The routine was then modified and uploaded to the server. Then users were easily able to access the routine via both the web as well as mobile application. It was easily viewable for both the user systems.

4.3.1 Unit Testing

Table 2 describes the test cases of Routine Management System.

Table 2 - Test Cases of Routine Management System

Test no.	Unit	Test	Expected Result	Test Outcome	Evidence
1	Log In	Log in to the system	User is logged in to the system	Successful	Test 1.1
2	Update	Update the routine	Routine is updated and the user views the updated routine	Successful	Test 1.2
3	View	View the routine	Routine of the chosen batch is displayed.	Successful	Test 1.3

Following are the test cases for the web application of Routine Management System.

Test 1.1:

Precondition: The application is accessed using web browser.

Assumption: User has opened the application and is on login page.

Input: User credentials are entered and login button is pressed.

Result: User is logged in to the system.

Test Case 1.2:

Precondition: The application is accessed using web browser.

Assumption: User is logged in and viewing the routine.

Input: Batch whose routine needs to be updated is chosen and update button is pressed.

Result: Routine is updated and the user views the updated routine.

Following is the test case for the mobile application of Routine Management System.

Test Case 1.3:

Precondition: The application is installed in the user's android phone.

Assumption: User opens the application.

Input: Batch to be stored as default is chosen and OK is pressed.

Result: Routine of the chosen batch is displayed.

CHAPTER 5: MAINTENANCE AND SUPPORT PLAN

Once the software is deployed and delivered, the maintenance phase starts. The system requires maintenance as there may exist some existing errors. As the system is tested with very users only, there is a high change for the system to crash if large number of users try to access the web system at once since the system is not optimized for such situations.

Also once the system is deployed, feedback from users is expected and maintenance of the system will be done accordingly.

5.1 Maintenance Plan

Routine Management System will implement corrective maintenance, adaptive maintenance, perfective maintenance and preventive maintenance.

5.1.1 Corrective Maintenance

In case of errors being occurred in the hosted application, I will get the copy of the application from the server and maintain it to correct the errors occurring. Then I'll replace the server's application with the corrected application. The errors will be resolved as soon as I get notified.

5.1.2 Adaptive Maintenance

Adaptive maintenance will be performed when the system needs to adapt to the changing demand and environment. This will be done to fulfill the changing requirements of the users.

5.1.3 Perfective Maintenance

Perfective maintenance will be maintained in the condition when the user demands for some features to be added. This will be done to enhance the system services.

5.1.4 Preventive Maintenance

Preventive maintenance will be implemented so that hackers or illegal users will not be able to access the system and its data.

CHAPTER 6: CONCLUSION AND RECOMMENDATION

6.1 Conclusion

Routine Management System was developed successfully. The system was developed with both web and mobile application. The system was successfully completed.

6.2 Recommendation

The system is only developed for creating and updating routines. Hence it has a good scope for upgrading the system to automatic routine generation system as well.

APPENDIX

Login Page

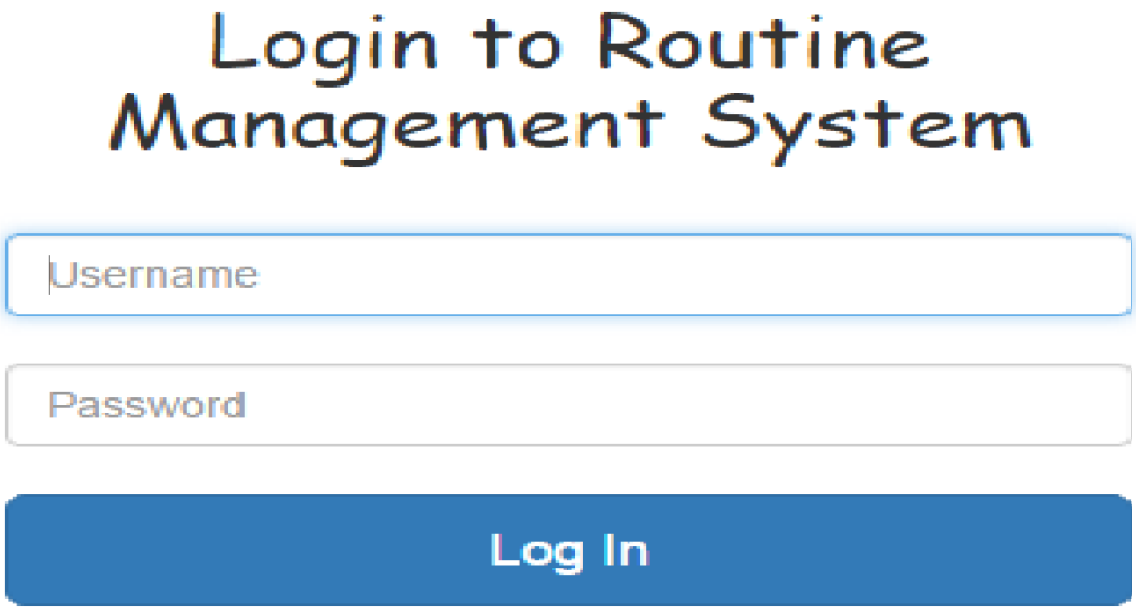

Figure - Login Page of Web Application

The above figure shows the login page of the web application of Routine Management System. In this page, the system takes two inputs; username and password where user enters the username and password of the user and clicks the Log In button.

Routine View Page

Figure - View Page of Web Application

The above Figure shows the view page of the web application of Routine Management System. In this page, the user can view the routine by clicking Show button, update the routine by clicking Update button and logout from the system by clicking Logout button.

Update Page

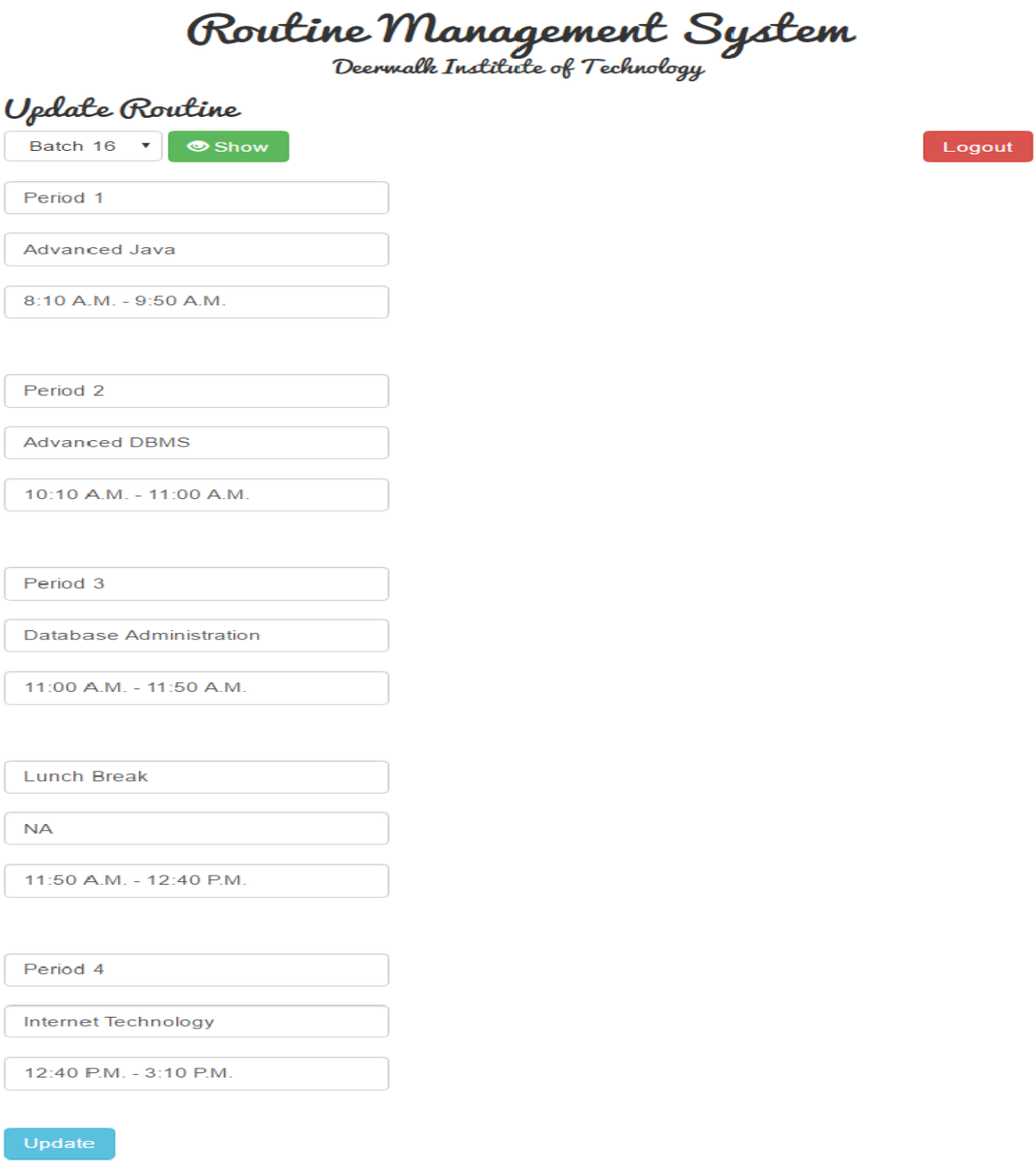

Figure - Update Page of Web Application

Figure above shows the update page of the web application where the user can update the routine by changing routine data and clicking the Update button.

View Page of Mobile Application

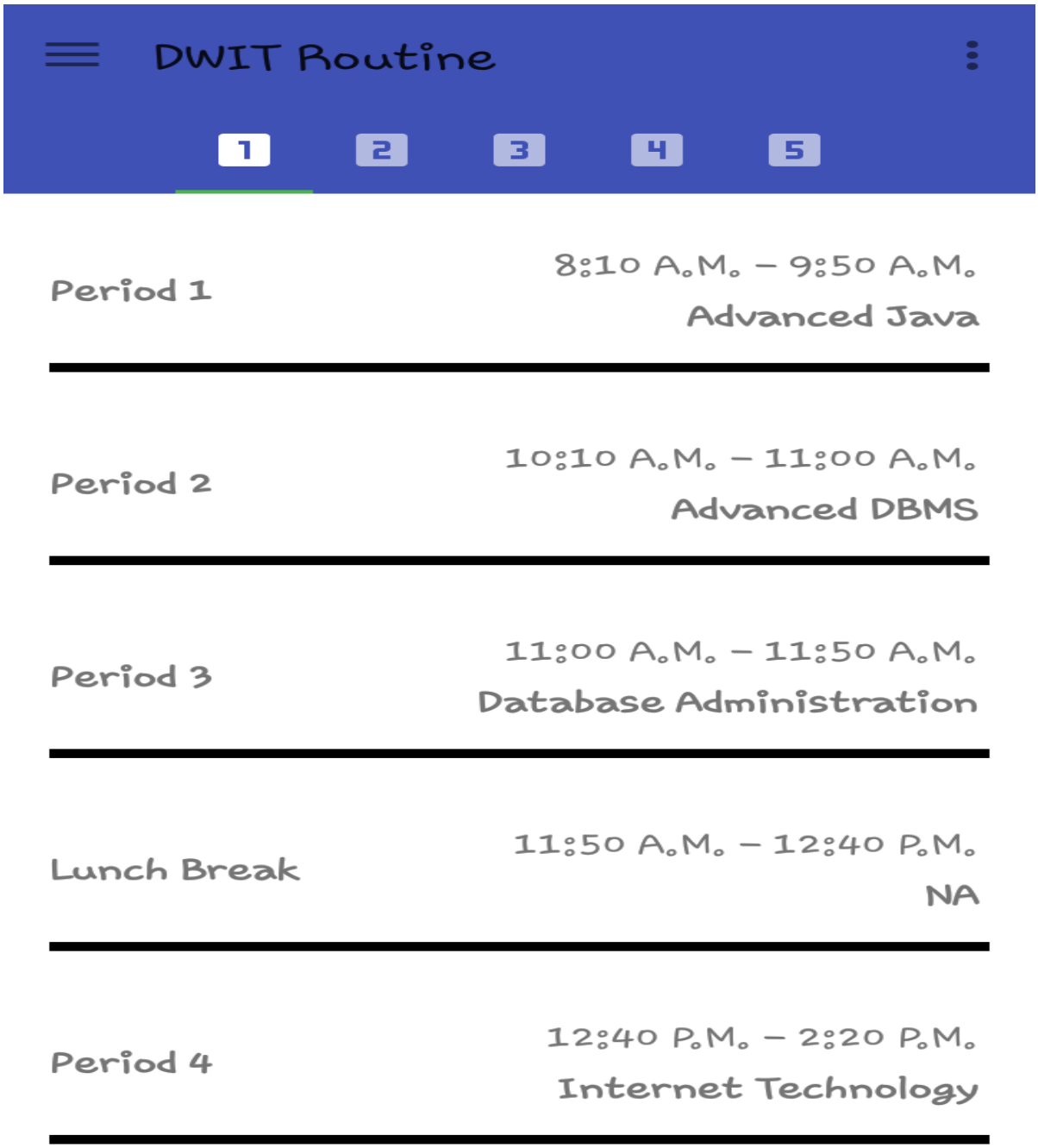

Figure -View Page of Mobile Application

Figure above shows the routine view page of the mobile application of Routine Management System.

REFERENCES

Bernard Kohan and Joseph Montanez, Native vs Hybrid / PhoneGap App Development Comparison, 2015

Luyi Xing, Xiaorui Pan, Rui Wang, Kan Yuan and XiaoFeng Wang, Upgrading Your Android, Elevating My Malware: Privilege Escalation Through Mobile OS Updating, 2014

Paulo R. M. de Andrade, Adriano B. Albuquerque, CROSS PLATFORM APP A COMPARATIVE STUDY, 2015

Rosario Madaudo and Patrizia Scandurra, Native versus Cross-platform frameworks for mobile application development, 2013