AN ANDROID APPLICATION FOR DIFFERENT LEVELS OF COLLEGE MANAGEMENT

A MINOR PROJECT REPORT

Submitted by

VIJAY KRISHNA V [Reg No: RA1511003020329]
CHIRAG G SAMTANI [Reg No: RA1511003020375]
SRUSHTI BOMPELLI [Reg No: RA1511003020344]
CHAITANYA BACHHAV [Reg No: RA1511003020383]

Under the guidance of
Mr M. Prabu
(Asst. Professor , Department of Computer Science & Engineering)

BACHELOR OF TECHNOLOGY

in

COMPUTER SCIENCE & ENGINEERING

of

FACULTY OF ENGINEERING AND TECHNOLOGY

SRM UNIVERSITY
(Under section 3 of UGC Act 1956)

Ramapuram,Chennai,600089

OCT 2017

SRM UNIVERSITY

(Under Section 3 of UGC Act, 1956)

BONAFIDE CERTIFICATE

Certified that this project report titled "**AN ANDROID APPLICATION FOR DIFFERENT LEVELS OF COLLEGE MANAGEMENT**" is the bonafide work of " **VIJAY KRISHNA V [Reg No: RA1511003020329], CHIRAG G SAMTANI [Reg No: RA1511003020375], SRUSHTI BOMPELLI [Reg No: RA1511003020344], CHAITANYA BACHHAV [Reg No: RA1511003020383]**", who carried out the project work under my supervision. Certified further, that to the best of my knowledge the work reported herein does not form any other project report or dissertation on the basis of which a degree or award was conferred on an earlier occasion on this or any other candidate.

SIGNATURE	**SIGNATURE**
Mr M. Prabu **GUIDE** Asst. Professor Dept. of Computer Science & Engineering	Dr. JAGADEESAN,M.Tech.,Ph.D **HEAD OF THE DEPARTMENT** Dept. of Computer Science & Engineering
Signature of the Internal Examiner	Signature of the External Examiner

ABSTRACT

Technical and scientific development has become one of the most essential changes in every aspect of the modern day to make our work efficient and easier. The proposed Android application has been designed to efficiently process the management system of an institution at their different levels of hierarchy, to reduce the physical work and confusion. Black board is an application specifically for the faculty members and the head of departments. The application will show required content based on who is signing into the app. For instance, if the dean logs into the app, he/she can view the details of all the heads and their staff according to their departments, a HOD will be able to see the details and availability of the faculty members of their own department. The application will be able to display most of the academic information like the timetables, availability of the faculty, location of the classrooms etc. The database of the application can be modified and updated through a web portal by a specific admin. Many future enhancements can be made to the mobile application based on the institutions requirements.

ACKNOWLEDGEMENTS

I would like to express my deepest gratitude to my guide, Mr. M. Prabu his valuable guidance, consistent encouragement, personal caring, timely help and providing me with an excellent atmosphere for doing research. All through the work, in spite of his busy schedule, he has extended cheerful and cordial support to me for completing this project.

Author

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

ABBREVIATIONS

LS	Least Square
PSO	Particle Swarm Optimization
SI	Structural Identification

LIST OF SYMBOLS

α, β	Damping constants
θ	Angle of twist, rad
ω	Angular velocity, rad/s
b	Width of the beam, m
h	Height of the beam, m
$\{f(t)\}$	force vector
$[K^e]$	Element stiffness matrix
$[M^e]$	Element mass matrix
$\{q(t)\}$	Displacement vector
$\{\dot{q}(t)\}$	Velocity vector
$\{\ddot{q}(t)\}$	Acceleration vector

CHAPTER 1

INTRODUCTION

1.1 Overview

The project is designed for the effective management of a college or an institution according to their different levels of hierarchy present. In today's world it is very important to work efficiently to avoid any confusion and for productive results. In the administration of a college or an institution there exists various levels of hierarchy, the responsibilities and functions differ at each level, in the existing system most of the work done is paper work and it is manually carried out, this may lead to confusion and increase of work load at each level. With increasing technology and tools in each and every aspect of the world, it has become much easier and efficient to carry out any task. Our project intends to give technical development to this particular field of college management. The proposed android application will be able to store and retrieve the academic data and can be accessed accordingly by the different levels of the management structure depending upon their necessities and functionalities.

The project consists of an android application with a web portal and a common database connecting them. The android application will have separate logins for the dean, HOD, class in charge, faculty etc. once they login they will able to access the information required by them. For instance, if the dean logs into the app, he/she can view the details of all the heads and their staff according to their departments, a HOD will be able to see the details and availability of the faculty members of their own department and other information about their department. The main objective of the application is that all the academic information should be available at one place in an organized manner which can be easily accessible by just a click. The application helps in making tedious tasks like searching for classrooms or staff easier. The application also provides the facility to make calls to the required person from the app. Features like checking the availability of free staff helps in the effective management of time without free periods being wasted. The data in the application is stored in the database which will be updated from time to time. The application will be very user friendly to avoid any complications and it can be used by all the management members.

A web portal will be designed to update the database from time to time every year or every semester depending upon the college. It is easier to update the information than through the application itself. The web portal will consist of a login page through which the admin can login and update the information; the admin can be any faculty member with a secure login ID and password. The login details of the admin will be checked carefully and secure access will be given. Once the admin logs into the web portal he will be given access to update the all the information like changed time tables, details of the new students, change of classes etc. the website can be designed according to the convenience of the college, it will be user friendly with advanced features for the convenience of the user. The android application and the web portal are connected using a common database. The database contains all the information which is retrieved by the app and updated using the web portal. The data is stored in the database using a private cloud the information can only be accessed through secure login.

The scope and implementation of the project can be increased and developed with changing requirements. The project is designed and implemented using the latest and updated technologies for more efficiency and ease of access.

1.2 Problem Statement

Organized data is easier to work with than manual paper work, The problem in the existing system is that most of the work done by the management is paper work, there is no scope of organized data, each person have to maintain different files to access all the academic information. The main drawbacks of the existing college management system are:

- Manual work for all the teachers regarding attendance, updating and changing the attendance is difficult.
- No way of knowing which teacher is free during which period.
- No provision for rectification of mistakes.

- Dean doesn't have access to everyone's contacts in one place.
- Locating classes is not an easy task.
- Contacting a higher official is difficult.

These drawbacks will be addressed by the project, the android application will facilitate the availability of all the information together, attendance can be updated through the app by the faculty whenever required; there will always be a way to rectify mistakes. The project will be implemented to address all the drawbacks of the existing system and improvise the effectiveness of the system.

1.3 Objective

The main objective of the proposed system is to introduce and implement technical advancement in the existing system of college management. The proposed the android application will be able to make the work easier by providing ease of access to the data required. Some of the main objectives of the project will be avoiding manual work for attendance, separate logins for different hierarchy of management, easier access to academic information, all information related to the particular person available in one application, effective management of time tables and free time, each level of management can access information relevant only to them (and not all information) through a secure login which keeps the data secured and avoids any misuse of data, increases overall efficiency of management. The web portal is used update the data in the database, one of the main objective of the web portal is to be user friendly to facilitate the admin to update the details with ease and the updated information should be secure which will be provide through secure login of the admin.

1.4 Organization of the report

There are five chapters in the report of the project android application for different levels of college management. The domain of the project is a common database connecting the android application and web portal. The report gives a comprehensible overview of the design and implementation details of the project, it helps understand the various technologies used in the project and the effective implementation of these technologies.

Chapter 1

This chapter gives the basic introduction of the project, it gives the basic overview of the entire project and gives the details about the problem statement and the main objective of the project.

Chapter 2

This chapter includes the literature survey which is a survey done on various base papers, it mainly gives the problems and conclusions for the domain of the project. The chapter also gives the description about all the features of the existing system and its drawbacks and the proposed system and its advantages.

Chapter 3

This chapter gives the requirements of the project in detail. It includes the basic purpose of the project, its scope, user requirements of the project. Software and Hardware specifications required by the project are given in detail in this chapter.

Chapter 4

This chapter the complete information about the system design and the various modules present in the project. The chapter gives the system architecture and its description , it gives the detailed description about the data flow diagram. The module implementation is given in detail with the screen shots from the projects and the diagrams required. Data flow diagram and the process description for each module is separately given for better understanding.

Chapter 5

This chapter deals with entire implementation of the system. The information about the platform of the project is given, the sample coding and the screen shots of the working model with proper description is given this chapter. The project will be done and the summary will be given.

Chapter 6

The conclusion of the project will be given in this chapter. The project also gives an insight about the future work which can be implemented through the project.

CHAPTER 2

SPECIFICATIONS

2.1 Introduction

In this chapter, the specifications of this project are going to be discussed in detail. Topics like purpose, project scope and much more is also included.

2.1.1 Purpose

Teachers should do exactly what the name suggests, teach. The current education system does not allow them to do just that. Unfortunately, as the teachers have no other choice, they follow the system. Doing all the manual work of attendance and calculation of marks and all that itself takes up half their time. Valuable time that could have been spent on imparting knowledge to the students. This is where comes the purpose of our project. To decrease manual work for the teachers in colleges so that they can concentrate on imparting their knowledge to students.

2.1.2 Project Scope

The scope of this project is huge. We plan to first implement this only in our college but as soon as that happens, we plan to approach various colleges where this problem of manual work persists and implement it there. After colleges, we can move on to schools and other institutions. The scope is endless.

2.2 Overall Description

In overall description we're going to discuss about how the project is implemented. Description of what was done, the features, the user characteristics, operating environment, implementation constraints and much more.

2.2.1 Product Features

The various feature in our application and web interface will include the following:

- Separate logins for different hierarchy of management.
- Easier access to academic information.
- All information related to the particular person available in one application.
- Effective management of timetables.
- Each level of management can access information relevant only to them (and not all information).
- Increases overall efficiency of management.
- Locations of classes in the campus can be found in the application.
- Academic information of all the students can be found under the separate class teacher's login.
- The dean can access the contact information of every teacher on the campus.
- The dean can call any teacher he wishes to from the application itself.
- The HOD of each department can view the performance of students of his/ her department.
- Free periods of teachers can be seen through the application which will help in allocating them to classes where teachers are absent.

2.2.2 User classes and characteristics

The various user classes and their characteristics are given below:

Login

The login class takes the login information from the user and authenticates it. If the information matches with that of the database, it directs it to another page based on whether the user is a teacher, dean or a HOD of a department.

Menu

The menu is where the user can select what functionality he/she wants to use. The various menu items include:

- Class time tables
- Teacher time tables
- Available Staff
- Class Location

Class Time Tables

This is where the user can select the particular class he/she wants the time table to and can view the time table of the same in the application itself.

Available Staff

This is where the senior staff can view the free periods of the teachers so that he/she can assign the teacher another class in case there is a class without a teacher at that point.

Class Location

This is where the user can enter the class he/she wants to go to and the result will be the location of the class in the campus. We later plan to add a student login where this feature will also be available.

Call Staff

This is where the dean will be able to browse through all the teachers and HODs and will be able to call any one of them from the application itself.

Teacher Time Tables

This is where the HOD can select the teacher from his/her department and view the time table of the same in the application itself.

2.2.3 Operating environment

The Android SDK includes a virtual mobile device emulator that runs on your computer. The emulator lets you prototype, develop and test Android applications without using a physical device. The Android Emulator simulates a device and displays it on your development computer. It lets you prototype, develop, and test Android apps without using a hardware device. You can launch an app on the emulator when you run your project, or you can drag an APK file onto the emulator to install it. As with a hardware device, after you install an app on a virtual device, it remains until you uninstall or replace it. If needed, you can test how multiple apps, such as your own or system apps, work with each other. You interact with the emulator just as you would with a hardware device, but using your mouse and keyboard, and emulator buttons and controls. The emulator supports virtual hardware buttons and touchscreens, including two-finger operations, as well as directional pads (D-pads), trackballs, wheels, and various sensors. You can dynamically resize the emulator window as needed, zoom in and out, change the orientation, and even take a screenshot.

When your app is running on the emulator, it can use the services of the Android platform to invoke other apps, access the network, play audio and video, accept audio input, store and retrieve data, notify the user, and render graphical transitions and themes. The emulator has controls that let you easily send incoming phone calls and text messages, specify the location of the device, simulate fingerprint scans, specify network speed and status, and simulate battery properties. The emulator can simulate an SD card and internal data storage; you can drag a file, such as a graphics or data file, onto the emulator to store it.

2.2.4 Design and Implementation Constraints, Assumptions and dependencies

The one dependency that this project has is that the data has to be fed into the web portal every semester and this responsibility has to be taken up by one individual.

Material Design is a design language developed by Google. Material design is a comprehensive guide for visual, motion, and interaction design across platforms and devices. A single underlying system that allows for a unified experience across platforms and device sizes.

2.3 External Interface Requirements

In this part, we're going to discuss about the external interface requirements like the user interface, hardware and software interface:

2.3.1 User Interface

We've used xml for making the user interface of the android application. And HTML, CSS for the user interface of the web portal. This is a very important part of the project as without a proper UI, the user wouldn't be able to navigate through the application properly and in the end, that is of no use.

2.3.2 Hardware Interface

There is no physical database where all the data is stored, it's all stored in the cloud. The other physical component required will be the smartphone itself which every user will undoubtedly have.

2.3.3 Software Interface

The various components of the software interface are given below:

Login Interface

The login class takes the login information from the user and authenticates it. If the information matches with that of the database, it directs it to another page based on whether the user is a teacher, dean or a HOD of a department.

Menu Interface

The menu is where the user can select what functionality he/she wants to use. The various menu items include:

- Class time tables
- Teacher time tables
- Available Staff
- Class Location

Class Time Tables Interface

This is where the user can select the particular class he/she wants the time table to and can view the time table of the same in the application itself.

Available Staff Interface

This is where the senior staff can view the free periods of the teachers so that he/she can assign the teacher another class in case there is a class without a teacher at that point.

Class Location Interface

This is where the user can enter the class he/she wants to go to and the result will be the location of the class in the campus. We later plan to add a student login where this feature will also be available.

Call Staff Interface

This is where the dean will be able to browse through all the teachers and HODs and will be able to call any one of them from the application itself.

Teacher Time Tables Interface

This is where the HOD can select the teacher from his/her department and view the time table of the same in the application itself.

2.3.4 Communication Interface

There are three levels of communications happening at a given moment in the app's lifecycle. They are mentioned below:

- Web-portal to database
- Database to web-portal
- Application to database

Note that the web-portal and the application do not communicate at all. The data is fed through the web portal and is stored in the database which is then accessed by the application to display it to the users.

2.4 Other non-functional Requirements

In this section, we're going to discuss about the other non-functional requirements like performance and security requirements.

2.4.1 Performance Requirements

Other than a smartphone and an internet connection, nothing will be required to run this application with the highest performance.

2.4.2 Security Requirement

The app automatically encrypts the password of the user and hence the passwords are kept safe. In case the user forgets his/ her password, extra steps will be taken to make sure that the user is identified first. And hence, security is taken care of.

CHAPTER 3

SYSTEM IMPLEMENTATION

3.1 Overview of the Platform

3.1.1 Java

The Java language is a key pillar in Android, an open source mobile operating system. Although Android, built on the Linux kernel, is written largely in C, the Android SDK uses the Java language as the basis for Android applications.

Java is a general-purpose computer programming language that is concurrent, class-based, object-oriented, and specifically designed to have as few implementation dependencies as possible. It is intended to let application developers "write once, run anywhere" (WORA), meaning that compiled Java code can run on all platforms that support Java without the need for recompilation. Java applications are typically compiled to bytecode that can run on any Java virtual machine (JVM) regardless of computer architecture.

3.1.2 Kotlin

Kotlin is a Concise, Simple,Safe and Statically typed programming language focused on Interoperability with Java.

One can Avoid entire classes of errors such as null pointer exceptions. Improves code readability as it Drastically reduces the amount of boilerplate code. While the syntax is not compatible with Java, Kotlin is reliant on Java code from the existing Java Class Library, such as the collections framework. Kotlin is a fully supported programming language on Android.

3.1.3 Django(Web Framework)

Django is a free and open-source web framework, written in Python, which follows the model-view-template (MVT) architectural pattern.

Django's primary goal is to ease the creation of complex, database-driven websites. Django emphasizes reusability and "pluggability" of components, rapid development, and the principle of don't repeat yourself. Python is used throughout, even for settings files and data models. Django also provides an optional administrative create, read, update and delete interface that is generated dynamically through introspection and configured via admin models.

3.1.4 Git

Git is a version control system for tracking changes in computer files and coordinating work on those files among multiple people. It is primarily used for source code management in software development, but it can be used to keep track of changes in any set of files. As a distributed revision control system it is aimed at speed, data integrity, and support for distributed, non-linear workflows.

3.1.5 RESTful API

Representational state transfer (REST) or RESTful web services is a way of providing interoperability between computer systems on the Internet. REST-compliant Web services allow requesting systems to access and manipulate textual representations of Web resources using a uniform and predefined set of stateless operations.

An API for a website is code that allows two software programs to communicate with each another.A RESTful API explicitly takes advantage of HTTP methodologies. They use GET to retrieve a resource; PUT to change the state of or update a resource, which can be an object, file or block; POST to create that resource; and DELETE to remove it.

3.1.6 JSON

JavaScript Object Notation or JSON is an open-standard file format that uses human-readable text to transmit data objects consisting of attribute-value pairs and array data types (or any other serializable value). It is a very common data format used for asynchronous browser-server communication.

JSON is built on two structures: A collection of name/value pairs. In various languages, this is realized as an object, record, struct, dictionary, hash table, keyed list, or associative array. An ordered list of values. In most languages, this is realized as an array, vector, list, or sequence.

3.1.7 Android

Android is a mobile operating system developed by Google, based on the Linux kernel and designed primarily for touchscreen mobile devices such as smartphones and tablets.

Android's user interface is mainly based on direct manipulation, using touch gestures that loosely correspond to real-world actions, such as swiping, tapping and pinching, to manipulate on-screen objects, along with a virtual keyboard for text input.

3.1.8 Android Studio

The Official IDE for Android. Android Studio provides the fastest tools for building apps on every type of Android device.

World-class code editing, debugging, performance tooling, a flexible build system, and an instant build/deploy system all allow you to focus on building unique and high quality apps.

3.1.9 SQLite

SQLite is an in-process library that implements a self-contained, serverless, zero-configuration, transactional SQL database engine. The code for SQLite is in the public domain and is thus free for use for any purpose, commercial or private.

SQLite is an embedded SQL database engine. Unlike most other SQL databases, SQLite does not have a separate server process.SQLite is a compact library. With all features enabled, the library size can be less than 500KiB. SQLite a popular database engine choice on memory constrained gadgets such as cellphones, PDAs, and MP3 players. Performance is usually quite good even in low-memory environments.

3.1.10 Open Source Libraries for Android

Glide Glide is a fast and efficient open source media management and image loading framework for Android that wraps media decoding, memory and disk caching, and resource pooling into a simple and easy to use interface.

Retrofit A type-safe HTTP client for Android and Java. This library provides a powerful framework for authenticating and interacting with APIs and sending network requests with OkHttp network. This library makes downloading JSON or XML data from a web API fairly straightforward. Once the data is downloaded then it is parsed into a Plain Old Java Object (POJO).

Stetho Stetho is a sophisticated debug bridge for Android applications developed by facebook. When enabled, developers have access to the Chrome Developer Tools feature natively part of the Chrome desktop browser. Network inspection is possible with the full spectrum of Chrome Developer Tools features, including image preview, JSON response helpers. SQLite databases can be visualized and interactively explored with full read/write capabilities.

ProGuard ProGuard is the most popular optimizer for Java bytecode. It makes your Java and Android applications up to 90% smaller and up to 20% faster. ProGuard also provides minimal protection against reverse engineering by obfuscating the names of classes, fields and methods.

3.2 Implementation Details

3.2.1 Simulation Parameters

The Android SDK includes a virtual mobile device emulator that runs on your computer. The emulator lets you prototype, develop and test Android applications without using a physical device.

Unit Testing

This is the first level of testing. In this different modules are tested against the specifications produced during the design of the module. During this testing the number of the arguments is compared to input parameters, matching of parameter and arguments etc. All five modules are checked separately, each test case is given to each unit and it is checked. The result is checked to see if the actual outcome is same as the expected outcome.A unit test is also called a module test because it tests the individual units of code that comprise the application. Each test validates a single module that was built to perform a certain task with the expectation that it will behave in a specific way. During this testing the number of the arguments is compared to input parameters, matching of parameter and arguments etc. All five modules are checked separately, each test case is given to each unit and it is checked. The result is checked to see if the actual outcome is same as the expected outcome.A unit test is also called a module test because it tests the individual units of code that comprise the application. Each test validates a single module that was built to perform a certain task with the expectation that it will behave in a specific way.

Integration testing

Integration testing is a process where all the separate modules are combined together and its working is checked. There are two approaches to this : bottom-up and top-down integration.At first the branch, semester and grade selection modules are integrated and checked.Top-down approach is being used in this software testing.

User acceptance testing

User acceptance of a system is the factor for the success of any system. The system under consideration is tested for the user acceptance by constantly keeping in touch with the prospective system user at the time of developing wherever required.

- Input screen design.
- Output screen design.
- Online message to guide the user.
- Format of the reports and other outputs

Functional Testing

Functional testing ensures that the application is working as per the requirements. Most of the test conducted for this is driven by the user interface and call flow

Performance Testing

This testing process is undertaken to check the performance and behavior of the application under certain conditions such as low battery, bad network coverage, low available memory, simultaneous access to applicationâĂŹs server by several users and other conditions. Performance of an application can be affected from two sides:applicationâĂŹs server side and clientâĂŹs side. Performance testing is carried out to check both.

Memory Leakage Testing

Memory leakage happens when a computer program or application is unable to manage the memory it is allocated resulting in poor performance of the application and the overall slowdown of the system. As mobile devices have significant constraints of available memory, memory leakage testing is crucial for the proper functioning of an application

Interrupt Testing

An application while functioning may face several interruptions like incoming calls or network coverage outage and recovery. The different types of interruptions are:

- Incoming and Outgoing SMS and MMS
- Incoming and Outgoing calls
- Incoming Notifications
- Battery Removal
- Cable Insertion and Removal for data transfer
- Network outage and recovery
- Media Player on/off

An application should be able to handle these interruptions by going into a suspended state and resuming afterwards.

Usability testing

Usability testing is carried out to verify if the application is achieving its goals and getting a favorable response from users. This is important as the usability of an application is its key to commercial success (it is nothing but user friendliness).Another important part of usability testing is to make sure that the user experience is uniform across all devices.This section of testing hopes to address the key challenges of the variety of mobile devices and the diversity in mobile platforms/OS, which is also called device fragmentation. One key portion of this type of usability testing is to be sure that there are no major errors in the functionality, placement, or sizing of the user interface on different devices.

Installation testing

Certain mobile applications come pre-installed on the device whereas others have to be installed from the store. Installation testing verifies that the installation process goes smoothly without the user having to face any difficulty. This testing process covers installation, updating and uninstalling of an application

Security Testing

Security Testing: To check for vulnerabilities to hacking, authentication and authorization policies, data security, session management and other security standards.

Results

Testing Parameters	Expected Output	Actual Output	Result
Authentication	Authenticate user by validating credentials from api	User is Authenticated	Pass
Fetch Faculty Timetable	Fetch faculty time-table from the API	Time-table fetched and stored locally	Pass
Fetch Class Timetable	Fetch class time-table from the API	Time-table fetched and stored locally	Pass
Fetch Class Location	Fetch class location from the API	class location displayed	Pass
Check Faculty Availability	fetch available faculty from the database	displayed available faculty	Pass
Admin Portal Authentication	Authenticate Admin	Credentials Validated	Pass
Admin Portal-Update/modify Database	Changes should reflect in the app	Data Changed	Pass

Table 3.1: Test-Cases

Name of Test	API Level	Test Result
Unit testing	Lollipop , Nougat	Pass
Integration testing	Lollipop , Nougat	Pass
User Acceptance testing	Lollipop , Nougat	Pass
Functional testing	Lollipop , Nougat	Pass
Usability testing	KitKat , Nougat	Pass
Memory Leakage testing	Marshmellow , Nougat	Pass
Performance testing	KitKat , Nougat	Pass
Interrupt testing	Marshmellow , Nougat	Pass
Installation tests	Lollipop , Nougat	Pass
Security Testing	Lollipop , Nougat	Pass

Table 3.2: Test-Results

3.2.2 Sample coding

Listing 3.1: Retrofit Interface

```
package com.notadeveloper.app.blackboard

import com.facebook.stetho.okhttp3.StethoInterceptor
import io.reactivex.Observable
import okhttp3.OkHttpClient
import retrofit2.Retrofit
import retrofit2.adapter.rxjava2.RxJava2CallAdapterFactory
import retrofit2.converter.moshi.MoshiConverterFactory
import retrofit2.http.Field
import retrofit2.http.FormUrlEncoded
import retrofit2.http.POST

interface RetrofitInterface {
@FormUrlEncoded
@POST("login")
fun authUser(@Field("user") user: String, @Field("pass") pass: String): Observable<
    Faculty>

@FormUrlEncoded
@POST("getfacultytimetable")
fun getfacultytimetable(@Field("user") user: String, @Field("pass") pass: String,
    @Field("id") id: String): Observable<timetable>

@FormUrlEncoded
@POST("getclasstimetable")
fun getclasstimetable(@Field("user") user: String, @Field("pass") pass: String, @Field
    ("class") class: String): Observable<timetable>

@FormUrlEncoded
@POST("getavailablestaff")
fun getavailablestaff(@Field("user") user: String, @Field("pass") pass: String, @Field
    ("day") day: String, @Field("hour") hour: String): Observable<List<String>>

companion object Factory {
fun create(): RetrofitInterface {
val retrofit = Retrofit.Builder()
.addCallAdapterFactory(RxJava2CallAdapterFactory.create())
.addConverterFactory(MoshiConverterFactory.create().asLenient())
.client(OkHttpClient.Builder().addNetworkInterceptor(StethoInterceptor()).build())
.baseUrl("http://blackboard.xyz/")
.build()
}
}
}
```

Listing 3.2: Faculty Dashboard

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
xmlns:app="http://schemas.android.com/apk/res-auto"
xmlns:tools="http://schemas.android.com/tools"
android:id="@+id/activity_main"
android:layout_width="match_parent"
android:layout_height="match_parent"
android:paddingBottom="@dimen/activity_vertical_margin"
android:paddingLeft="@dimen/activity_horizontal_margin"
android:paddingRight="@dimen/activity_horizontal_margin"
android:paddingTop="@dimen/activity_vertical_margin"
tools:context="com.example.lenovo.bb.HodActivity">

<Button
android:text="Available staff"
android:layout_width="match_parent"
android:layout_height="wrap_content"
android:layout_below="@+id/button"
android:theme="@style/button"
android:id="@+id/button2" />

<Button
android:text="class location"
android:layout_width="match_parent"
android:layout_height="wrap_content"
android:layout_below="@+id/button2"
android:theme="@style/button"
android:id="@+id/button3"
/>

<Button
android:text="class time table"
android:layout_width="match_parent"
android:layout_height="wrap_content"
android:id="@+id/button4"
android:layout_below="@+id/imageView2"
android:theme="@style/button"
android:layout_centerHorizontal="true"
/>

<ImageView
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:src="@drawable/blckbrdcover"
android:id="@+id/imageView2"
android:adjustViewBounds="true"
android:scaleType="fitXY"
```

```
android:layout_alignParentTop="true"
android:layout_alignParentLeft="true"
/>

<Button
android:text="Staff time table and responsibilities"
android:layout_width="match_parent"
android:layout_height="wrap_content"
android:id="@+id/button"
android:theme="@style/button"
android:layout_below="@+id/button4"
android:layout_alignParentLeft="true"
android:layout_alignParentStart="true" />
</RelativeLayout>
```

3.2.3 Screen Shots

Faculty Login (fig3.1)

Every faculty whomsoever install the application will have to provide its email-id, password and phone number during its one time registration process. These informations are first matched with the online database and then stored in mobile database.

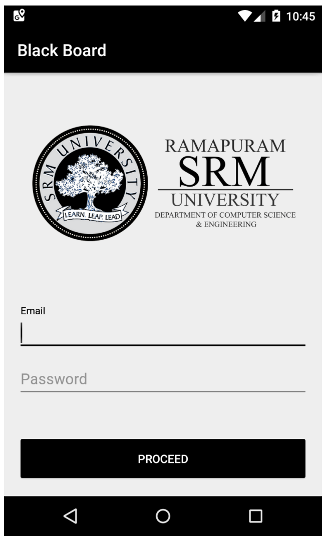


Figure 3.1: Faculty Login

Faculty Dashboard (fig3.2)

All information related to the particular person available in one application. Faculty can check his/her timetable, class location, class timetable with the click of a button.

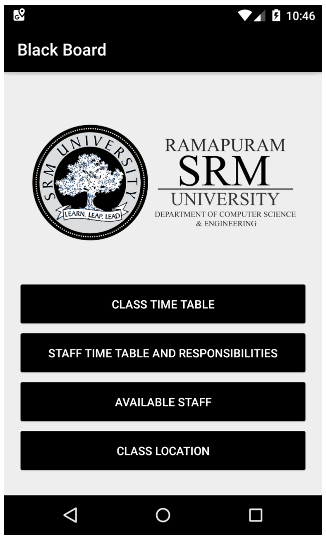
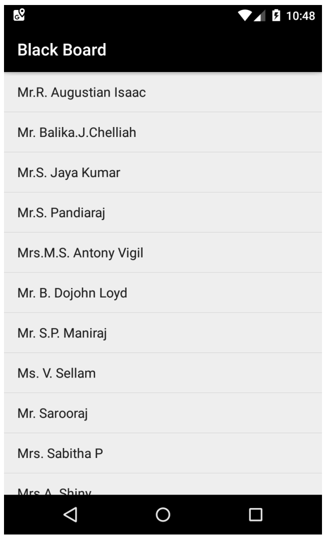


Figure 3.2: Faculty Dashboard

Check available faculty (fig3.3)

Hod can easily check the available faculty by selecting the day and hour.


10:48

Black Board

Mr.R. Augustian Isaac
Mr. Balika.J.Chelliah
Mr.S. Jaya Kumar
Mr.S. Pandiaraj
Mrs.M.S. Antony Vigil
Mr. B. Dojohn Loyd
Mr. S.P. Maniraj
Ms. V. Sellam
Mr. Sarooraj
Mrs. Sabitha P
Mrs.A. Shiny

Figure 3.3: Check Available Faculty

Faculty Timetable (fig3.4)

The Hod can check the timetable of a particular faculty by querying the database.

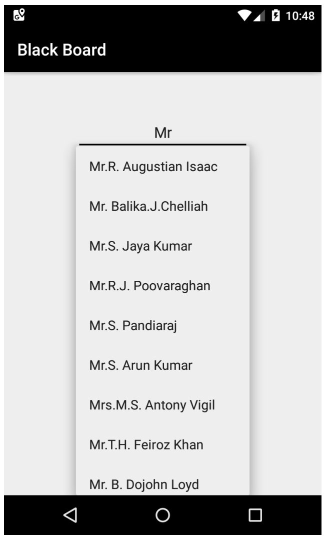


Figure 3.4: Faculty Timetable

Check Class Timetable (fig3.5)

The faculty can check the timetable of a particular class by selecting the year, section of the class.

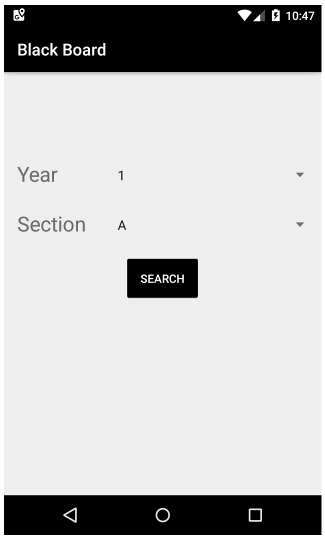


Figure 3.5: Class Timetable

Class Location (fig3.6)

The faculty can check the location of a particular class by selecting the dept, year, section of the class

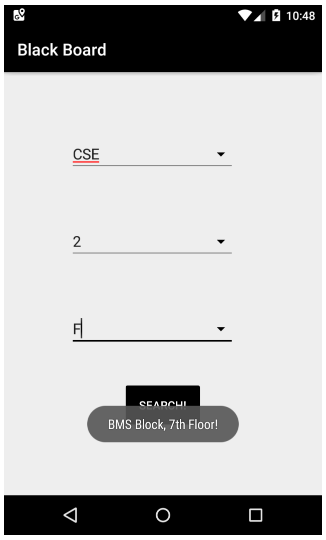


Figure 3.6: Class Location

Admin Login (fig3.7)

Login page for the Webportal where admin can login to change data.

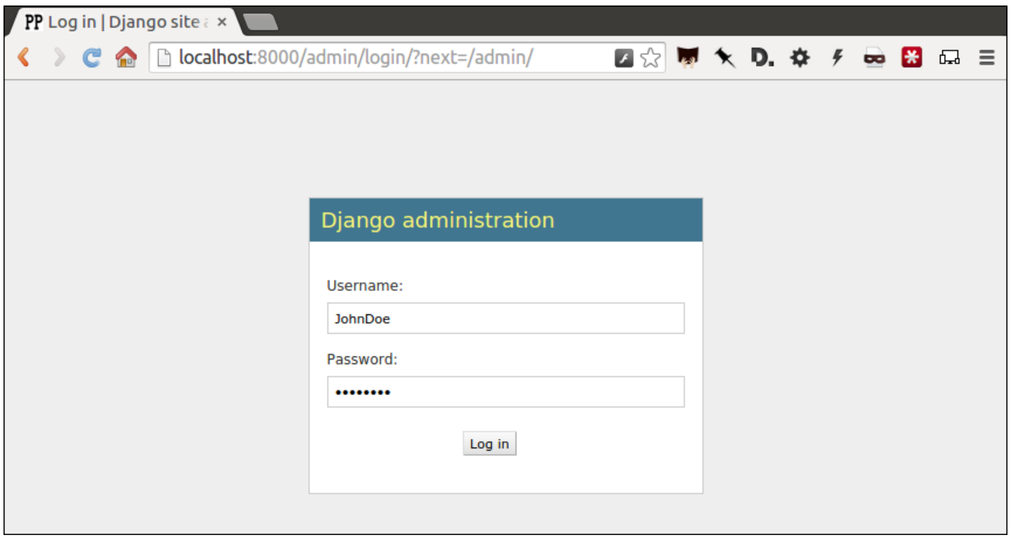


Figure 3.7: BlackBoard Admin Login

Admin Dashboard (fig3.8)

Dashboard for the admin Webportal where database can be modified/updated/added/deleted.

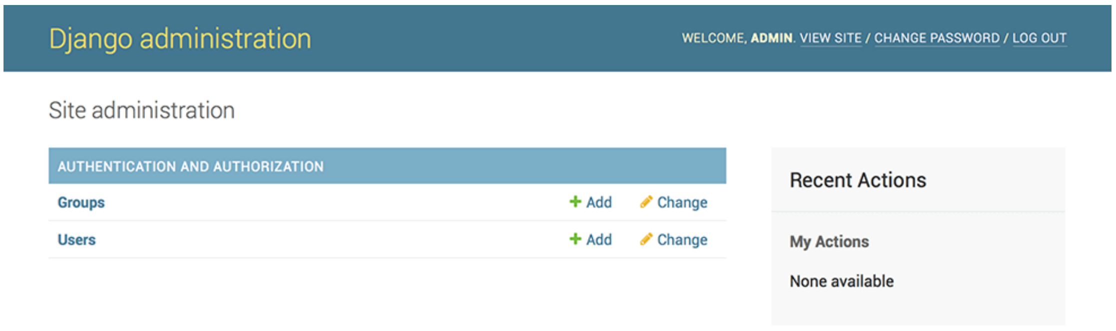


Figure 3.8: BlackBoard Admin Dashboard

3.3 Summary

The mobile application, incorporating the college management system, is a very effective tool which can be used for improving the overall efficiency in a college/university.During these tests, the number of arguments are compared to input parameters, matching of parameters and arguments etc.

The Application was tested on an emulator which lets you prototype, develop and test Android applications without using a physical device. The application performed extremely well in low resource environments where the storage space and network connection is slow. Memory leaks were detected and resolved during development using Leak Canary, Proguard has been used to optimize the code(remove unused resources) and also to obfuscate the core logic for security.

CHAPTER 4

INTRODUCTION

4.1 Inverse Problem

Structural Identification (SI) is typically an inverse process whereby structural parameters such as stiffness, damping properties are identified from input excitation and output responses.

Generally, Engineering problems can be classified into forward and inverse problems (Koh et al., 2003). In forward problems, the system output responses are calculated from the known system properties and input responses as shown in Figure 4.1 whereas in inverse problems, the system parameters are identified based on the input and output responses of the system which is shown in Figure 4.2. For a structure, the

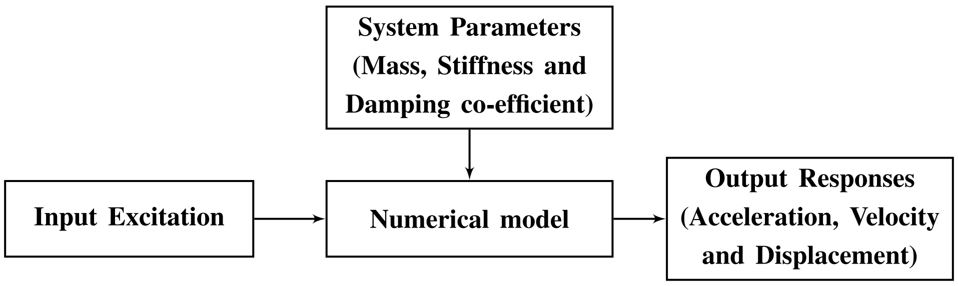

System Parameters (Mass, Stiffness and Damping co-efficient)
Input Excitation
Numerical model
Output Responses (Acceleration, Velocity and Displacement)

Figure 4.1: Forward problem

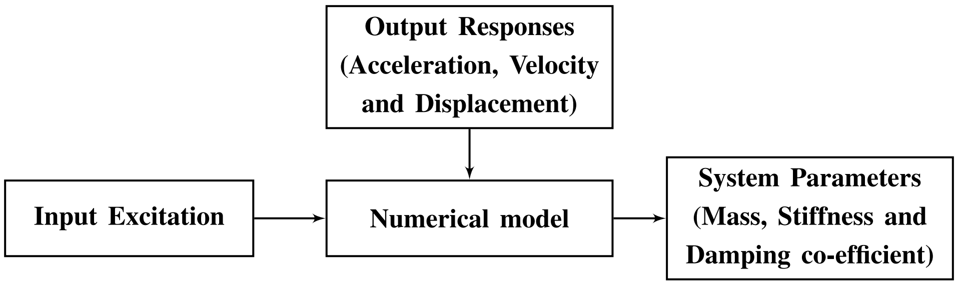

Output Responses (Acceleration, Velocity and Displacement)
Input Excitation
Numerical model
System Parameters (Mass, Stiffness and Damping co-efficient)

Figure 4.2: Inverse problem

input excitation is a periodic force and the output responses are displacement, velocity and acceleration. The input force can be measured using force transducer and the output responses can be measured respectively using vibration pick-ups, velometer and accelerometer. Some SI algorithms require measurement of all responses or any one of the output responses. Since the input and output responses are measurable for a structure with unknown parameters, the SI problem is an inverse problem which identifies structural or damage parameters.

4.1.1 Sub sections

Sub-sections must be numbered as shown in this text.

Sub-sub sections

Sub-sections of sub section are not to be numbered and it should be in bold as shown in this text.

CHAPTER 5

LITERATURE SURVEY

5.1 Frequency Domain SI

Hearn and Testa (1989) showed that the magnitude of change in natural frequencies is a function of the severity and of the location of deterioration in structures. The modal analysis has been carried out on a welded steel frame and a wire rope with damage. Grédias and Paris (1996) proposed a direct method for determining six flexural stiffnesses of a thin anisotropic plate. In this method, natural frequencies (Meirovitch, 2001) and mode shapes have been processed using Least Square (LS) technique.

5.2 Particle Swarm Optimization

A basic variant of the Particle Swarm Optimization (PSO) (James and Eberhart, 1995) algorithm works by having a population (called a swarm) of candidate solutions (called particles). These particles are moved around in the search-space according to a few simple formulae. The movements of the particles are guided by their own best known position in the search-space as well as the entire swarm's best known position. When improved positions are being discovered these will then come to guide the movements of the swarm. The process is repeated and by doing so it is hoped, but not guaranteed, that a satisfactory solution will eventually be discovered.

CHAPTER 6

SYSTEM ANALYSIS

All the symbols used in the text must be listed out in the list of symbols page in alphabetic order, and explained with units. Equations must be written with central alignment and numbered with section number . equation number.

$$[M^e]\{\ddot{q}(t)\} + [C^e]\{\dot{q}(t)\} + [K^e]\{q(t)\} = \{f(t)\} \tag{6.1}$$

the element stiffness matrix is

$$[K^e] = \frac{EI}{l_e^3}\begin{bmatrix} 12 & 6l_e & -12 & 6l_e \\ 6l_e & 4l_e^2 & -6l_e & 2l_e^2 \\ -12 & -6l_e & 12 & -6l_e \\ 6l_e & 2l_e^2 & -6l_e & 4l_e^2 \end{bmatrix} \tag{6.2}$$

and the element consistent mass matrix is

$$[M^e] = \frac{\rho A l_e}{420}\begin{pmatrix} 156 & 22l_e & 54 & -13l_e \\ 22l_e & 4l_e^2 & 13l_e & -3l_e^2 \\ 54 & 13l_e & 156 & -22l_e \\ -13l_e & -3l_e^2 & -22l_e & 4l_e^2 \end{pmatrix} \tag{6.3}$$

$$[C] = \alpha[M] + \beta[K] \tag{6.4}$$

A matrix equation with square matrices and vectors,

$$\begin{Bmatrix} -M_1(t) \\ -V_1(t) \\ M_2(t) \\ V_2(t) \end{Bmatrix} = \begin{bmatrix} D_{11} & D_{12} & D_{13} & D_{14} \\ D_{21} & D_{22} & D_{23} & D_{24} \\ D_{31} & D_{32} & D_{33} & D_{34} \\ D_{41} & D_{42} & D_{43} & D_{44} \end{bmatrix} \begin{Bmatrix} v_1(t) \\ \theta_1(t) \\ v_2(t) \\ \theta_2(t) \end{Bmatrix} \tag{6.5}$$

CHAPTER 7

SYSTEM DESIGN

7.1 Tables and Figures

By the word Table, is meant tabulated numerical data in the body of the project report as well as in the appendices. All other non-verbal materials used in the body of the project work and appendices such as charts, graphs, maps, photographs and diagrams may be designated as figures[1]. The format for tables is given below. The table must referred in the text as Table. (**??**). The title of the table with table number should be written at the top of the table with center aligned as shown below.[2].

Title of figures must be center aligned and should be displayed at the bottom of the figure as shown.

Multiple figures with same caption can be arranged as shown in Figure **??** and they are referred in text such as Figure **??** and Figure **??**. The graphs should be drawn at appropriate places with center alignment and it should be referred in text.

[1]type footnote here
[2]An example footnote.

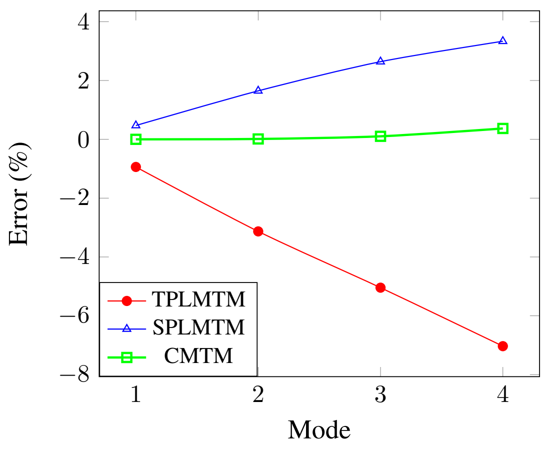


Figure 7.1: Error in natural frequencies

CHAPTER 8

CODING, TESTING

In this chapter, the program coding related to your work using MATLAB™or C can be presented here. Numbered list can be typeset in LaTex as follows.

1. **Online LaTex editors** Such as **Share LaTex**, **Write LaTex**, **Papeeria** are also available which do not require any installation.

2. Documents can be typeset or edited in the online LaTex and output file can be down loaded.

3. The LaTex template can generally be down loaded from the University/publisher's/ conference website. (.tex file)

4. Type the document in the respective template and run the program like C or MATLAB.

5. The output document is formatted as .pdf or .ps and saved in the same file name and same folder.

6. Some Standard LaTex templates are *IEEE*, ASCE, ASME, Elsevier etc.

Bullet points are generated in LaTex as follows:

- Requires installation of two packages.
- **MiKTeX** is a distribution of the typesetting system.
- MiKTeX provides the tools necessary to prepare documents using the TeX/LaTeX markup language.
- **TeXstudio** is a cross-platform open source LaTeX editor with an interface.
- Some other cross-platforms are Texmaker, Technic center, Winedt etc.,
- Install MiKTex first and TeXstudio at last.

CHAPTER 9

CONCLUSION

The Entire Project Document should have a **maximum of 80 Pages** (from cover to cover).

The Project Document along with Application Software should be submitted in a Soft Copy (CD)

N.B.: Number of Copies to be submitted: Guide - 1 hard copy, Department Library -1 hard copy & Each Candidate -1 hard copy and Soft Copy (in CD)- 2 copies

CHAPTER 10

FUTURE ENHANCEMENT

APPENDIX A

VECTOR ALGEBRA

A.1 Product of Two Vectors

The product of two vectors are may be a scalar product or vector product. The scalar product of two vectors is also called as dot product. It is defined as $\vec{a}.\vec{b} = |\vec{a}||\vec{b}|cos\theta$ where θ is the angle between the two vectors $\vec{a}$ and $\vec{b}$

the cross product or vector product is a binary operation on two vectors in three-dimensional space and is denoted by the symbol $\times$. The cross product $\vec{a} \times \vec{b}$ of two linearly independent vectors $\vec{a}$ and $\vec{b}$ is a vector that is perpendicular to both and therefore normal to the plane containing them.

APPENDIX B

MATRIX ALGEBRA

B.1 Matrix Multiplication

Matrix multiplication is a binary operation that takes a pair of matrices, and produces another matrix. Numbers such as the real or complex numbers can be multiplied according to elementary arithmetic. On the other hand, matrices are arrays of numbers, so there is no unique way to define multiplication of matrices. As such, in general the term “matrix multiplication” refers to a number of different ways to multiply matrices.

REFERENCES

1. Grédias, M. and Paris, P. A. (1996). “Direct identification of elastic constants of anisotropic plates by modal analysis:theoretical and numerical aspects.” *Journal of Sound and Vibration*, 195(3), 401–415.

2. Hearn, G. and Testa, R. B. (1989). “Modal analysis for damage detection in structures.” *Journal of structural engineering*, 117(10), 3042–3063.

3. James, K. and Eberhart, R. (1995). “Particle swarm optimization.” *In Proc.of IEEE International Conference on Neural Networks, 1995.*, Vol. 4, 1942 –1948.

4. Koh, C. G., Hong, B., and Liaw, C. Y. (2003). “Substructural and progressive structural identification methods.” *Engineering Structures*, 25, 1551–1563.

5. Meirovitch, L. (2001). *Fundamentals of Vibrations*. McGraw-Hill Book Company, 1 edition.