



VISVESVARAYA TECHNOLOGICAL UNIVERSITY
“JNANA SANGAMA”, BELGAUM-590018.



A
Mini Project Report
On

SIMPLE DATABASE APPLICATION

SUBMITTED IN FULFILLMENT OF THE REQUIREMENT IN
6TH SEMESTER MOBILE APPLICATION DEVELOPMENT LABORATORY MINI
PROJECT WORK OF BACHELOR OF ENGINEERING
IN

COMPUTER SCIENCE AND **ENGINEERING**



By
Suhitha LR (1JB18CS153)
Sumit Prakash (1JB18CS155)

UNDER THE GUIDANCE OF

Mrs. RAJATHA
Assistant Professor
Dept. of CSE

ARUN KUMAR D.R
Assistant Professor
Dept. of CSE



DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

S. J. B INSTITUTE OF TECHNOLOGY

B G S HEALTH AND EDUCATION CITY

Kengeri, Bangalore-560060.

2020-2021

||Jai Sri Gurudev||
Sri Adichunchanagiri Shikshana Trust ®

S. J. B INSTITUTE OF TECHNOLOGY
BGS Health & Education City, Kengeri, Bangalore-560060.

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING



CERTIFICATE

This is to certify that the technical seminar report entitled **“Simple Database Application”** is a bonafied work carried out **Suhitha LR [1JB18CS153]** and **Sumit Prakash [1JB18CS155]**, in partial fulfillment of mini project of 6th semester Mobile Application Development Laboratory in **COMPUTER SCIENCE AND ENGINEERING** of **VISVESVARAYA TECHNOLOGICAL UNIVERSITY, BELGAUM**, during the academic year **2020-2021**. It is certified that all corrections/suggestions have been incorporated in the report deposited in the department library. The mini project report has been approved as it satisfies the academic requirements.

Signature of the guide
[Mrs.RAJATHA]
Assistant Professor
Dept. of CSE

Signature of the HOD
[Dr. KRISHNA A N]
Professor & Head
Dept. of CSE

Signature of the guide
[ARUN KUMAR D.R.]
Assistant Professor
Dept. of CSE

Name of the Examiner

1. _____

2. _____



ACKNOWLEDGEMENT

We would like to express our profound gratefulness to His Divine Soul **Jagadguru Padmabhushan Sri Sri Sri Dr. Balagangadharanatha Mahaswamiji** and His Holiness **Jagadguru Sri Sri Sri Dr. Nirmalanandanatha Mahaswamiji** for providing us with an opportunity to study in this esteemed Institution.

We would like to express our profound thanks to **Reverend Sri Sri Dr. Prakashnath Swamiji**, Managing Director, SJB Institute of Technology, for his continuous support in providing amenities to carry out this mini project in this admired institution.

We express my gratitude to **Dr. Ajai Chandran C K, Principal**, SJB Institute of Technology, for providing me with excellent facilities and academic ambience; which have helped us in satisfactory completion of this mini project.

We extend our sincere thanks to **Dr. Krishna A N**, Head of the Department of Computer Science and Engineering, for providing us with invaluable support throughout the period of our mini project.

We wish to express our heartfelt gratitude to our guides, **Mrs. Rajatha and Arun Kumar D.R, Assistant Professors** for their valuable guidance, suggestions and cheerful encouragement during the entire period of this min project work.

Finally, we take this opportunity to extend our earnest gratitude and respect to our parents, Teaching & Non-teaching staffs of the department, the library staff and all our friends, who have directly or indirectly supported us during the period of our mini project work.

Regards,

Suhitha LR (1JB18CS153)

Sumit Prakash (1JB18CS155)

ABSTRACT

Student Database application provides a simple interface for maintenance of student information. It can be used by educational institutes or colleges to maintain the records of students easily. The creation and management of accurate, up-to-date information regarding a students' academic career is critically important in the university as well as colleges. Student information system deals with all kind of student details, academic related reports, college details, course details, curriculum, batch details, placement details and other resource related details too. It tracks all the details of a student from the day one to the end of the course which can be used for all reporting purpose, tracking of attendance, progress in the course, completed semesters, years, coming semester year curriculum details, exam details, project or any other assignment details, final exam result and all these will be available through a secure, online interface embedded in the college's website. It will also have faculty details, batch execution details, students' details in all aspects, the various academic notifications to the staff and students updated by the college administration. It also facilitate us explore all the activities happening in the college, Different reports and Queries can be generated based on vast options related to students, batch, course, faculty, exams, semesters, certification and even for the entire college.

Student database application holds the record of the USN, Name and other details of student. We can display, delete and update the information of the student. It give complete security that means only the administrator can delete ,update and display the information of the student.

TABLE OF CONTENTS

Table of Contents

CONTENTS	Page No.
ABSTRACT	
ACKNOWLEDGEMENT	
1. INTRODUCTION	8
1.1 About Android Studio	8
1.1.1 Features	8
1.2 About Project	9
1.2.1. Scope of The Project	10
2. SYSTEM REQUIREMAENTS	11
2.1 Basic System Software Requirements for Android Studio	11
2.1.1 Operating System Version	11
2.2 Basic System Hardware Requirements for Android Studio	12
3. SYSTEM DESIGN	15
3.1 Architecture Design	13
3.1.1 Introduction	13
3.2 Dataflow Diagram	14
3.3 Sequence Diagram	16
3.4 ER Diagram	17
3.5 Schema Diagram	19
4. IMPLEMENTATION	20
4.1 Module Description	20
4.2 Library Used for Project	20
4.3 Algorithm of the project	23
5. RESULTS	26
5.1 Insert	26
5.2 Delete	27
5.3 Update	29
CONCLUSION	31
BIBLOGRAPHY	32

List of Figures

Figures	Page No
3.1-Airchitectutre Design	13
3.2.1-Data Flow Diagram Level 1 of Simple Database Application	14
3.2.2- Data Flow Diagram Level 2 of Simple Database Application	15
3.3-Sequence Diagram	16
3.4-ER Diagram	18
3.5-Schema Diagram	19
5.1-Insert Application Snapshot	26
5.2-Delete Application Snapshot	27
5.3-Update application Snapshot	30

Chapter 1

INTRODUCTION

1.1 ABOUT ANDROID STUDIO

Android Studio is the official integrated development environment (IDE) for Google's Android operating system, built on JetBrains IntelliJ IDEA software and designed specifically for Android development. It is available for download on Windows Mac and Linux based operating systems or as a subscription-based service in 2020. It is a replacement for the Eclipse Android Development Tool (E-ADT) as the primary IDE for native Android application development.

Android Studio was announced on May 16, 2013 at the Google I/O conference. It was in early access preview stage starting from version 0.1 in May 2013, then entered beta stage starting from version 0.8 which was released in June 2014. The first stable build was released in December 2014, starting from version 1.0.

On May 7, 2019, Kotlin replaced Java as Google's preferred language for Android app development. Java is still supported, as is C++.

1.1.1 FEATURES

A specific feature of the Android Studio is an absence of the possibility to switch auto save feature off.

The following features are provided in the current stable version.

- Gradle -based build support
- Android-specific refactoring and quick fixes
- Lint tools to catch performance, usability, version compatibility and other problems
- Pro Guard integration and app-signing capabilities
- Template-based wizards to create common Android designs and components
- A rich layout editor that allows users to drag-and-drop UI components, option to preview layouts on multiple screen configurations
- Support for building Android Wear apps
- Built-in support for Google Cloud Platform, enabling integration with Firebase Cloud Messaging (Earlier 'Google Cloud Messaging') and Google App Engine
- Android Virtual Device (Emulator) to run and debug apps in the Android studio.

Android Studio supports all the same programming languages of IntelliJ (and CLion) e.g. Java, C++, and more with extensions, such as Go and Android Studio 3.0 or later supports Kotlin and "all Java 7 language features and a subset of Java 8 language features that vary by platform version." External projects back port some Java 9 features. While IntelliJ states that Android Studio supports all released Java versions, and Java 12, it's not clear to what level Android Studio supports Java versions up to Java 12 (the documentation mentions partial Java 8 support). At least some new language features up to Java 12 are usable in Android.

1.2 ABOUT PROJECT

Managing an educational system requires careful planning and time management. Today, a school has evolved into a complex institution with multiple campuses, manifold departments, several sections and a very large number of students. For schools moving from a traditional paper-based data management system to a digital, automated system is a critical need of the hour. In its simplest form, it uses a digital tracking system to maintain a record of all the students. However, in practicality, the system has much more complex functions. It maintains a record of every student-related data ranging from fees and financial records, examination records, transport facilities being provided by the school and availed by the students along with the usage of libraries, laboratory, computer lab and other school facilities used regularly by the students.

A student database management system allows schools to save and access these records as needed by them, thus simplifying the work of the school administration team.

Let's have a look at some top benefits

- ***Streamlined Activities***

The student database management system allows for reminders to set up for the various events and activities that need to be performed by the school administration team. It gives a comprehensive look at all the data on a single screen and ensures that all essential data is highlighted. This results in better productivity and growth with minimum effort.

- ***Better Communication***

A student database management system allows for easier communication between students, parents-teachers, students-teachers and even allows for the school alumni to stay connected with each other.

1.2.1. SCOPE OF THE PROJECT

modern times, facilities offered by schools are not limited to basic functioning instead, the authorities have been looking for advanced system. This system is aimed at total user-friendly as well as efficient management of varied tasks. These tasks may range from registering new students, managing fees payment, examination management to all the essential features necessary for making the administrative division of school effective. In order to cope up with all these factors, the school management system was developed and nowadays, it has even been recognized by most of the Indian schools. As a matter of fact, this system based on smart technology has become an integral part of many schools.

this segment, it is crucial to discuss the purpose served by before proceeding. To begin with, the school management system is basically manufactured to compile all the manual activities of administrative importance in the form of software. This software further makes it easier for officials to finish off their work in a lesser span of time. Most of all, the mechanism of software is easy to understand that even if any school is utilizing it for the first time, the users will not have to toil hard to learn its function. On the other hand, there is a vast range of apps that are included in this software for different streams of management in any school. For instance, if you have purchased a school app or similar software, then, its various modules will make your work simpler yet very accurate.

Chapter 2

SYSTEM REQUIREMENT

2.1 Basic system Software requirements for Android Studio

2.1.1 Operating System Version

- Windows 7/8/10 (32- or 64-bit)
- The Android Emulator only supports 64-bit Windows
- Mac OS X 10.10 (Yosemite) or higher, up to 10.14 (mac OS Mojave)
- GNOME or KDE desktop
- Tested on g Linux based on Debin(4.19.67-2rodete2)
- Free Digital Source
- 2 GB of available digital storage minimum, 4 GB recommended (500 MB for IDE + 1.5 GB for Android SDK and emulator system image).
- **Minimum required JDK version**
- JAVA Development Kit 8

The Android Emulator has additional requirements beyond the basic system requirements for Android Studio, which are described below:

- SDK Tools 26.1.1 or higher;
- 64-bit processor;
- Windows: CPU with UG (unrestricted guest) support;
- Intel Hardware Accelerated Execution Manager (**HAXM**) 6.2.1 or later (HAXM 7.2.0 or later recommended).

The use of hardware acceleration has additional requirements on Windows and Linux:

- Intel processor on Windows or Linux: Intel processor with support for Intel VT-x, Intel EM64T (Intel 64), and Execute Disable (XD) Bit functionality;

- AMD processor on Linux: AMD processor with support for AMD Virtualization (AMD-V) and Supplemental Streaming SIMD Extensions 3 (SSSE3);
- AMD processor on Windows: Android Studio 3.2 or higher and Windows 10 April 2018 release or higher for Windows Hypervisor Platform (WHPX) functionality.

To work with Android 8.1 (API level 27) and higher system images, an attached webcam must have the capability to capture 720p frames.

2.2 Basic system Hardware requirements for Android Studio

- Peripherals - Keyboard, Mouse
- 64-bit Microsoft Windows 8/10.
- X86_64 CPU architecture; 2nd generation Intel Core or newer, or AMD CPU with support for a Windows Hypervisor.
- 8 GB RAM or more.
- 8 GB of available disk space minimum (IDE + Android SDK + Android Emulator)
- 1280 x 800 minimum screen resolutions.

Chapter 3

System Design

3.1 Architecture Design

3.1.1 Introduction

The purpose of this document is to provide an architectural design for the Book Cart. The design will show the presentation tier, the middle tier consisting of classes, sequence diagrams, and the data tier consisting of the database design diagram

Three-tier (layer) is a client-server architecture in which the user interface, business process (business rules) and data storage and data access are developed and maintained as independent modules or most often on separate platforms. The Architecture of Book Cart is based on three-tier architecture. The three logical tiers are

- Presentation tier – Android studios.

- Application tier – JAVA language, XML.

- Data tier- SQL.

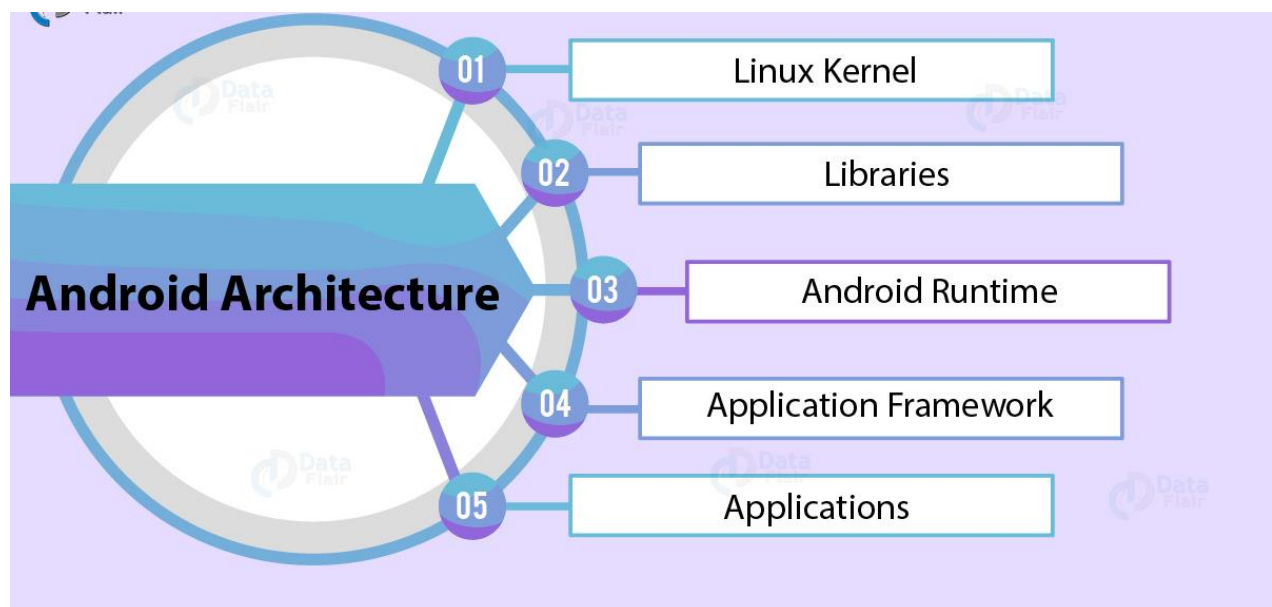


Fig 3.1 Architecture Design

3.2 Data flow diagram

A Data Flow Diagram (DFD) is a graphical representation of the “flow” of data through an information system, modeling its process aspects. Often it is a preliminary step used to create an overview of the system that can later be elaborated. DFDs can also be used for the visualization of data processing (structured design) and show what kind of information will be input to and output from the system, where the data will come from and go to, and where the data will be stored. It does not show information about the timing of processes or information about whether processes will operate in sequence or in parallel.

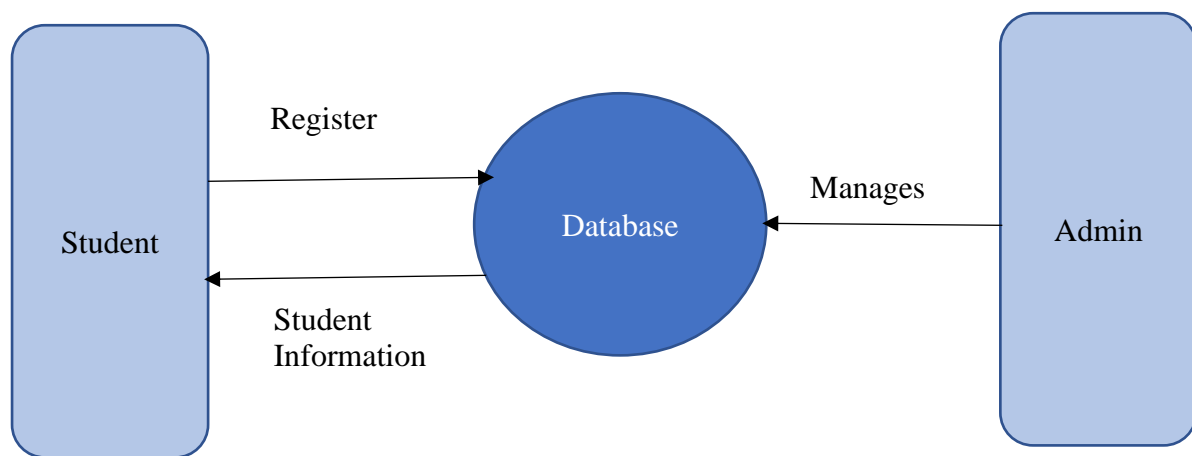


Fig 3.2.1: DFD Level 1 of Simple Database Application.

A level 1 DFD notates each of the main sub-processes that together form the complete system. We can think of a level 1 DFD as an “exploded view” of the context diagram.

The diagrams where the whole system is represented as a single process. The single process node from the context diagram is broken down into sub processes. As these processes are added, the diagram will need additional data flows and data stores to link them together.

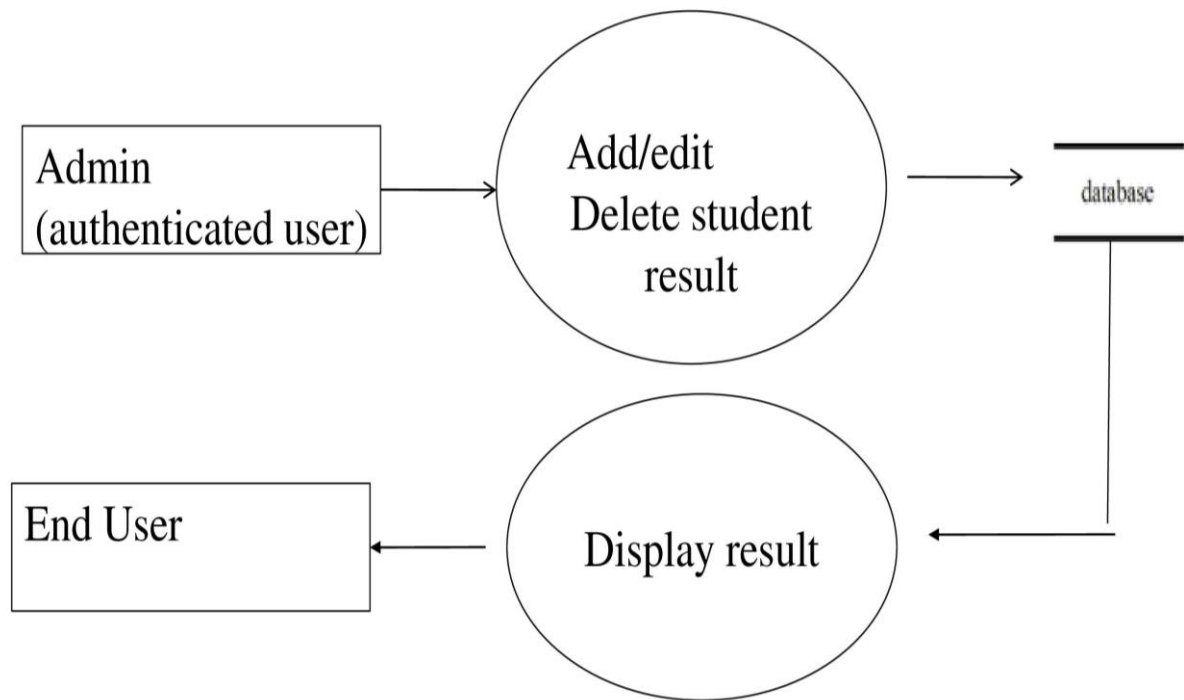


Fig 3.2.2: DFD Level 2 of Simple database application

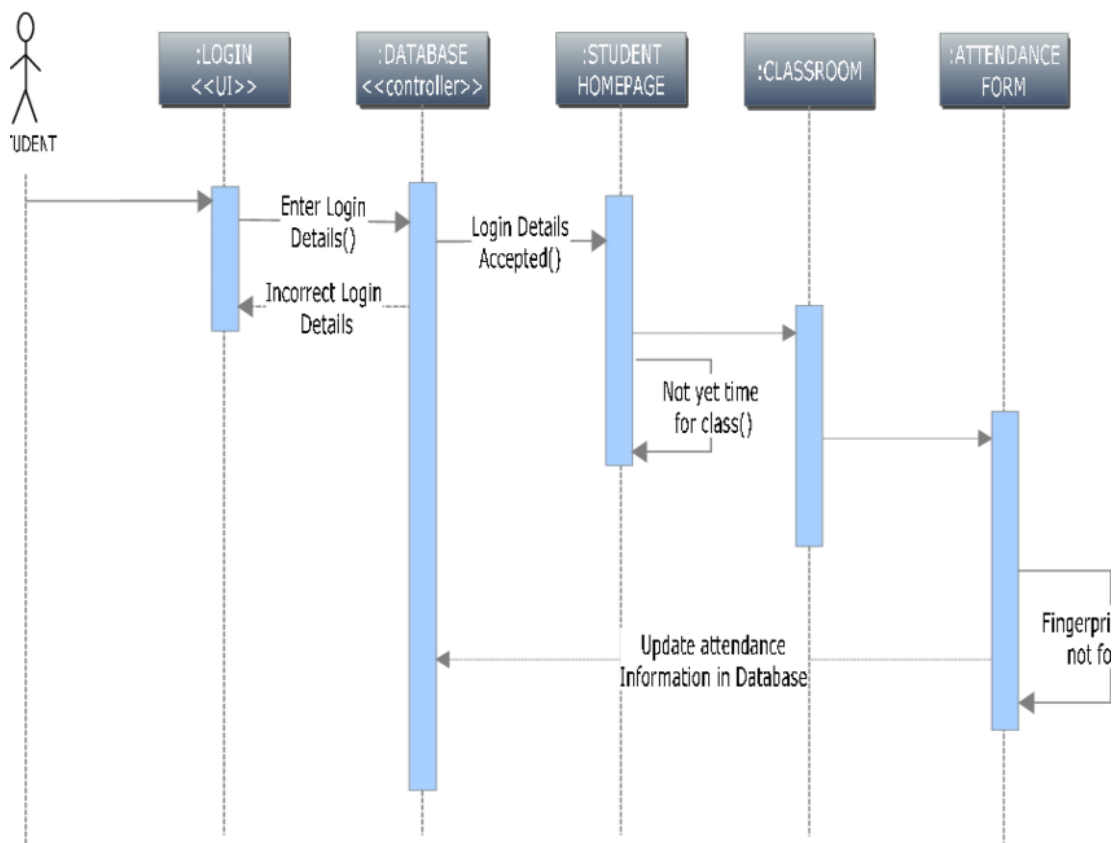
The level 2 DFD expands the processes in the level 1 diagram. The level two data flow diagram (DFD) template can map out information flow, visualize an entire system, and be shared with your stakeholders.

A DFD that represents a decomposed level 1 DFD process is called a level 2 DFD. There can be a level 2 DFD for each process that appears in the level 1 DFD. The every data-flow into and out of the parent process must appear as part of the child DFD.

3.3 Sequence diagram

A sequence diagram shows object interactions arranged in time sequence. It depicts the objects involved in the scenario and the sequence of messages exchanged between the objects needed to carry out the functionality of the scenario. Sequence diagrams are typically associated with use case realizations in the logical view of the system under development. Sequence diagrams are sometimes called event diagrams or event scenarios.

A sequence diagram shows, as parallel vertical lines (*lifelines*), different processes or objects that live simultaneously, and, as horizontal arrows, the messages exchanged between them, in the order in which they occur.



3.3 Sequence diagram of Simple database application

3.4 ER diagram

An entity-relationship model is usually the result of systematic analysis to define and describe what is important to process in an area of a business.

An E-R model does not define the business processes; it only presents a business data schema in a graphical form. It is usually drawn in a graphical form as boxes (entities) that are connected by lines (relationships) which express the associations and dependencies between entities.

Entities maybe characterized not only by relationships, but also by additional properties (attributes), which include identifiers called primary keys. Diagrams created to represent attributes as well as entities and relationships may be called entity-attribute-relationship diagrams rather than entity-relationship diagrams.

An Entity-Relationship Diagram (ERD) is a visual presentation of entities and relationships. That type of diagrams is often used in the semi-structured or unstructured data in databases and information systems. At first glance ERD is similar to a flows

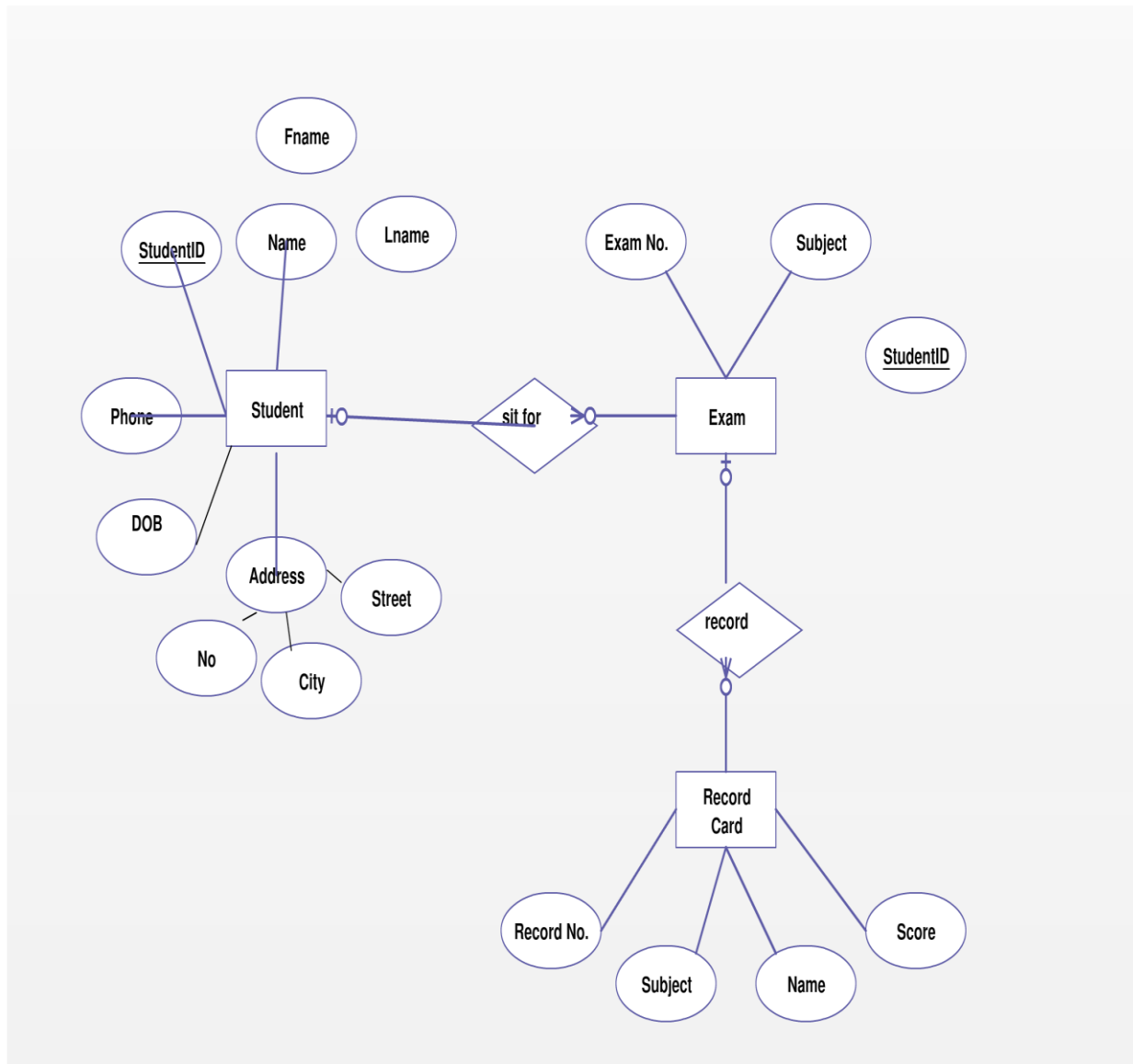


Fig3.4: ER diagram of Simple database application

3.5 Schema Diagram

The term schema refers to the organization of data as a blueprint of how the database is constructed (divided into database tables in case of relational databases). The formal definition of a database schema is a set of formulas (sentences) called integrity constraint is imposed on a database.

A relational schema shows references among fields in the database. When a primary key is referenced in another table in the database, it is called a foreign key. This is denoted by an arrow with the head pointing at the referenced key attribute.

A schema diagram helps organize values in the database. It also gives an idea of what order the tables should be created in. the following diagram shows the schema for the database.

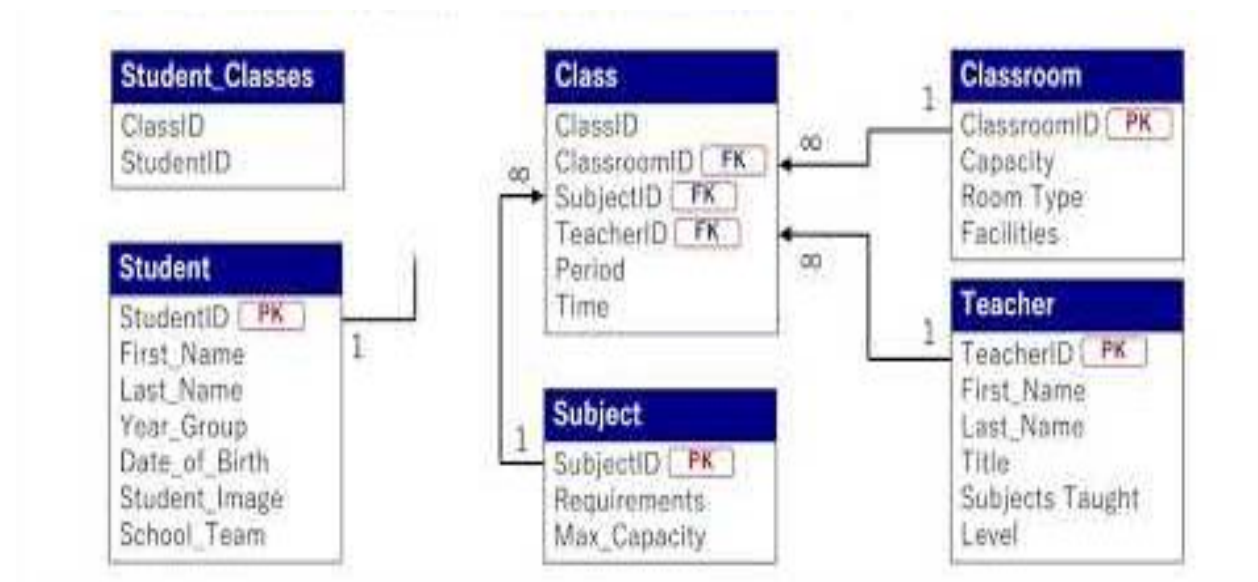


Fig 3.5: Schema Diagram of Simple Database diagram

Chapter 4

Implementation

4.1 Modules of the project

Modules provide a container for your app's source code, resource files, and app level settings, such as the module-level build file and Android manifest file. Each module can be independently built, tested, and debugged. Android Studio uses modules to make it easy to add new devices to your project.

Android Studio offers build automation, dependency management, and customizable build configurations. You can configure your project to include local and hosted libraries, and define build variants that include different code and resources, and apply different code shrinking and app signing configurations.

In the application “Simple database application ” we would classify to 3 modules

- Adding student name, USN
- Update student name, USN
- Delete student name, USN

4.2 Libraries used for the project

An Android library is structurally the same as an Android app module. It can include everything needed to build an app, including source code, resource files, and an Android manifest. ... AAR files can contain Java libraries for use by the app module's Java code.

- `import android.os.Bundle;`
- `import android.widget.*;`
- `import android.view.*;`
- `import java.util.ArrayList;`

import android.os.Bundle; - Android Bundle is used to pass data between activities. The values that are to be passed are mapped to String keys which are later used in the next activity to retrieve the values. Following are the major types that are passed/retrieved to/from a Bundle. PutInt (String key, int value), getInt (String key, int value).

An Android App Bundle is a publishing format that includes all your app's compiled code and resources, and defers APK generation and signing to Google Play. ... You no longer have to build, sign, and manage multiple APKs to optimize support for different devices, and users get smaller, more-optimized downloads.

import android.widget.*; - The widget package contains (mostly visual) UI elements to use on your Application screen. You can also design your own.

To create your own widget, extend View or a subclass. To use your widget in layout XML, there are two additional files for you to create. Here is a list of files you'll need to create to implement a custom widget:

- **Java implementation file** - This is the file that implements the behavior of the widget. If you can instantiate the object from layout XML, you will also have to code a constructor that retrieves all the attribute values from the layout XML file.
- **XML definition file** - An XML file in res/values/ that defines the XML element used to instantiate your widget, and the attributes that it supports. Other applications will use this element and attributes in their layout XML.
- **Layout XML [optional]**- An optional XML file inside res/layout/ that describes the layout of your widget. You could also do this in code in your Java file.
- ApiDemos sample application has an example of creating a custom layout XML tag, LabelView. See the following files that demonstrate implementing and using a custom widget:
 - **LabelView.java** - The implementation file
 - **res/values/attrs.xml** - Definition file
 - **res/layout/custom_view_1.xml** - Layout file

import android. View.*; - This class represents the basic building block for user interface components. A View occupies a rectangular area on the screen and is responsible for drawing and event handling. View is the base class for *widgets*, which are used to create interactive UI components (buttons, text fields, etc.). The View Group subclass is the base class for *layouts*, which are invisible containers that hold other Views (or other ViewGroups) and define their layout properties.

All of the views in a window are arranged in a single tree. You can add views either from code or by specifying a tree of views in one or more XML layout files. There are many specialized subclasses of views that act as controls or are capable of displaying text, images, or other content.

Once you have created a tree of views, there are typically a few types of common operations you may wish to perform:

- **Set properties:** for example setting the text of a Text View. The available properties and the methods that set them will vary among the different subclasses of views. Note that properties that are known at build time can be set in the XML layout files.
- **Set focus:** The framework will handle moving focus in response to user input. To force focus to a specific view, call `requestFocus ()`.
- **Set up listeners:** Views allow clients to set listeners that will be notified when something interesting happens to the view. For example, all views will let you set a listener to be notified when the view gains or loses focus. You can register such a listener using `setOnFocusChangeListener` (`android.view.View.OnFocusChangeListener`). Other view subclasses offer more specialized listeners. For example, a Button exposes a listener to notify clients when the button is clicked.
- **Set visibility:** You can hide or show views using `setVisibility (int)`.

import java.util.ArrayList; - The ArrayList class is a resizable array, which can be found in the java.util package. The difference between a built-in array and an ArrayList in Java is that the size of an array cannot be modified (if you want to add or remove elements to/from an array, you have to create a new one).

An ArrayList is a dynamic data structure, meaning items can be added and removed from the list. To set up an ArrayList, you first have to import the package from the java.

The `java.util` package provides Java collections framework classes, internationalization support classes, a service loader, properties, random number generation, string parsing and scanning classes, base64 encoding and decoding, a bit array, and several miscellaneous utility classes. This package also contains legacy collection classes and legacy date and time classes.

Package `java.util`

Package `java.util` contains the collections framework, legacy collection classes, event model, date and time facilities, internationalization, and miscellaneous utility classes (a string tokenizer, a random-number generator, and a bit array).

4.3 Algorithm of the project

```
package com.example.example;

import androidx.appcompat.app.AppCompatActivity;
import androidx.annotation.Nullable;
import androidx.appcompat.app.AppCompatActivity;
import android.content.ContentValues;
import android.content.Context;
import android.database.Cursor;
import android.database.sqlite.SQLiteDatabase;
import android.database.sqlite.SQLiteOpenHelper;
import android.os.Bundle;
import java.util.ArrayList;

public class DatabaseAccessActivity extends SQLiteOpenHelper {

    public static final String db = "studentdb";
    public static final String tab = "studenttab";
    public static final String col1 = "regno";
    public static final String col2 = "name";
    public ArrayList studdata = new ArrayList();
```

```
public DatabaseAccessActivity(@Nullable Context context) {
    super(context, "studentdb", null, 1);
}

@Override

public void onCreate(SQLiteDatabase db) {
    String query = "create table " + tab + "(" + col1 + " text, " + col2 + " text)";
    db.execSQL(query);
}

@Override

public void onUpgrade(SQLiteDatabase db, int i, int i1) {
}

public void Insertstud(String regno, String name) {
    SQLiteDatabase db = getWritableDatabase();
    ContentValues cv = new ContentValues();
    cv.put(col1, regno);
    cv.put(col2, name);
    db.insert(tab, null, cv);
    db.close();
}

public ArrayList DisplayRecord() {
    SQLiteDatabase db = getReadableDatabase();
    Cursor cu = db.rawQuery("Select * from " + tab, null);
    while (cu.moveToNext()) {
        studdata.add(cu.getString(0) + "\t\t" + cu.getString(1));
    }
    return studdata;
}

public void DeleteRecord(String regno) {
    SQLiteDatabase db = getWritableDatabase();
    // db.execSQL("delete from "+tab+" where "+col1+" = "+regno);
```

```
// db.close();

db.delete(tab, col1 + " = ?", new String[]{regno});
}

public void UpdateRecord(String regno, String name) {
    SQLiteDatabase db = getWritableDatabase();

    db.execSQL("update " + tab + " SET " + col2 + " = " + "\"" + name + "\"" + " where " + col1 +
" = " + "\"" + regno + "\"");

    db.close();
}
```


Chapter 5

RESULTS

5.1 INSERT

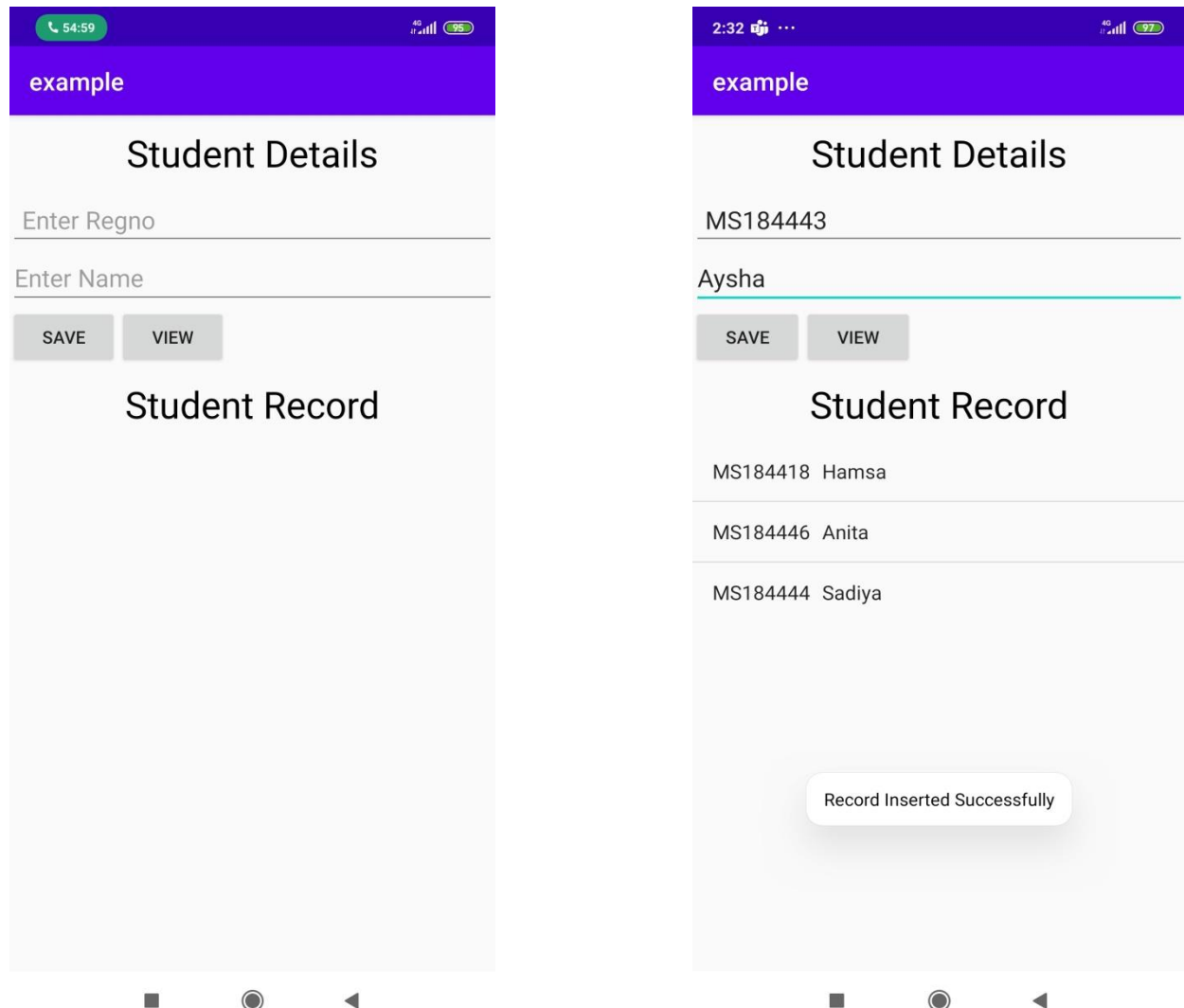
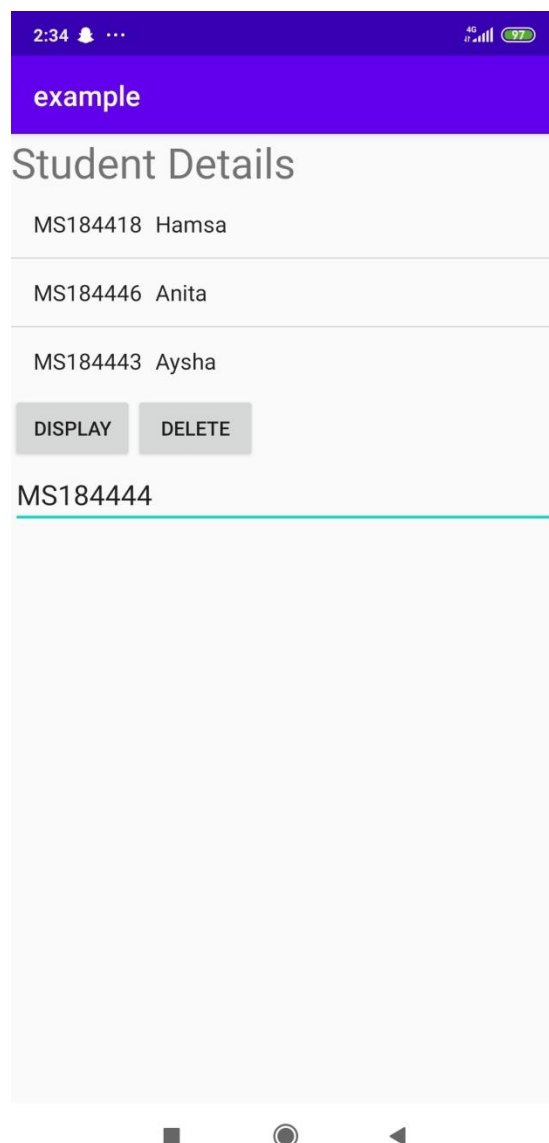
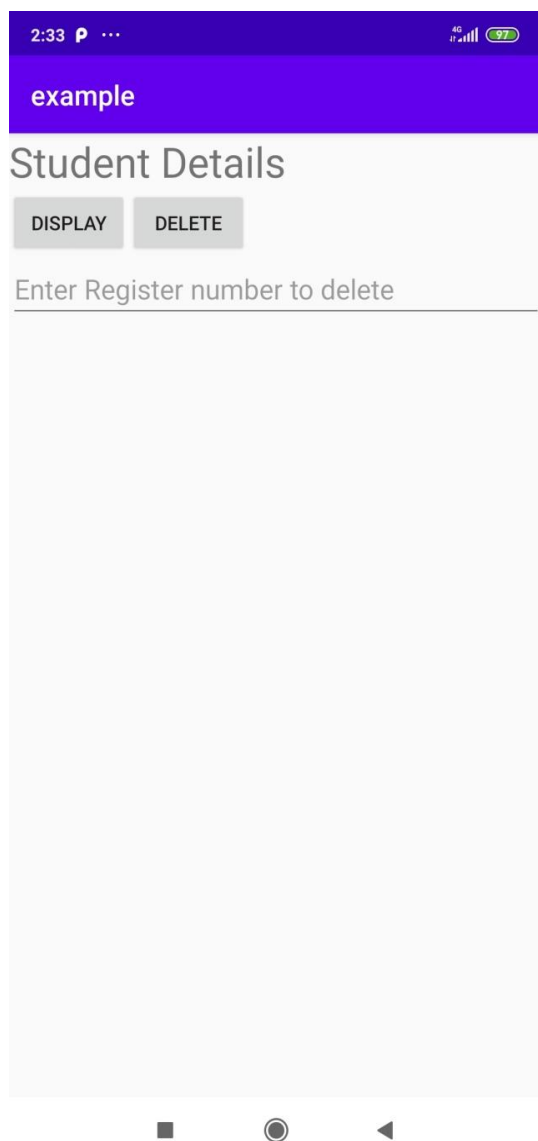


Fig 5.1 Insert Screen

5.2 DELETE



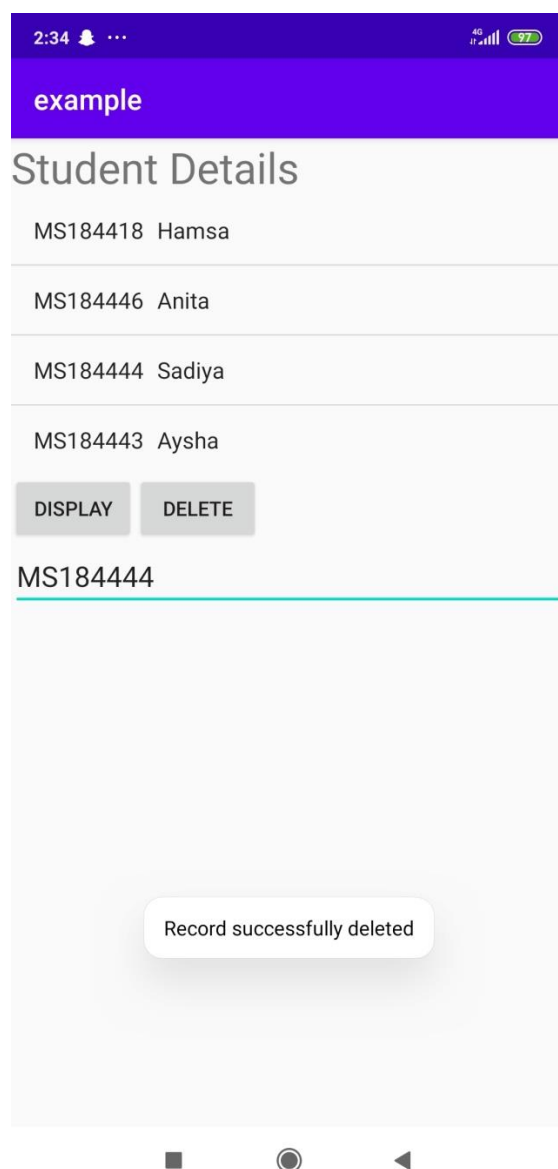
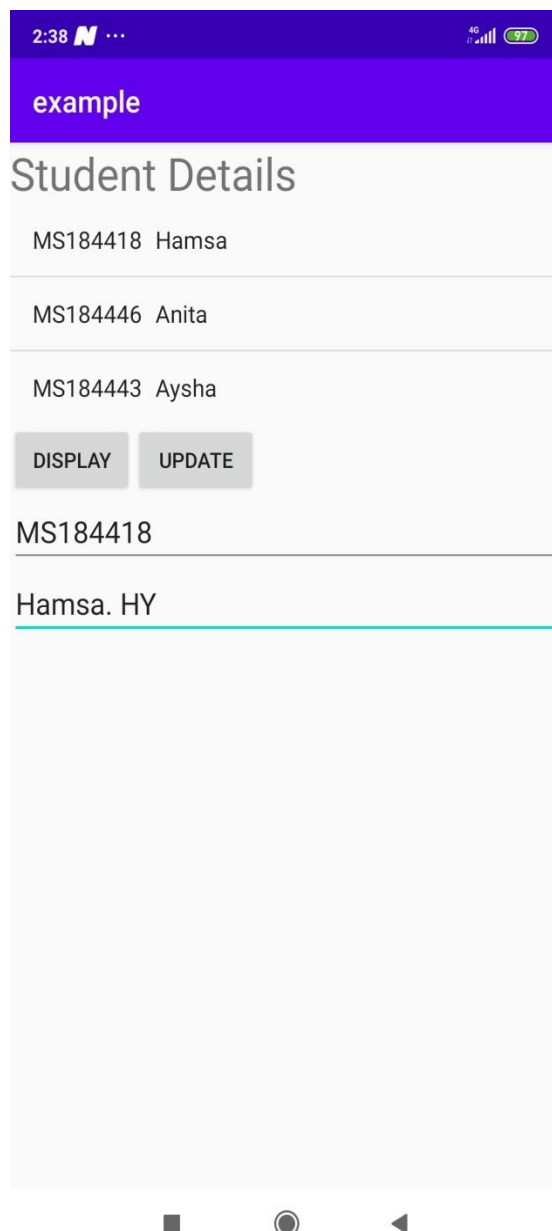
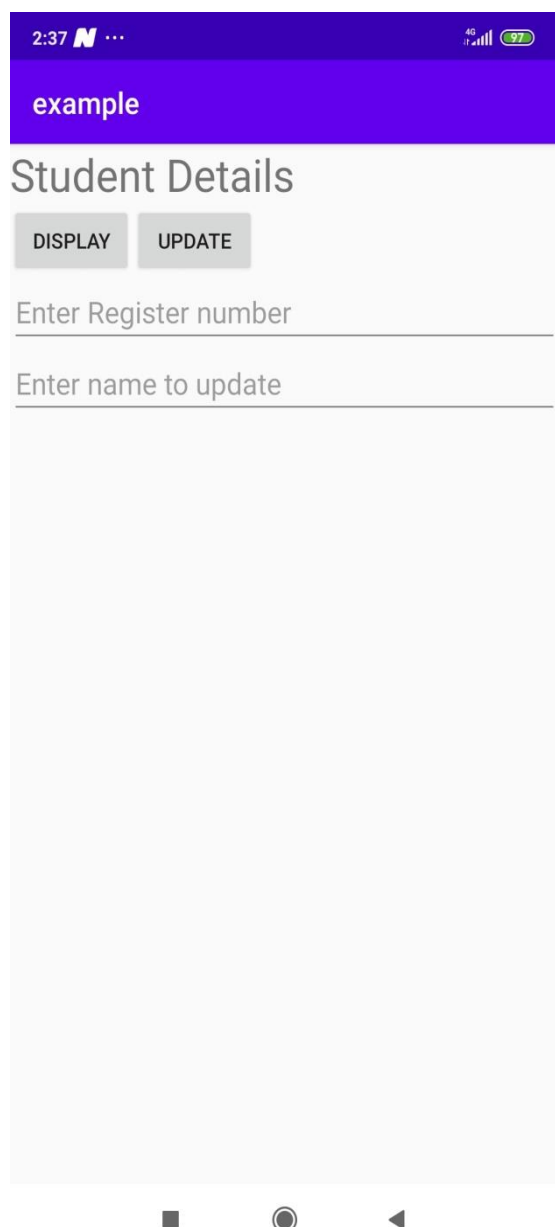


Fig 5.2 Delete Screen

5.3 UPDATE



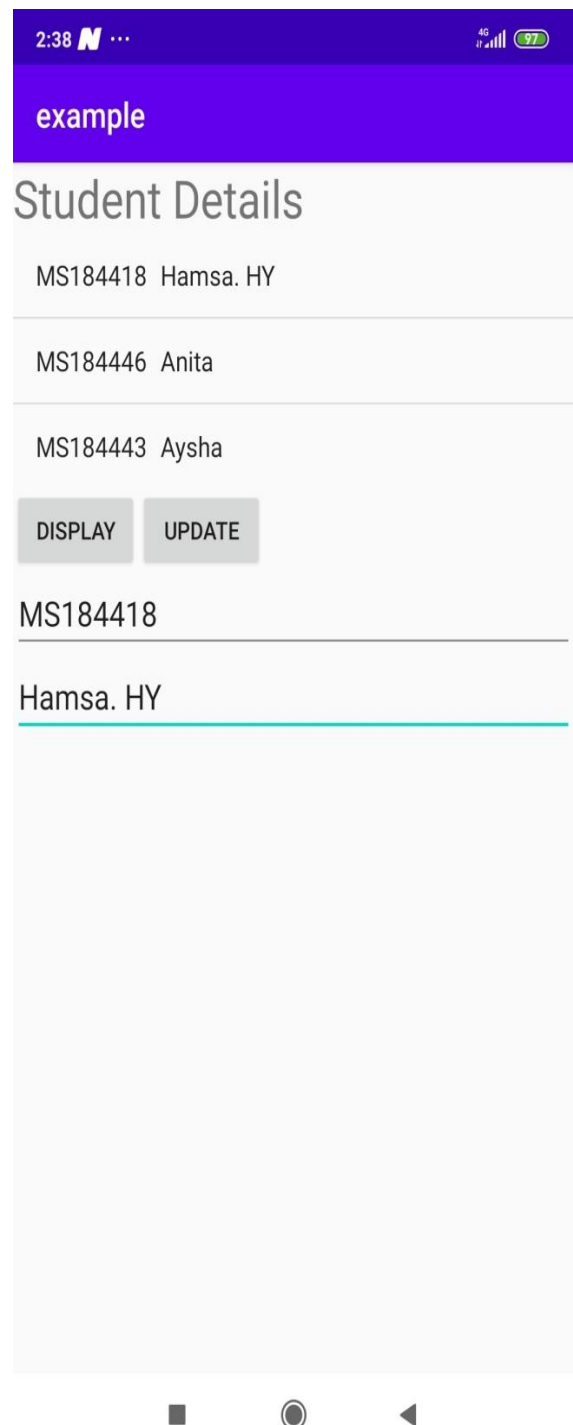
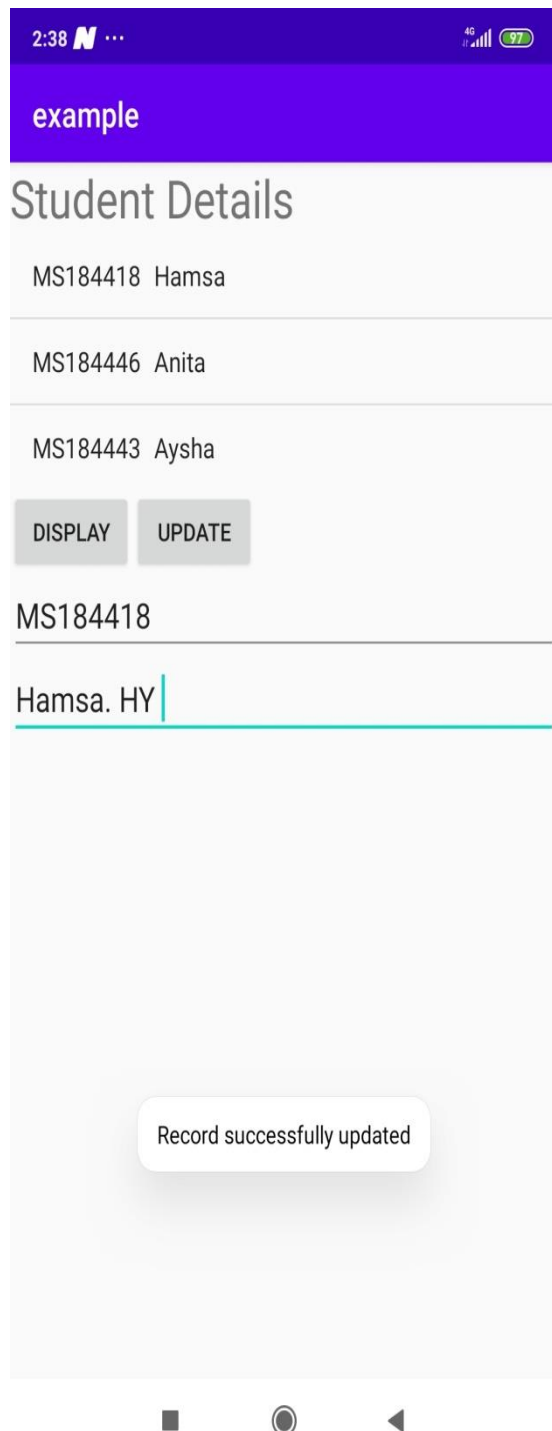


Fig 5.3 Update Section

CONCLUSION

The application is only a humble venture to satisfy the needs to manage their application work. Several user-friendly coding has also been adopted. The objective of software application planning is to provide framework that enables the manager to make reasonable estimates made within a limited time frame at the beginning of the application software project and should be updated regularly as the project progresses. It has described the requirement specifications of the system and the actions that can be done on these things. We have included features and operations in detail, including screen layouts. We have also designed user interface and security issues related to the system. Finally the application system is implemented.

In a nutshell, it can be summarized that the future scope of the project circles around maintaining information regarding:

1. We can add printer in future
2. We can give more advance application software System including more facilities.
3. We will host the platform on online servers to make it accessible worldwide.
4. Implement the backup of code base and database on regular basis on different servers.

The above-mentioned points are the enhancements which can be done to increase the applicability and usage of the application project. We have left all the options open so that if there is any future requirement in the system application by the user for the enhancement of the system application then it is possible to implement them. We hope that the project will serve its purpose for which it is developed by underlying success of process.

BIBLIOGRAPHY

1. web.stanford.edu. Android App Development by Stanford. ...
2. coursera.org. Android App Development Specialization. ...
3. udacity.com. Android Development for Beginners. ...
4. youtube.com. Android App Development for Beginners. ...
5. Code Abbey. The next site that I am going to share with you guys is Code Abbey. ...
6. Code Academy. One of the most popular websites, which I used to learn JavaScript, is Code Academy. ...
7. <https://projectworlds.in/android-projects-with-source-code/>
8. <https://projectsgeek.com/android-projects-with-source>
9. code<https://www.freeprojectz.com/android-projects>
10. <https://freeprojectsforall.com/android-mini-projects/>