**LOCATION FINDER APP**

A Mini Project Report Submitted by

**ASHIKA S SHETTY (4NM17CS032)**

**ANUSHA JAGANNATH SOORENJI (4NM17CS027)**

**UNDER THE GUIDANCE OF**

**Ms.SHABARI SHEDTHI B**

Assistant Professor Gd II

Department of Computer Science and Engineering

# in partial fulfilment of the requirements for the award of the Degree of

Bachelor of Engineering in

Computer Science & Engineering

from

# Visvesvaraya Technological University, Belagavi



DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

# N.M.A.M. INSTITUTE OF TECHNOLOGY

(An Autonomous Institution under VTU, Belgaum) (AICTE approved, NBA Accredited, ISO 9001:2008 Certified) NITTE -574 110, Udupi District, KARNATAKA.

**April 2020**



(ISO 9001:2015 Certified), Accredited with ‘A’ Grade by NAAC

: 08258 - 281039 – 281263, Fax: 08258 – 281265

## Department of Computer Science and Engineering

B.E. CSE Program Accredited by NBA, New Delhi from 1-7-2018 to 30-6-2021

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

CERTIFICATE

“LOCATION FINDER”

is a bonafide work carried out by

ASHIKA S SHETTY (4NM17CS032)

ANUSHA JAGANNATH SOORENJI (4NM17CS027)

in partial fulfillment of the requirements for the award of Bachelor of Engineering Degree in Computer Science and Engineering prescribed by Visvesvaraya Technological University, Belagavi during the year 2019-2020.

It is certified that all corrections/suggestions indicated for Internal Assessment have been incorporated in the report.

The Mini project report has been approved as it satisfies the academic requirements in respect of the project work prescribed for the Bachelor of Engineering Degree.

Signature of Guide Signature of HOD

## ACKNOWLEDGEMENT

We believe that our project will be complete only after we thank the people who have contributed to make this project successful.

First and foremost, we express our deep sense of gratitude and indebtedness to our guide **Ms Shabari Shedthi B**, Assistant Professor Gd. II, Department of Computer Science and Engineering, for her inspiring guidance, constant encouragement, support and suggestions for improvement during the course of our project.

We sincerely thank **Dr. K.R. Udaya Kumar Reddy**, Head of Department of Computer Science and Engineering, Nitte Mahalinga Adyantaya Memorial Institute of Technology, Nitte.

Our sincere thanks to our beloved principal, **Dr. Niranjan N. Chiplunkar** for giving us an opportunity to carry out our project work at our college and providing us with all the needed facilities.

We also thank all those who have supported us throughout the entire duration of our project.

Finally, we thank the staff members of the Department of Computer Science and Engineering and all our friends for their honest opinions and suggestions throughout the course of our project.

**ABSTRACT**

NMAMIT is a college with vast campus. Freshers find difficulty in locating their classrooms and labs. So, location finder app is useful in solving their problem.Location finder is an app where students can directly access their classroom address and lab address.Students are also given provision to locate staffrooms of a particular faculty.So it is very convenient for the students. One of the most important feature of this app is that it saves lot of time and effort in locating any rooms in the college during exams.Also it is simple to use.

## 

Location Finder

**Table Of Contents:**

**Page no.**

1. **INTRODUCTION 6**

1. **LITERATURE SURVEY 7**

1. **SYSTEM REQUIREMENT AND SPECIFICATION 8-11**

1. **SYSTEM DESIGN 12-15**

1. **IMPLEMENTATION 16-20**

1. **SCREENSHOTS 21-25**

1. **CONCLUSION AND FUTURE WORK 26**
2. **REFERENCES 27**

**CHAPTER 1**

**INTRODUCTION**

### 1.1 Scope

Location finder is a flexible, easy to use, secure designed to benefit the student. It is used to access the classrooms ,labs and even faculty detailes easily and manage them efficiently.

### 1.2 Importance

NMAMIT is a college with vast campus surrounded with greenary. It is a tedious task to find any classrooms or labs for freshers and for the people who visit our campus for certain occasion. Also students find difficulty in locating staffrooms. The application is implemented on android platform which is linked to the SQLite database.

**Objective:**

In real world people find difficulty in locating places.This problem is solved with the help of google maps.Also in college students find it difficult to locate their classrooms during exams if the campus is vast. Students even find it difficult to approach concerned faculty.

**CHAPTER 2**

**LITERATURE SURVEY**

**2.1 Technical Background:**

In this app the information regarding classrooms,labs,department and faculty information are stored in the backend i.e., in SQLite database. GUI of the application enables to access and modify data efficiently.

**2.2 Existing system:**

* Classrooms and lab detailes are made available only on that particular floor
* Faculty detailes are put up on the college website which doesn’t guarantee that all the student will be aware of this.

### 2.3 Proposed system

In this App all the existing traditional system can be digitalized and can be made available at the tip of the finger. All the above existing system is upgraded and manual work can be reduced . There will be one administrator (lecturer) who will manage the data at the back end.Information are displayed using alert box. Lecturer/student has the permission to post related classrooms /lab detailes, which helps student to access the data and will be available online through mobile application.

* Students can access classrooms,faculty and department detailes anywhere through a mobile application.

**CHAPTER 2**

**SYSTEM REQUIREMENT AND SPECIFICATION**

### 3.1 Introduction

Requirements are during early stages of a system development as a specification of what should be implemented or as a constraint of some kind of on the system. They may be a user level facility description, a detailed specification of expected system behavior, a general system property, a specific constraint on the system, and information on how to carry out some computation or a constraint on the development of the system. The end product of the requirement analysis phase is a requirement specification. The requirement specification is a reconstruction of the result of this analysis phase. Its purpose is to communicate this result to others. System requirements are more detailed descriptions of the user requirements. They may serve as the basis for a contract to the implementation of the system and should therefore be a complete and consistent specification of the whole system. They are used by software engineers as the starting point of system design. In principle, the system requirements should state what the system should do and not how it should be implemented. However, at the level of detail required to specify the system completely, it is virtually impossible to exclude all design information.

Natural language is often used to write system requirements specifications. Further problems with natural language can arise when it is used for more detailed specification:

1. Natural language understanding relies on the specification of the readers and writers using the same words for the same concept. This leads to misunderstandings because of the ambiguity of the natural language.
2. A natural language requirements specification is over-flexible. You can say the same thing in completely different ways. It is up to the reader to find out when requirements are same and when they are distinct.
3. There is no easy way to modularize natural language requirements. It may be difficult to find all the related requirements. To discover the consequence of a change, you may have to look at every requirement rather than just a group of related requirements.

3.2 Functional Requirements

The functional requirements are the statement of services the system should provide, how system reacts to particular inputs and how system should behave in particular situation. It describes the functionality that the system provides.

Our app requires:

1. Active internet connection.
2. A SQLite database to store the data

**3.3 User Requirements**

Both student and lecturer requires active internet connection to use the app.

### 3.4 Software Requirements

1. Operating System: Windows 7/8/10 (32-bit or 64-bit)
2. Android SDK III. Android Studio IV. Sqlite database

**3.4.1 Android SDK**

I. The Android SDK provides you the API libraries and developer tools necessary to build, test, and debug apps for Android. The ADT bundle includes the essential Android SDK components and a version of the Eclipse IDE with built-in Android Developer Tools to streamline the Android app development. ADT bundle consists of following components for developing the application II. Eclipse ADT plugin.

1. Android SDK Tools
2. Android Platform-tools
3. The latest Android platform
4. The latest Android system image for the emulator

#### 3.4.2 Android Studo

**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, macOS and Linux based operating systems. It is a replacement for the Eclipse Android Development Tools (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. The current stable version is 3.3, which was released in January 2019.

#### 3.4.3 SQlite

**SQLite**

**SQLite**is a self-contained, high-reliability, embedded, full-featured, public-domain, SQL database engine. It is the most used database engine in the world. It is an in-process library and its code is publicly available. It is free for use for any purpose, commercial or private. It is basically an embedded SQL database engine. Ordinary disk files can be easily read and write by SQLite because it does not have any separate server like SQL. The SQLite database file format is cross-platform so that anyone can easily copy a database between 32-bit and 64-bit systems. Due to all these features, it is a popular choice as an Application File Format.

**History:**  
 It was designed by D. Richard Hipp for the purpose of no administration required for operating a program. in August 2000. As it is very lightweight compared to others like MySql and Oracle, it is called SQLite. Different versions of SQLite are released since 2000.

### Why SQLite?

### • SQLite does not require a separate server process or system to operate (serverless).

### • SQLite comes with zero-configuration, which means no setup or administration needed.

### • A complete SQLite database is stored in a single cross-platform disk file.

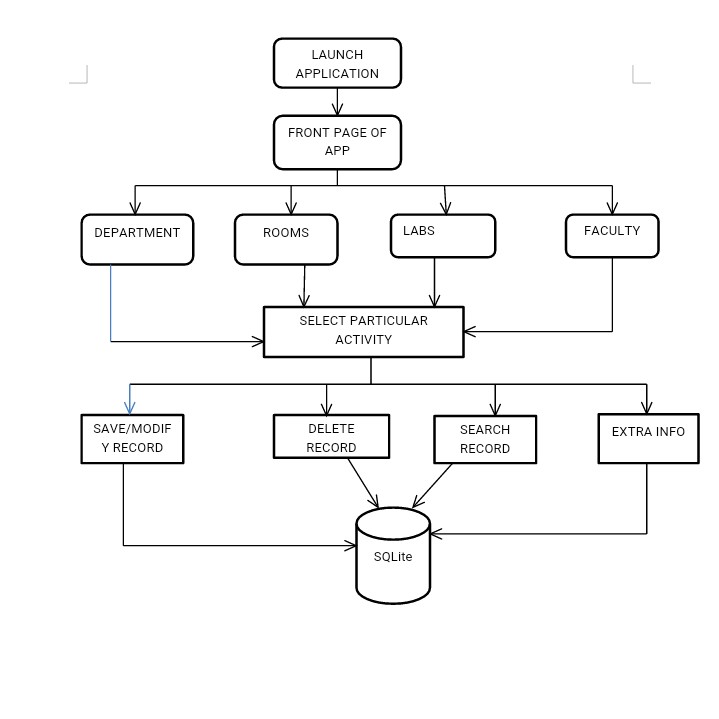
### 3.5 HARDWARE REQUIREMENTS

1. Intel i Series Processor.
2. Minimum 4 GB RAM (8GB recommended).
3. 5GB free disk space

**CHAPTER 4**

**SYSTEM DESIGN**

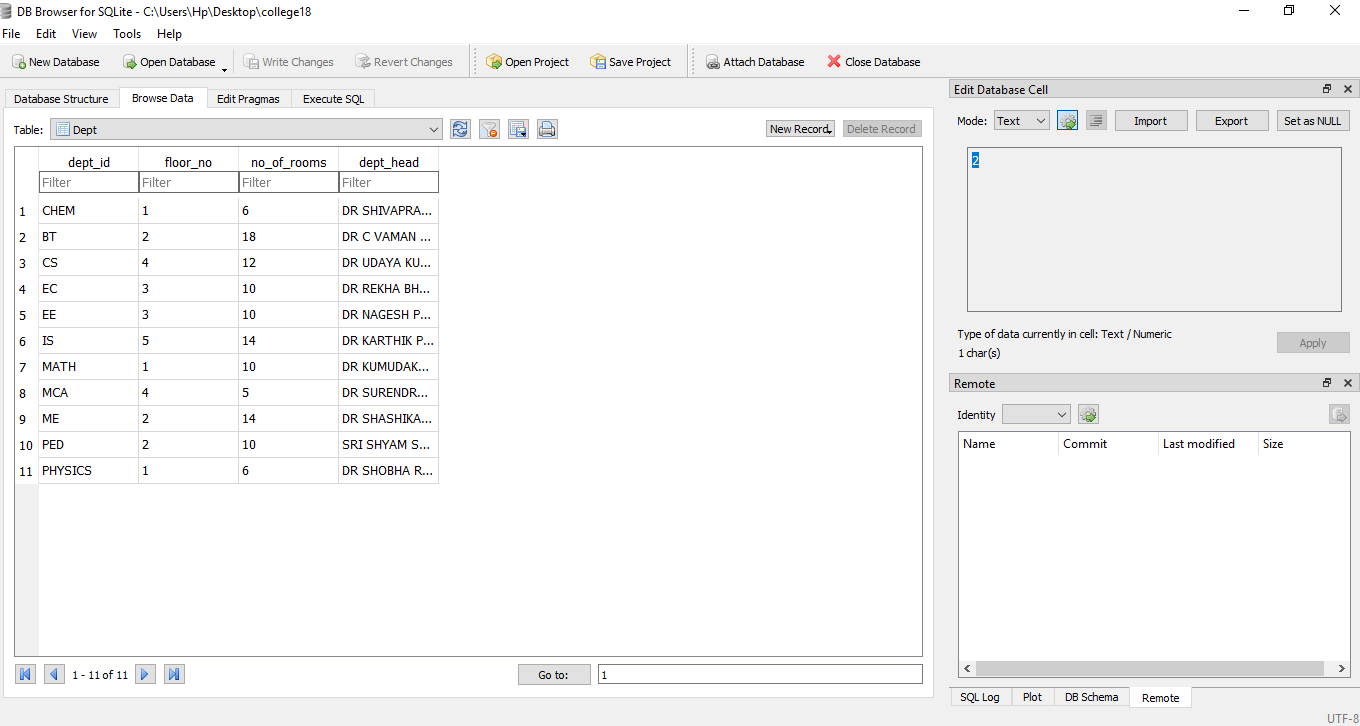
**4.2 Dataflow Diagram**

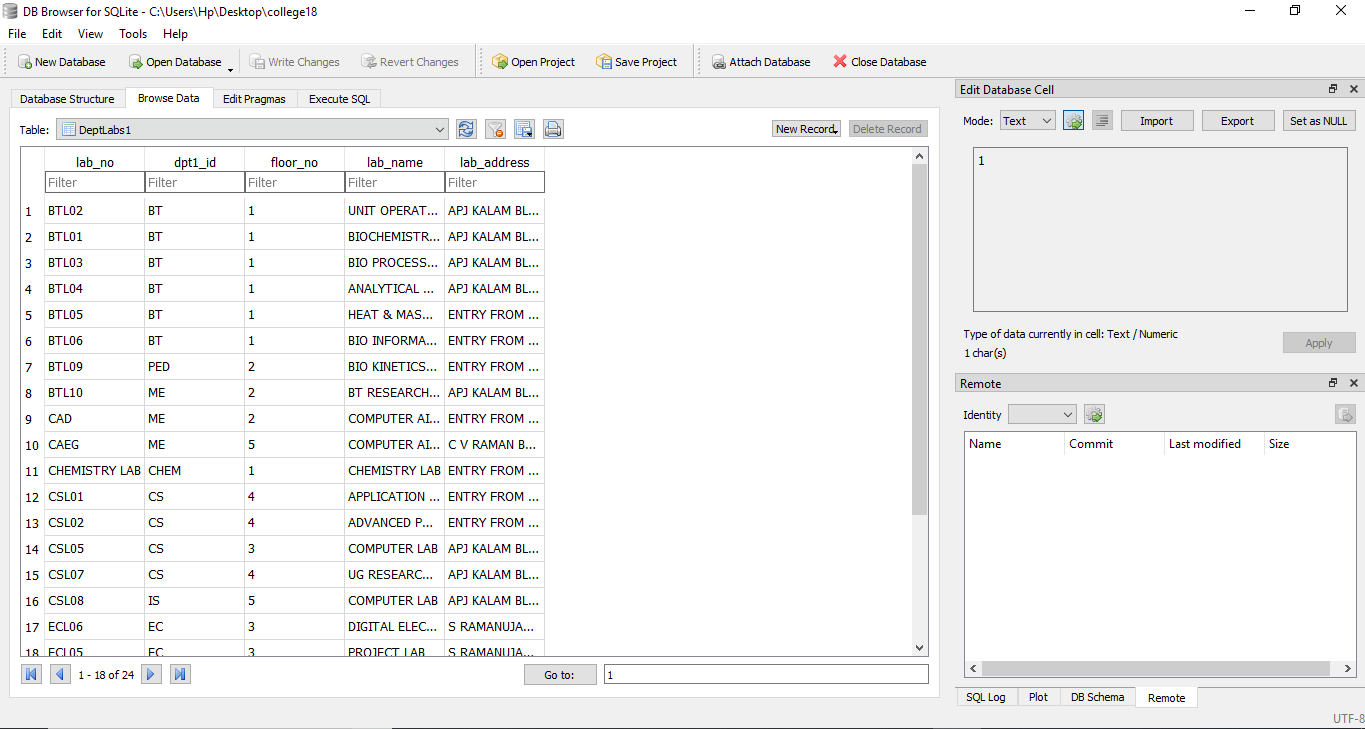


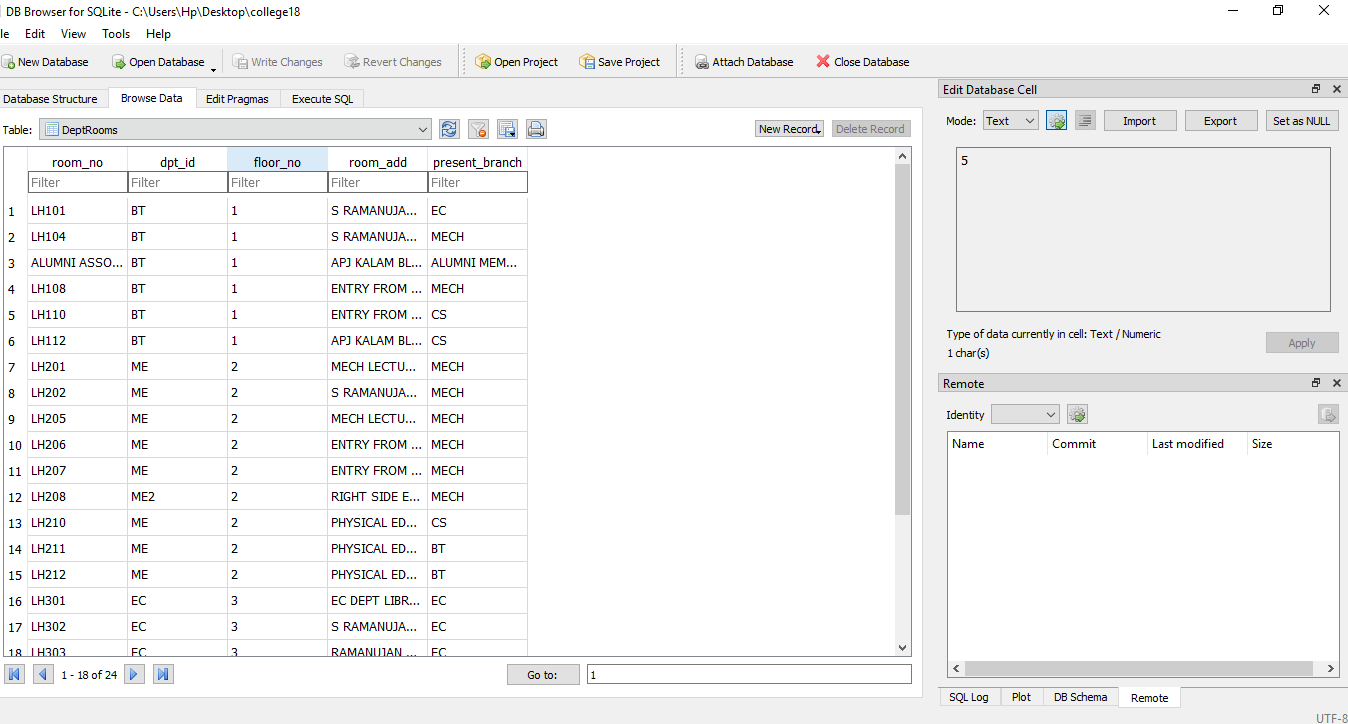
**4.4 Database Design**

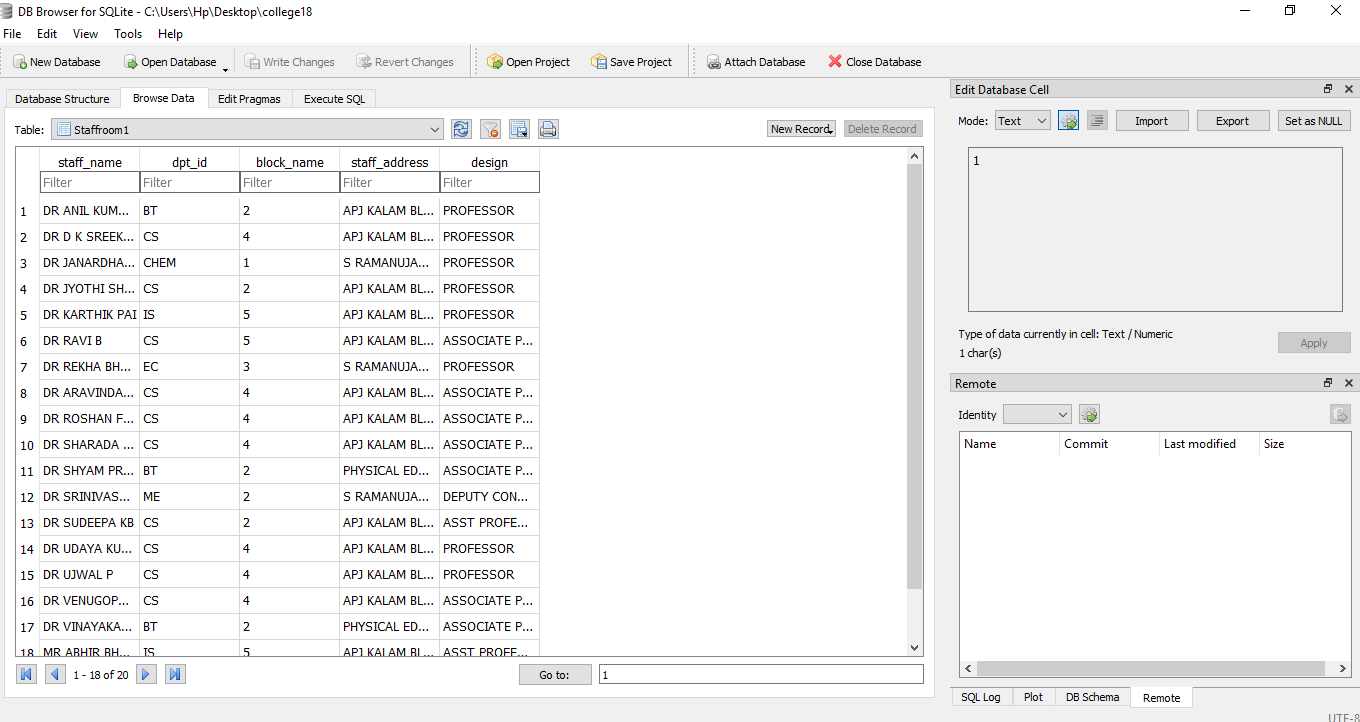
The database is designed using Google Firebase Console in which data is stored in popular data structure known as JSON tree (JavaScript Object Notation). Every time when the data transfer happens from client end, the information given to the UI is converted into JSON tree structure which is efficient and faster way to retrieve and store data.

### A snapshot of the data in the SQLite database

**Department table**

**Dept Labs table**

**Dept Rooms Table**

**Faculty table**

**CHAPTER 5**

**IMPLEMENTATION**

We are here designing location finder app to automate and manage all the activities. The student remain updated with all the events.

### 5.1 SOLUTION APPROACH/METHODOLOGY

We are here using xml and java for the front end and SQLite for the backend as a server.

**5.1.1 SQLite**

SQLite is considered as web application platform. It helps query for inserting, updating, deleting or adding data to it. It is the backend of a system that is used as a database for storing data. SQLite is real-time database feature is very easy to use.

**5.1.2 STORAGE**

The files like images, audio, video etc can be stored in the app. The data stored is highly secured and is robust in nature means it resumes from the last point if any network error occurs. The steps below are to be followed to use storage feature in Android application: added to the application, create instance of

#### 5.1.3 SQLITE AND ANDROID APP

#### 

An Android application has been developed for the demonstration of SQLite. In this app images along with strings are loaded to SQLite and retrieved from SQLite similar to Instagram. For the development of an Android app to demonstrate the use of SQLite prototyping model has been followed.

**Steps for connecting App to Firebase:**

**Step1**: Create a database helper class and include all of its methods

openDatabase(String path, SQLiteDatabase.CursorFactory factory, int flags, DatabaseErrorHandler errorHandler)

This method only opens the existing database with the appropriate flag mode. The common flags mode could be OPEN\_READWRITE OPEN\_READONLY

**Step2**: In the main activitywe canuse execSQL method**.**execSQL(String sql, Object[] bindArgs)

This method not only insert data , but also used to update or modify already existing data in database using bind arguments.

**Step3:** Create a Cursor to display data from the table of the database.

**Step4**:Close the database **connectivity**.

**It’s a Realtime Database**

stem. SQLite engine is not a standalone process like other databases, you can link it statically or dynamically as per your requirement with your application. SQLite accesses its storage files directly.

.

**It’s File Storage**

SQLite Storage provides a simple way to save binary files — most often images.

#### 5.1.4 Java

There are several ways to create apps for Android devices, but the recommended method for most developers is to write native apps using Java and the Android SDK. Java for Android apps is both similar and quite different from other types of Java applications.

If you have experience with Java (or a similar language) then you’ll probably feel comfortable diving right into the code and learning how to use the Android SDK to make your app run. But if you’re new to programming or object-oriented languages then you’ll probably want to get familiar with the syntax of the Java language and how to accomplish basic programming tasks before learning how to use the Android SDK.

### 5.2 IMPLEMENTATION CODE

### 

Database creation**:**

@Override  
**public void** onCreate(SQLiteDatabase db) {  
 db.execSQL(**"create table "** + ***TABLE\_NAME1*** + **"(dept\_id TEXT primary key,floor\_no INTEGER,no\_of\_rooms INTEGER,dept\_head TEXT)"**);  
 db.execSQL(**"create table "**+***TABLE3***+**"(lab\_no TEXT primary key,dpt1\_id TEXT,floor\_no TEXT,lab\_name TEXT,lab\_address TEXT)"**);  
 db.execSQL(**"create table "**+***TABLE4***+**"(staff\_name TEXT primary key,dpt\_id TEXT,block\_name TEXT,staff\_address TEXT,design TEXT)"**);  
 db.execSQL(**"create table "** + ***TABLE2*** + **"(room\_no TEXT primary key,dpt\_id TEXT, floor\_no TEXT,room\_add TEXT,present\_branch TEXT)"**);  
  
  
}  
  
@Override  
**public void** onUpgrade(SQLiteDatabase db, **int** oldVersion, **int** newVersion) {  
 db.execSQL(**"DROP TABLE IF EXISTS "** + ***TABLE\_NAME1***);  
 *// onCreate(db);*}

**Database insertion:**

**public boolean** insertdata(String dept\_id, **int** floor\_no,**int** no\_of\_rooms, String dept\_head) {  
 SQLiteDatabase db = **this**.getWritableDatabase();  
 ContentValues contentValues = **new** ContentValues();  
 contentValues.put(***COL\_1***, dept\_id);  
 contentValues.put(***COL\_2***, floor\_no);  
 contentValues.put(***COL\_3***, no\_of\_rooms);  
 contentValues.put(***COL\_4***, dept\_head);  
 **long** result = db.insert(***TABLE\_NAME1***, **null**, contentValues);  
 **if** (result == -1)  
 **return false**;  
 **else  
 return true**;  
}

Fetching Data

**public** Cursor getAlldata(String s)  
{  
 SQLiteDatabase db=**this**.getWritableDatabase();  
 Cursor res1=db.rawQuery(**"select \* from Dept where dept\_id LIKE '%"**+s+**"%'"** ,**null**);  
 **return** res1;  
  
  
  
}

**Database Updation**

**public boolean** updatedata(String deptid,**int** floorno,**int** noofrooms,String depthead)  
{  
 SQLiteDatabase db=**this**.getWritableDatabase();  
 ContentValues contentValues=**new** ContentValues();  
 contentValues.put(***COL\_1***,deptid);  
 contentValues.put(***COL\_2***,floorno);  
 contentValues.put(***COL\_3***,noofrooms);  
 contentValues.put(***COL\_4***,depthead);  
 db.update(***TABLE\_NAME1***,contentValues,**"dept\_id=?"**,**new** String[]{deptid});  
 **return true**;  
  
  
}

**Deletion**

**public** Integer deletedata(String id){  
 SQLiteDatabase db=**this**.getWritableDatabase();  
 **return** db.delete(***TABLE\_NAME1***,**"dept\_id = ?"**,**new** String[] {id});  
}

Store Image

**public void** storeImage(ModelClass objectModelClass)  
{  
 **try**{  
 SQLiteDatabase db=**this**.getWritableDatabase();  
 Bitmap imageTostore=objectModelClass.getImage();  
 **objectByteArrayOutputStream**=**new** ByteArrayOutputStream();  
 imageTostore.compress(Bitmap.CompressFormat.***JPEG***,100,**objectByteArrayOutputStream**);  
 **imageToByte**=**objectByteArrayOutputStream**.toByteArray();  
 ContentValues contentValues=**new** ContentValues();  
 contentValues.put(**"image\_name"**,objectModelClass.getImageName());  
 contentValues.put(**"image"**,**imageToByte**);  
 **long** i=db.insert(**"imageinfo"**,**null**,contentValues);  
 **if**(i!=-1) {  
 Toast.*makeText*(**context**, **"Image added"**, Toast.***LENGTH\_SHORT***).show();  
 db.close();  
 }  
 **else** Toast.*makeText*(**context**,**"image not added"**,Toast.***LENGTH\_SHORT***).show();  
  
 } **catch** (Exception e) {  
 e.printStackTrace();  
 }  
}

**public** ArrayList<ModelClass> getAllImagesData()  
{  
 **try**{  
  
 SQLiteDatabase db=**this**.getReadableDatabase();  
 ArrayList<ModelClass> objectModelClassList=**new** ArrayList<>();  
 Cursor cursor=db.rawQuery(**"select \* from imageinfo"**,**null**);  
 **if**(cursor.getCount()!=0)  
 {  
 **while**(cursor.moveToNext())  
 {  
 String name=cursor.getString(0);  
 **byte**[] imageBytes=cursor.getBlob(1);  
 Bitmap objectBitmap= BitmapFactory.*decodeByteArray*(imageBytes,0,imageBytes.**length**);  
 objectModelClassList.add(**new** ModelClass(name,objectBitmap));  
 }  
 **return** objectModelClassList;  
 }  
 **else** Toast.*makeText*(**context**,**"no records"**,Toast.***LENGTH\_SHORT***).show();  
 **return null**;  
  
 } **catch** (Exception e) {  
  
 **return null**;  
 }  
  
}

**CHAPTER 7**

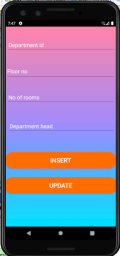
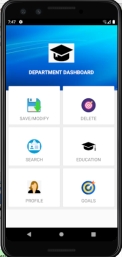
**SCREENSHOTS**

**Home Page**

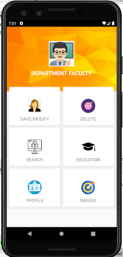
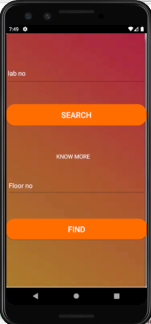
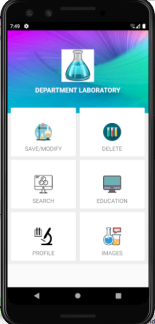


#### 

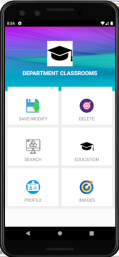
### 1.Dept Dashboard 2. Insert/update 3.delete



### 4.Lab Dashboard 5. Insert/update 6. Search



**7. Faculty Dashoard 8.Save image 9.Display image**



#### 10.Class Dashboard 11.Insert/Modify 12.Search

**CHAPTER 8**

**CONCLUSION AND FUTURE WORK**

**8.1 Results/conclusion:**

* Location finder app is an android application used to manage the various activities
* A platform for scheduling and notifying can prove to be most effective and reliable.
* As the application is designed in android platform, it can be installed in every faculty members’s and student’s cell phone.

##### 8.2 Future Works

In the future, we may extend this project by adding extra features to our android app like Notifications, Chat and Class groups.

##### 8.3 References

* <https://www.geeksforgeeks.org/introduction-to-sqlite/>
* <https://www.tutorialspoint.com/android/index.htm>