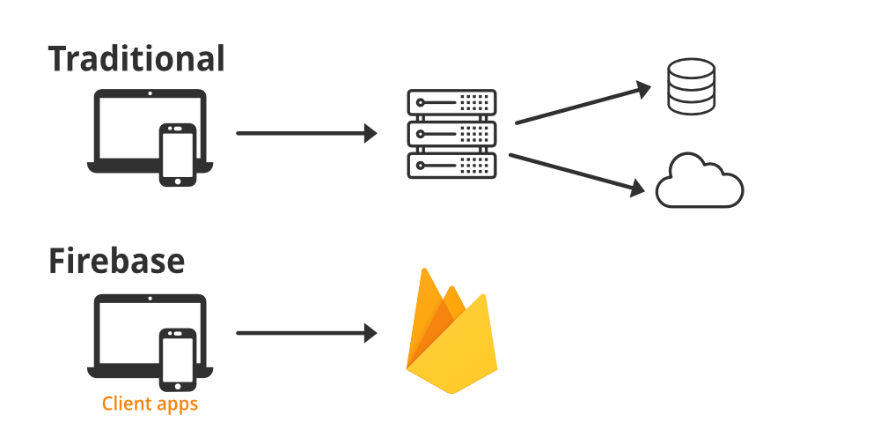
**CHAPTER 1:**

**INTRODUCTION**

**SYNOPSIS:**

The “GPS Tracking for Bike Rental” project is developed using Android Studio, coded in Java, & utilizes Firebase as the backend database. This project aims to create a robust and user-friendly mobile application for bike rental services, incorporating GPS tracking to enhance the overall user experience, ensure bike security, and streamline operational processes.

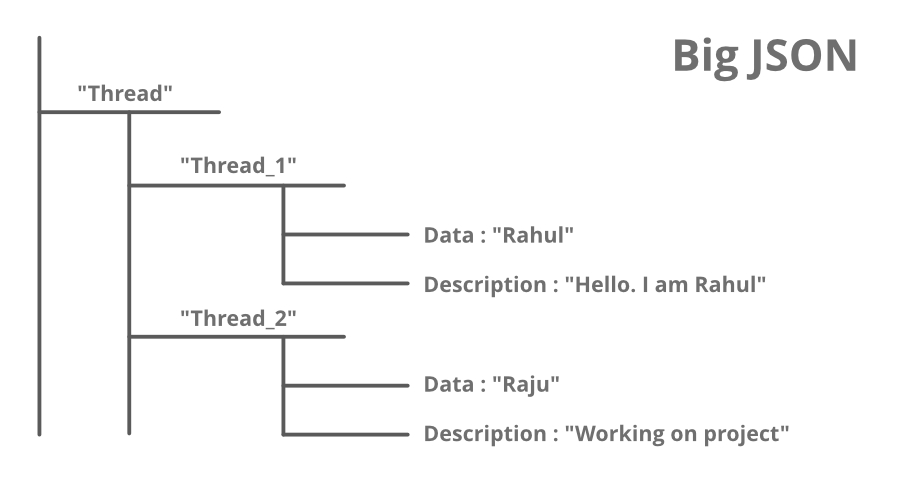
**Firebase**: The Firebase Realtime Database is a cloud-hosted NoSQL database that lets you store and sync data between your users in real-time. Cloud Firestore enables you to store, sync and query app data at global scale.

****

**Setup Firebase:**

Create a Firebase project on the Firebase Console(<https://console.firebase.google.com/>).

Add your Android app to the Firebase project, and download the ‘google-services.json’ configuration file. Firebase is chosen as the backend for several reasons the selection of Android Studio, Java, and Firebase as our technology stack ensures a powerful, efficient, and scalable solution for our GPS tracking system. These technologies are well-established in the Android app development ecosystem and provide the necessary tools for creating a reliable and user-friendly application.



**Java**: It is a widely used, high level, object-oriented programming language known for its platform independence, readability, and robustness. Here’s a brief Java is a widely-used programming language for coding web applications.

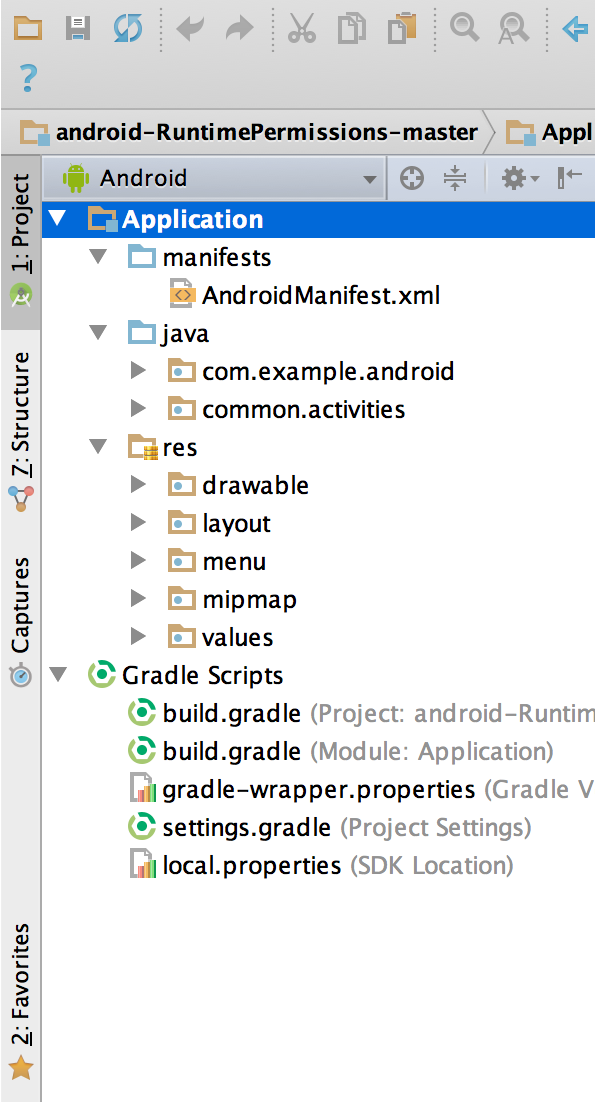
Java is a time-tested and reliable programming language with a strong presence in Android development.It offers platform compatibility and a vast developer community, making it ideal for building Android applications.Java's object-oriented nature simplifies app development and maintenance. Java programming language this includes the basic vocabulary and rules used to write algorithms such as primitive data types, if/else blocks, loops, etc.

**ANDROID STUDIO**: Android Studio is the official Integrated Development Environment (IDE)for Android app development. It offers a robust set of tools and resources for creating high-quality Android applications.It provides an intuitive user interface and a range of debugging and testing capabilities, making it the preferred choice for Android development.

Android Studio Setup:

Create a new Android Studio project or open an existing one.

Make sure your Android Studio project is configured to use Firebase by adding the ‘google-services.json’ file to the app module.



Permission:

In Androidmainfest.xml file, request the necessary permission for location access. You’ll need the ‘ACCESS\_COARSE\_LOCATION’ permissions.

Google Play Services:

Add Google Play Services dependency in your app’s build.gradle file.

Location Services:

Create a Java class to handle location services. You can use the Fused Location Provider Client to request and retrieve the user’s current location. These key features collectively enhance user experience, security, and operational efficiency within the bike rental service, making it a comprehensive and innovative solution.

### ABOUT DEPARTMENT

Our organization offers two master degree courses namely (M.Sc) Master in Science (Computer Science) and Master in Computer Applications (M.C.A). The best facilities are provided by the Department for the education of the students. Highly talented faculties are available for the quality education and a very friendly environment is created. Department is very well-equipped with laboratory, class rooms, library. Also department library is equipped with 2000 volumes, 72 e-journals are accessible through UGC-INFLIBNET, 4 popular journals related to Information Technology, Three Computer laboratories are setup for the good education for students. Backups are maintained well even though there are 90 computers in the department. Students are provided with the facility of Wi-Fi so as to get connected to the internet. Our organization is one of the best organizations in quality education.

**CHAPTER 2:**

**SYSTEM STUDY**

**EXISTING SYSTEM WITH LIMITATION:**

In the present day bike tracking is becoming essential for the purpose of improving our life condition. Convenience and ease of using bike is what home bike tracking is offering. Bike tracking offers a futuristic way of life in which an individual gets to control his bike using a smart phone, from tracking a bike /detecting accidental place of a bike; it also offers an efficient use of technology. But to get or acquire such system installed will cost a lot of money and that is the major reason of why bike tracking has not received much demand and attention, adding to that also the complexity of installing it and configuring it. Thus it is essential to make it cost effective and easy to configure, if this is granted to people then they will be willing to acquire it in their personal bikes, school buses and taxes/cabs etc. In other words, a system modification for the bike tracking is required in order to lower the price of applying it to bikes. Also this tracking project can be used to purpose of women safety as well as parents can be used to take bikes of their child/kid for the safety or missing purpose or to track their activities for their future. Even more realistically this project can be used to track airline baggage because as we know every year almost 13% airline baggage used to get missing by a worldwide survey.

**PROPOSED SYSTEM WITH OBJECTIVES:**

The proposed system is used for positioning and navigating the bike. The Exact location is indicated in the form of latitude and longitude along with the exact Navigated track on Google map. The system tracks the location of particular bike and sends to user’s mobile in form of data and also to microcontroller. The arrived data, in the form of latitude and longitude is used to locate the bike on the Google maps.

A user-friendly app that empowers users to easily locate, rent, and enjoy bikes.Enhanced bike security through real-time GPS tracking.Data-driven insights for providers to improve fleet management and service quality.A sustainable and eco-friendly solution for urban transportation.Feel free to elaborate on each objective and its intended outcomes during your presentation.

**FEASIBILITY STUDY:**

Conducting a feasibility study for GPS tracking of bike rentals is a valuable step to assess the viability of such a system. Here are some business proposition

* 1. Economic Feasibility
  2. Technical Feasibility
  3. Duration

### Economic Feasibility:

### Research the market for bike rentals in your target area. Analyze the competition, demand for bike rental & potential customers base. Create economic projection that include revenue estimate, cost savings & ROL calculations over a specified time period.

### Technical Feasibility:

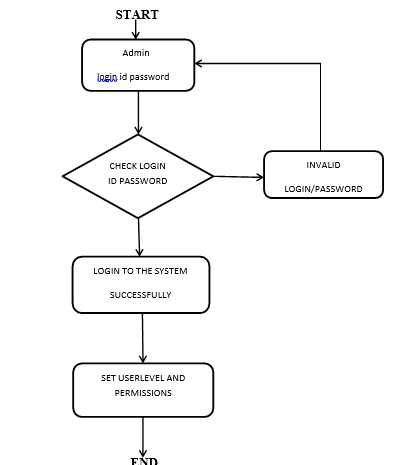
### The technical feasibility study compares cost associated with implementing GPS tracking, include H/W, S/W & maintenance. Compare these costs to the potential benefits, such as reduce theft, improved fleet management & increased customer satisfaction.

**DURATION:**

|  |  |  |
| --- | --- | --- |
| DESCRIPTION | FROM | TO |
| Requirements analysis, understating the basic problem statement | 04/06/2023 | 05/07/2023 |
| Scheduling & planning for the project | 07/07/2023 | 15/07/2023 |
| Data Analysis & Exploration | 17/07/2023 | 29/08/2023 |
| Coding & Testing for conditions | 01/08/2023 | 02/09/2023 |
| Resolving the Errors (if found while testing) | 04/09/2023 | 13/09/2023 |
| Documentation & Submission of the final project report. | 15/09/2023 | 30/09/2023 |

**CHAPTER 3:**

**SYSTEM ANALYSIS**



### FIG: SYSTEM FLOWCHART

**ENTITY RELATIONSHIP DIAGRAM (E-R DIAGRAMS):**

A conceptual model describes the essential features of system data. This conceptual model is described modeling method known as Entity Relationship. Entity Relationships is uses three major abstractions to describe data. To describe the basic data-structuring concepts and constraints of the ER model and discuss their use in the design of conceptual schemas for database applications. We also present the diagrammatic notation associated with the ER model, known as “**E-R diagrams”**.

**Elements of ER-Diagram**

1. **ENTITY:**

An Entity can be any object place, person or class. In ER-Diagram, An Entity is represented using rectangles. Consider an example of an organization. Employee, Manager, Department, Product and many more can be taken as Entities from an organization.

1. **WEAK ENTITY:**

Weak Entity is an entity that depends on another entity. Weak entity doesn’t have key attribute of their own. A weak entity, in the context of database design, is an entity that does not have sufficient attributes to form a primary key. It depends on another entity, called a strong entity, to provide part of or all of its primary key

1. **ATTRIBUTE:**

An attribute describes the properties or characteristics of an entity. For Example, name, age, address etc. can be attributes of an Employee. An attribute is represented using ellipse.

1. **KEY ATTRIBUTE:**

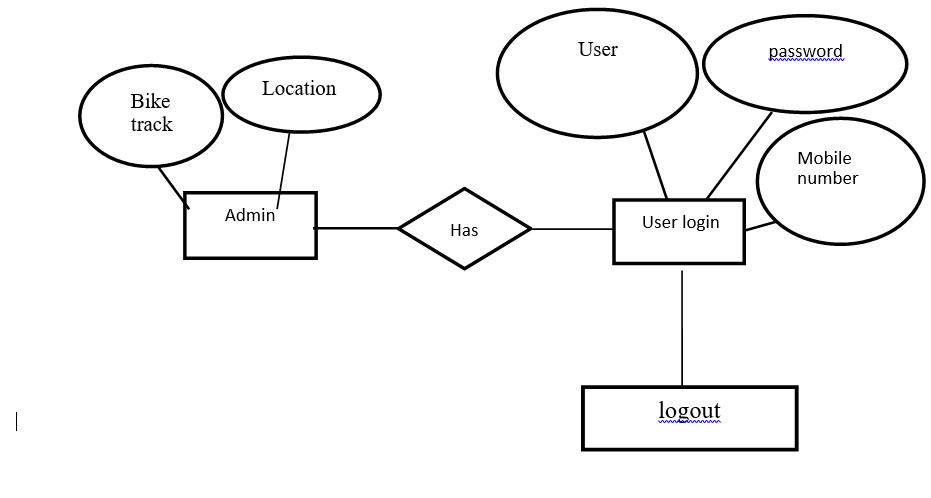
Key attribute represents the main characteristic of an entity. Is use to represent primary key. Ellipse with underlines represent key attribute.

1. **MULTIVALUE ATTRIBUTE:**

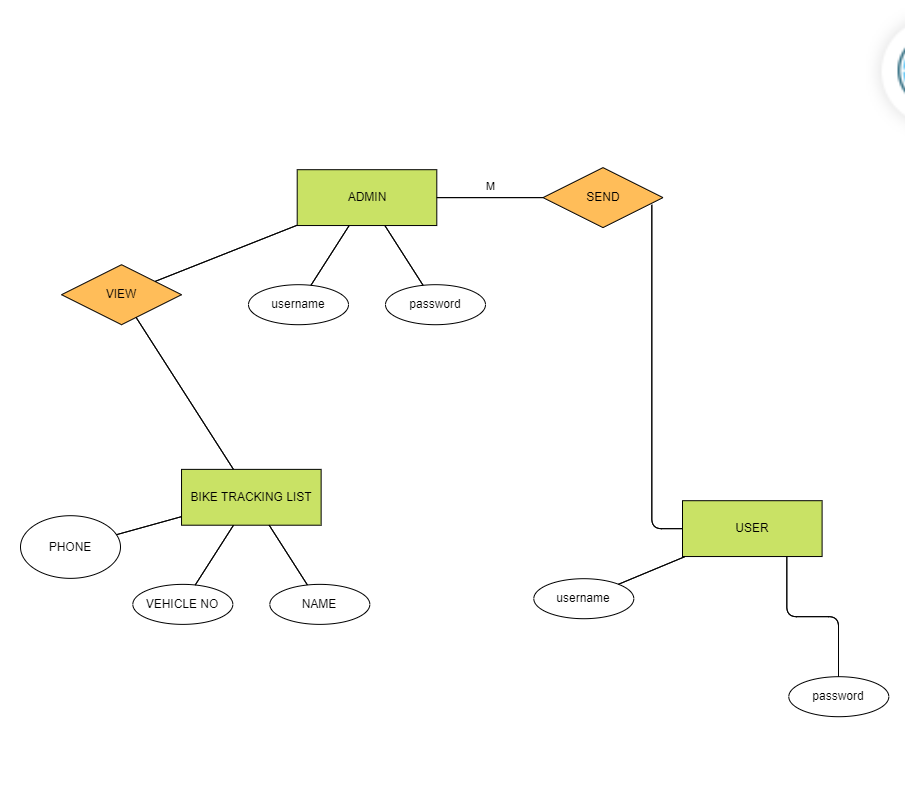
A multi-valued attribute in the context of database design is an attribute that can hold multiple values for a single entity. In other words, a multi-valued attribute can have a set of values associated with a single entity occurrence.

1. **RELATIONSHIP:**

A Relationship describes the relation between entities. Relationship is represents using diamonds.



**FIG: E-R DIAGRAM**



**Fig: Project Flow and ER-Diagram**

### 

### DATA FLOW DIAGRAM (DFDs):

Data flow diagrams are a graphical tool used to describe and analyse the movement of data through a system. DFD’s are used to capture the essential feature of both existing real system and future physical implementation of the system.

### Different notations used in DFD are:

**Functional Processing:**

It is represented by an oval. The processing or main transactions are specified by this notation.

### Data Flow:

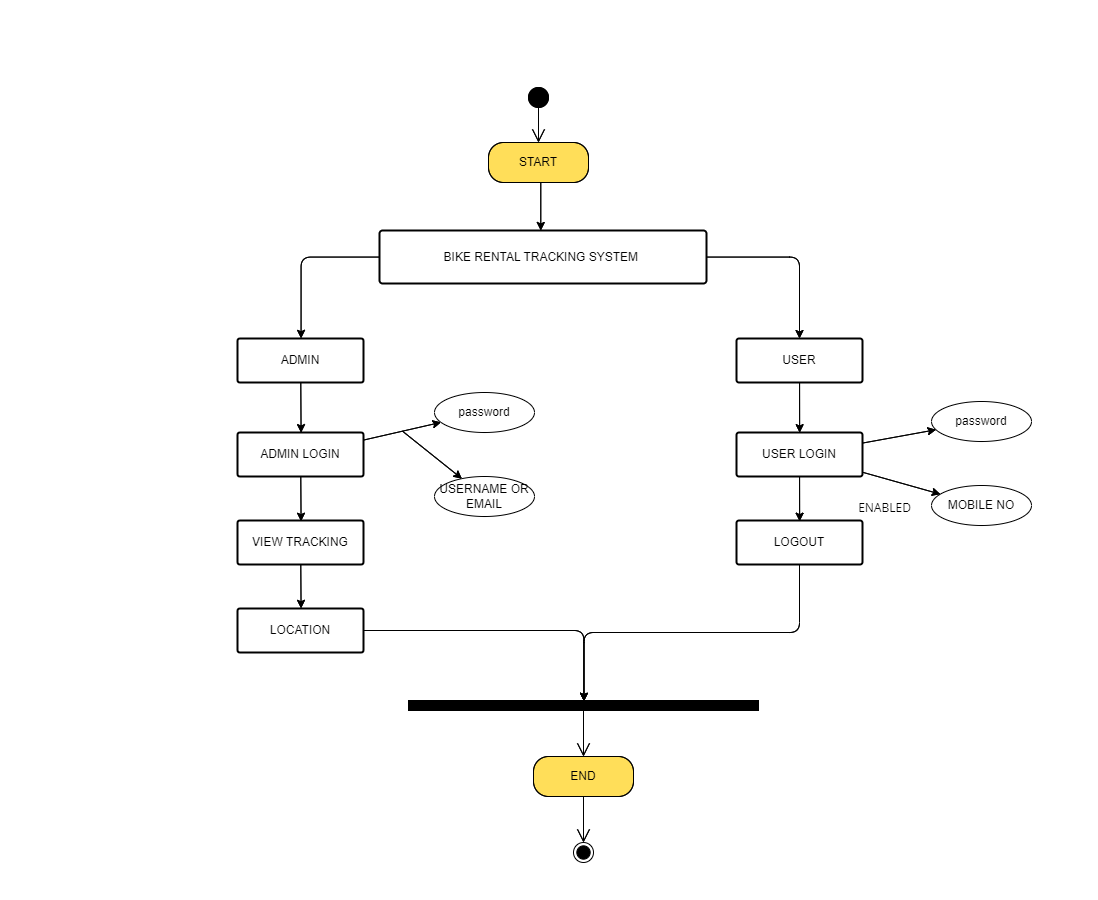
It is represented by an arrow line and name of the data is specified by the side of the line as label. This is used for data movement.

### Data Store:

It is represented by one open-end rectangle. The databases used in the system are specified by this notation.

### Source or sink:

It is represented by one open-end rectangle. It is used for specifying from where data comes and where it reaches.

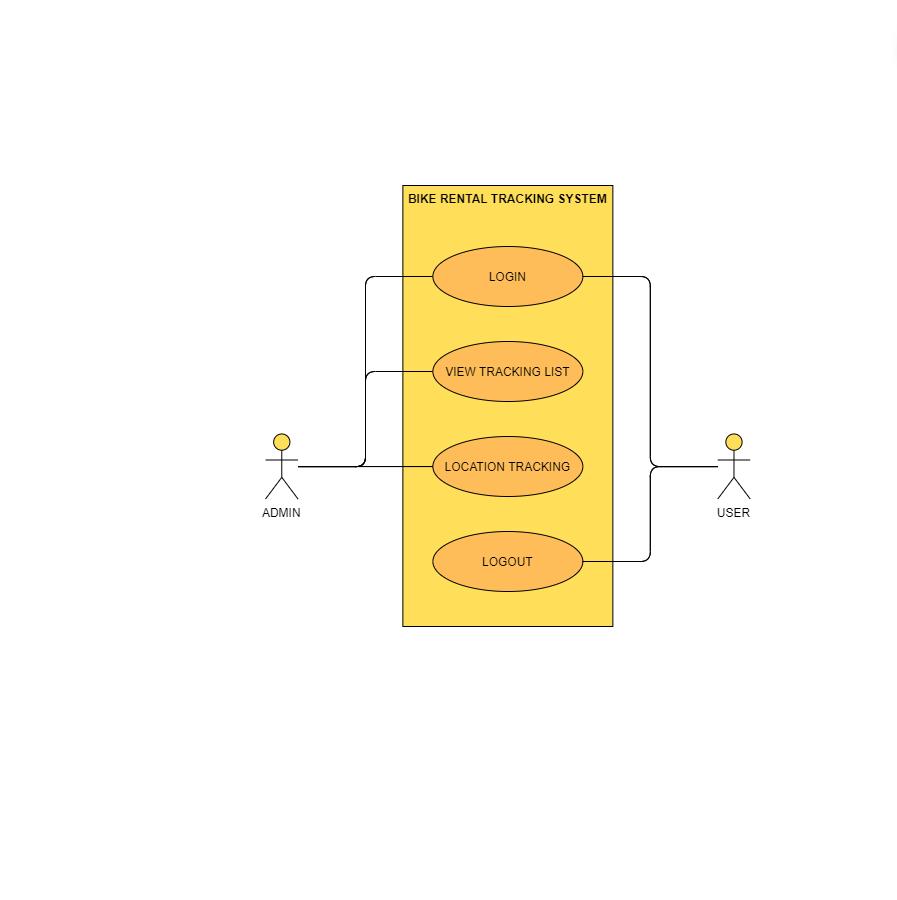


### Fig: DFD DIAGRAM

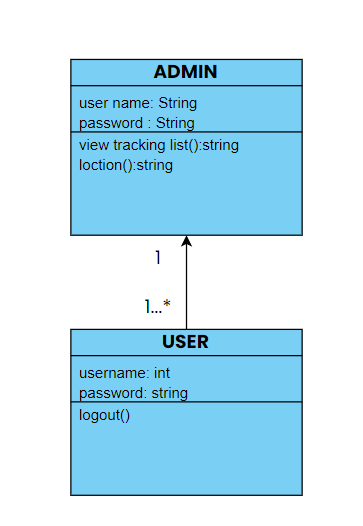
### 

### Unified Modeling Diagram (UML):

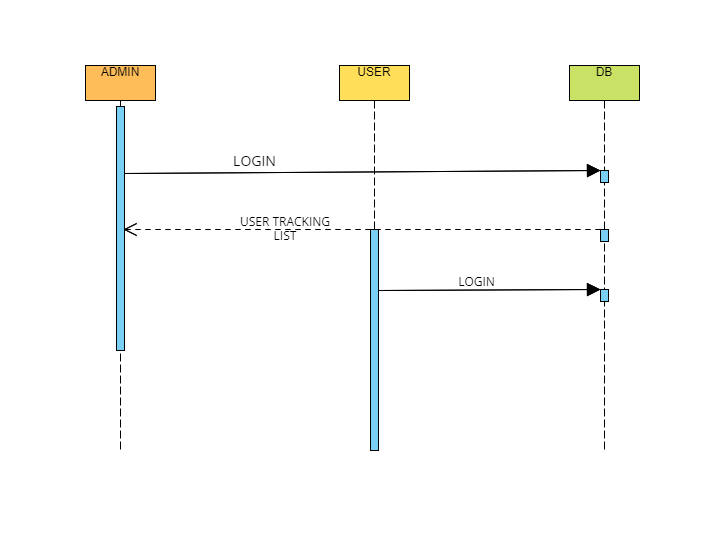
1. **Use Case Diagram**

****

**2. Class/Object Diagram**

****

**3. Sequence Diagram**

****

**REQUIREMENT SPECIFICATION:**

System requirements are the configuration that a system must have in order for a hardware or software application to run smoothly and efficiently. Failure to meet these requirements can result in installation problems or performance problems. The former may prevent a device or application from getting installed, whereas the latter may cause a product to malfunction or perform below expectation or even to hang or crash.

**Hardware Requirements:**

Hardware basically refers to the item in a PC that can be touched and felt like keyboard, monitor, mouse and the system unit. Additional hardware components that can be added to the PC are modem, printer, scanner etc.

The hardware thus used here are consists of:

Processor : intel i3 & Mobile

Hard Disk : 500GB

RAM : 4GB & above

Monitor : 15” LED

Input Devices : Keyboard. Mouse.

#### Software Requirement:

There is use of large software in developing this project. Software is basically the logical program that handles different components which cannot be touched or felt and helps to interact with one another in a Hassle-free manner.

The software used here consists of:

Operating system : WINDOWS 10.

Coding language : JAVA.

Tool : Android studio

Database : Firebase.

### CHAPTER 4:

### SYSTEM DESIGN

### File/Database Design:

### 

### 1. ADMIN:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **COLUMN**  **NAME** | **COLUMN**  **TYPE** | **LENGTH** | **NULL?** | **CONSTRAINTS** |
| Id | INT | 50 | NOT NULL | PRIMARY KEY |
| NAME | VARCHAR | 50 | NO |  |
| ADDRESS | VARCHAR | 50 | NO |  |

**2. ADMIN\_PERMISSION:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **COLUMN**  **NAME** | **COLUMN**  **TYPE** | **LENGTH** | **NULL?** | **CONSTRAINTS** |
| CODENAME | VARCHAR | 50 | NO |  |
| CONTENT\_TYPE  \_ID | INT | 50 | NO | MUL |
| ID | INT | 50 | NO | PRIMARY KEY |
| NAME | VARCHAR | 50 | NO |  |

### 3. ADMIN\_USER\_PERMISSION:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **COLUMN**  **NAME** | **COLUMN**  **TYPE** | **LENGTH** | **NULL?** | **CONSTRAINTS** |
| ID | BIGINT | 50 | NOT NULL | PRIMARY KEY |
| PERMISSION\_ID | INT | 50 | NO | MUL |
| USER\_ID | INT | 50 | NO | MUL |

* 1. **User :**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **COLUMN**  **NAME** | **COLUMN**  **TYPE** | **LENGTH** | **NULL?** | **CONSTRAINTS** |
| MODIFIEDDATE | DATETIME | - | NO |  |
| USER\_CONTACT | VARCHAR | 50 | NO |  |
| USER\_CREATED\_  ON | DATETIME | - | NO |  |
| USER\_EMAIL | VARCHAR | 50 | NO |  |
| USER\_PASSWORD | VARCHAR | 50 | NO |  |

### NORMALIZATION

The normalization process takes a relational schema through a series of tests to “certify” whether it satisfies a certain normal form. The process, which proceeds in a top down fashion, by evaluating each relation against the criteria for normal forms and decomposing the relations as necessary can thus be considered as the normalization process or relational design by analysis. The normalization criteria followed in the project include the following:

1. First Normal Form (1NF)
2. Second Normal Form (2NF)
3. Third Normal Form (3NF)

### 1. First Normal Form:

It is defined to disallow multi-valued attributes, composite attributes and their combinations. It states that the domain of the attribute must include only atomic values and that the value of any attribute in the domain must be a single value from the domain of that attribute. Hence 1NF disallows having a set of values, a tuple of values, a combination of both as an attribute value of a single tuple. In other words, 1NF disallows

### 2. Second Normal Form:

The test for 2NF involves testing for functional dependencies whose left-hand attributes are part of the primary-key. If the primary-key has a single attribute then the test need not be applied at all. A relational schema R is said to be in 2NF if every non-prime attribute in R is fully functionally dependent on the primary key of R.**Remedy:** Decompose and setup a new relation for each partial key with its dependent attributes. Make sure to keep a relation with the original primary key and any attributes that are functionally dependent on it.

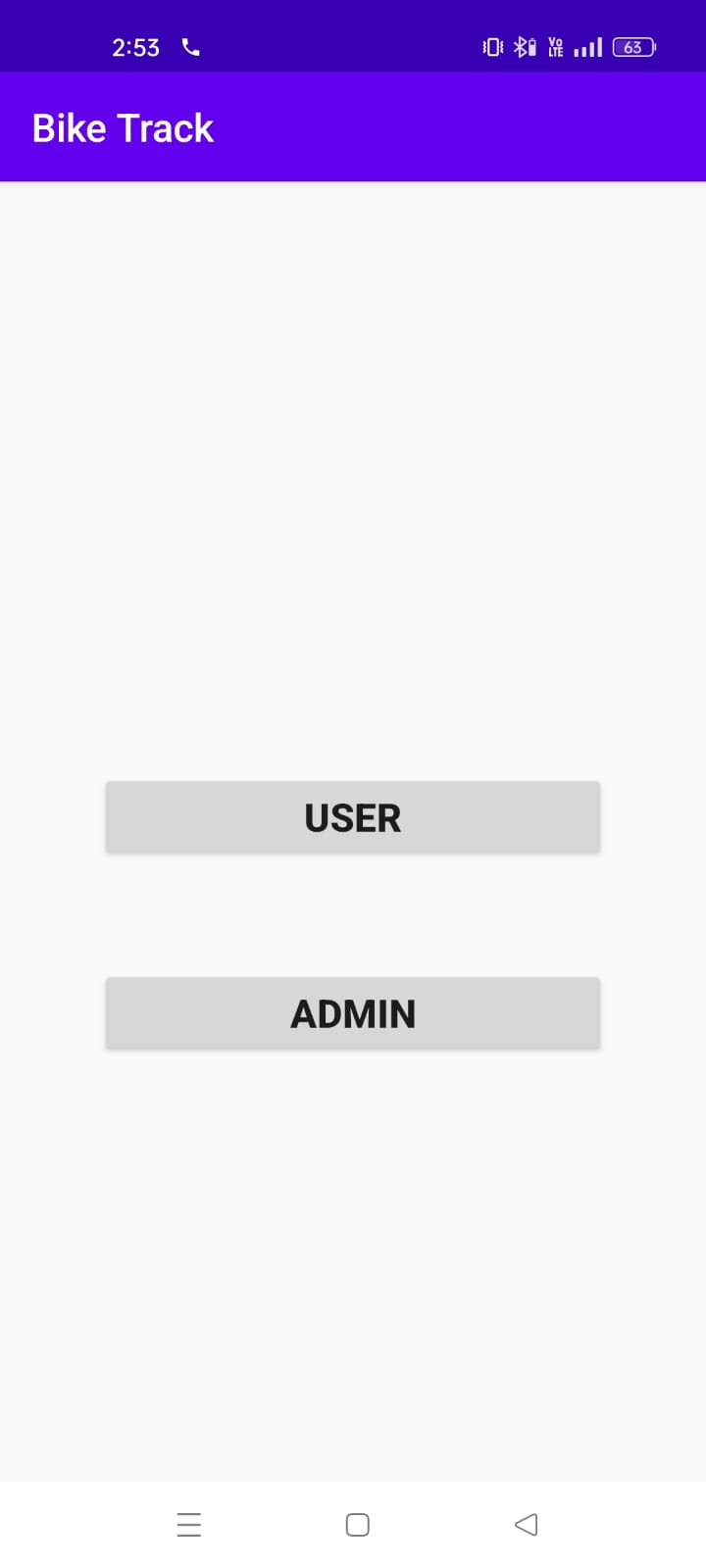
### 3. Third Normal Form:

This is based on the concept of transitive dependency. A functional dependency X -> Y in a relation R is a transitive dependency if there is a set of attributes Z that is neither a candidate key nor a subset of any key of R, and both X –>Z and Z -> Y hold. In other words, a relation should not have a non-key attribute functionally determined by another non-key attribute (or by a set of non-key attributes).**Remedy:** Decompose and set up a relation that includes the non- key attribute(s) that functionally determine other non-key attributes.

All the relations in our project satisfy fully “functional dependency” property and hence are in the Third Normal Form.

**Input / Output form design:**

**HOME PAGE:**



### LOGIN PAGE :

### C:\Users\hp\OneDrive\Pictures\Screenshots\WhatsApp Image 2023-04-15 at 3.44.25 PM (1).jpegSTATUS PAGE:

### C:\Users\hp\OneDrive\Pictures\Screenshots\WhatsApp Image 2023-04-15 at 3.44.26 PM.jpeg

### Bike tracking list:

### C:\Users\hp\OneDrive\Pictures\Screenshots\WhatsApp Image 2023-04-15 at 3.44.26 PM (1).jpeg

**Location:**

### C:\Users\hp\Downloads\WhatsApp Image 2023-04-15 at 3.44.27 PM.jpeg

### CHAPTER 5:

### CODING

### Source/Pseudo Code:

**Admin log activity**

packagecom.bike.track;

importandroid.content.Intent;

importandroid.os.Bundle;

importandroid.view.View;

importandroid.widget.ArrayAdapter;

importandroid.widget.Button;

importandroid.widget.EditText;

importandroid.widget.Spinner;

importandroid.widget.TextView;

importandroid.widget.Toast;

importandroidx.appcompat.app.AppCompatActivity;

importcom.google.firebase.database.FirebaseDatabase;

importjava.util.HashMap;

public class AdminLogActivity extends AppCompatActivity {

EditTexteditEmail, editPass;

Button btnLogin;

@Override

protected void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.activity\_adminlog);

editEmail = findViewById(R.id.editEmail);

editPass = findViewById(R.id.editPass);

btnLogin = findViewById(R.id.btnLogin);

btnLogin.setOnClickListener(new View.OnClickListener() {

@Override

public void onClick(View view) {

if(editEmail.getText().tostring().equal([admini@gmail.com](mailto:admini@gmail.com))

&&editPass.getText().toString().equals(“admin”)) {

Intent i = new Intent(AdminLogActivity.this, UsersList.class);

startActivity(i);

}

else {

Toast.makeText(getpplictionContext(),”invalid Data”

Toast.LENGTH\_LONG).show();

}

}

});

}

}

**Main activity**

packagecom.bike.track;

importandroid.content.Intent;

importandroid.os.Bundle;

importandroid.view.View;

importandroid.widget.Button;

importandroidx.appcompat.app.AppCompatActivity;

public class MainActivity extends AppCompatActivity {

Button btnUser, btnAdmin;

@Override

protected void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.activity\_main);

btnUser = findViewById(R.id.btnUser);

btnAdmin = findViewById(R.id.btnAdmin);

btnUser.setOnClickListener(new View.OnClickListener() {

@Override

public void onClick(View view) {

Intent i = new Intent(MainActivity.this, UserLogActivity.class);

startActivity(i);

}

})

btnAdmin.setOnClickListener(new View.OnClickListener() {

@Override

public void onClick(View view) {

Intent i = new Intent(MainActivity.this, AdminLogActivity.class);

startActivity(i);

}

});

}

}

**Map activity**

packagecom.bike.track;

importandroidx.appcompat.app.AppCompatActivity;

importandroid.os.Bundle;

importandroid.webkit.WebView;

public class MapActivity extends AppCompatActivity {

@Override

protected void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.activity\_map);

WebViewwebView = (WebView) findViewById(R.id.webView);

webView.getSettings().setJavaScriptEnabled(true);

webView.loadUrl("http://maps.google.com/maps?q=loc:"+12.9581667+","+77.7146389+"(""Label Which you want" + ")");

}

}

**User**

packagecom.bike.track;

importandroid.app.Activity;

importandroid.app.ActivityManager;

importandroid.app.Service;

importandroid.content.Context;

importandroid.content.Intent;

importandroid.content.pm.PackageManager;

importandroid.location.LocationManager;

importandroid.os.Bundle;

importandroid.util.Log;

importandroid.view.View;

importandroid.widget.Button;

importandroid.widget.Toast;

importandroidx.appcompat.app.AppCompatActivity;

importandroidx.core.app.ActivityCompat;

importandroidx.core.content.ContextCompat;

importcom.bike.track.gps.GPSTracker;

public class User extends AppCompatActivity {

private static final String TAG = "HomeScreen";

@Override

protected void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.activity\_user);

try {

if((ActivityCompat.checkSelfPermission(this,android.Manifest.permission.

ACCESS\_FINE\_LOCATION)!=PackageManager.PERMISSION\_GRANTED) &&(ActivityCompat.checkSelfPermission(this,android.Manifest.permission.ACCESS\_COARSE\_LOCATION)!=PackageManager.PERMISSION\_GRANTED) ){ActivityCompat.requestPermissions(this,newString[]{

android.Manifest.permission.ACCESS\_FINE\_LOCATION,android.Manifest.permission.ACCESS\_COA RSE\_LOCATION}, 101);

}

} catch (Exception e){

e.printStackTrace();

}

getLocation();

Button logout = findViewById(R.id.btn\_logout);

logout.setOnClickListener(view -> {

stopService(new Intent(this, GPSTracker.class));

finish();

});

}

@Override

protected void onDestroy() {

super.onDestroy();

Log.d(TAG, "onDestroy: called");

}

public void getLocation(){

Log.d(TAG, "getLocation: called");

if(!isMyServiceRunning(GPSTracker.class)) {

startService(new Intent(this, GPSTracker.class));

}else{

Log.d(TAG, "getLocation: service already running");

}

}

privatebooleanisMyServiceRunning(Class<?>serviceClass) {

ActivityManager manager = (ActivityManager) getSystemService

(Context.ACTIVITY\_SERVICE);

for(ActivityManager.RunningServiceInfoservice:manager.getRunningServices

(Integer.MAX\_VALUE)) {

if (serviceClass.getName().equals(service.service.getClassName())) {

Log.d(TAG,"isMyServiceRunning:serviceRunning"+service.service.

getClassName());

return true;

}

}

return false;

}

}

**User data**

packagecom.bike.track;

public class UserData {

private static UserData INSTANCE;

private String user = null;

privateUserData() {

}

public static UserDatagetInstance() {

if(INSTANCE == null) {

INSTANCE = new UserData();

}

return INSTANCE;

}

public String getUser() {

return user;

}

public void setUser(String user) {

this.user = user;

}

}

**User log activity**

packagecom.bike.track;

importandroid.content.Intent;

importandroid.os.Bundle;

importandroid.text.TextUtils;

importandroid.view.View;

importandroid.widget.Button;

importandroid.widget.EditText;

importandroidx.appcompat.app.AppCompatActivity;

importcom.google.firebase.database.DatabaseReference;

importcom.google.firebase.database.FirebaseDatabase;

public class UserLogActivity extends AppCompatActivity {

EditTexteditMob, editPass;

Button btnLogin;

@Override

protected void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.activity\_userlog);

editMob = findViewById(R.id.editMob);

editPass = findViewById(R.id.editPass);

btnLogin = findViewById(R.id.btnLogin);

btnLogin.setOnClickListener(new View.OnClickListener() {

@Override

public void onClick(View view) {

String mob1 = "1234567890";

String pass1 = "akshay123";

String mob2 = "0987654321";

String pass2 = "sonu123";

if(TextUtils.isEmpty(editMob.getText().toString())) {

editMob.setError("Mobile number cannot be Empty");

return;

}

if(TextUtils.isEmpty(editPass.getText().toString())) {

editPass.setError("Password cannot be Empty");

return;

}

if(editMob.getText().toString().equals(mob1)&&editPass.getText().to

String().equals(pass1))

{

UserData.getInstance().setUser("user1");

Intent i = new Intent(UserLogActivity.this, User.class);

startActivity(i);

}

If(editMob.getText().toString().equals(mob2) &&editPass.getText().to

String().equals(pass2))

{

UserData.getInstance().setUser("user2");

Intent i = new Intent(UserLogActivity.this, User.class);

startActivity(i);

}

}

});

}

}

**User list**

packagecom.bike.track;

importandroidx.annotation.NonNull;

importandroidx.appcompat.app.AppCompatActivity;

importandroid.content.Intent;

importandroid.net.Uri;

importandroid.os.Bundle;

importandroid.util.Log;

importandroid.widget.Button;

importandroid.widget.TextView;

importcom.google.firebase.database.DataSnapshot;

importcom.google.firebase.database.DatabaseError;

importcom.google.firebase.database.DatabaseReference;

importcom.google.firebase.database.FirebaseDatabase;

importcom.google.firebase.database.ValueEventListener;

public class UsersList extends AppCompatActivity {

private static final String TAG = "UsersList"{

privateTextView tvLoc1, tvLoc2;

@Override

protected void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.activity\_userslist);

Button userView1 = findViewById(R.id.btn\_view1);

Button userView2 = findViewById(R.id.btn\_view2);

tvLoc1 = findViewById(R.id.tv\_loc1);

tvLoc2 = findViewById(R.id.tv\_loc2);

Location user1Location = fetchLocationData("user1");

Location user2Location = fetchLocationData("user2");

userView1.setOnClickListener(view -> {

openMap(user1Location.lat, user1Location.lng, "user1");

});

userView2.setOnClickListener(view -> {

openMap(user2Location.lat, user2Location.lng, "user2");

});

}

private Location fetchLocationData(String user) {

Location location = new Location();

DatabaseReferencedbRefFirebaseDatabase.getInstance().getReference().

child("Users").child(user);

dbRef.getRef().addValueEventListener(new ValueEventListener() {

@Override

public void onDataChange(@NonNullDataSnapshot snapshot) {

try {

location.lat = (double) snapshot.child("Lat").getValue();

location.lng = (double) snapshot.child("Lng").getValue();

if(user.equals("user1")) {

tvLoc1.setText("Lat: "+location.lat+", Lng: "+location.lng);

}else if(user.equals("user2")) {

tvLoc2.setText("Lat: "+location.lat+", Lng: "+location.lng);

}

} catch (Exception e) {

e.printStackTrace();

}

}

@Override

public void onCancelled(@NonNullDatabaseError error) {

// Getting Post failed, log a message

Log.e(TAG, "loadPost:onCancelled", error.toException());

}

});

return location;

}

private void openMap(double lat, double lng, String user) {

String strUri="http://maps.google.com/maps?q=loc:"+lat+","+lng+"

("+user+ ")";

Intent intent = new Intent(android.content.Intent.ACTION\_

VIEW, Uri.parse(strUri));

intent.setClassName("com.google.android.apps.maps","com.google

.android.maps.MapsActivity");

startActivity(intent);

}

public class Location {

doublelat;

doublelng;

}

}

### 

### CHAPTER 6:

### SYSTEM TESTING

### Preparation of Test Data & Test:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Test Case** | **Input** | **Expected O/P** | **Actual O/P** | **Result** |
| 1 | Valid Username and Password | It should display respective page according to user type. | Respective Home is displayed  . | Passed |
| 2 | InvalidUsername and Password | It should give appropriate error message saying “Enter proper User-Id and Password” | Error message Displayed | Passed |
| 3 | Add/delete/ Inventory items | It should Add/delete Inventory items | Added/Deleted inventory items successfully | Passed |
| 4 | Add User Details | It should  Add User detail | Added User details  successfully | passed |
| 5 | List users | Add user details | Track location  successfully | passed |

### 

### Unit Testing and other Classes of Testing

### Black box Testing:

Black box testing is done to find:

* Incorrect or missing functions.
* Interface Errors
* Errors in external database access
* Performance error.
* Initialization and termination error.

### White box Testing:

White box testing is done to find out:

* + Check whether all independent paths within a module have been exercised at least once
  + Exercise all logical decision on their true and false sides
  + Execute all loops at their boundaries and within their bounds.
  + Execute all loops at their boundaries and within their bounds.
  + Ensure whether all the possible validity checks and validity lookups have been provided to validate data entry.

### Unit Testing:

Unit testing involves, checking all the modules in the system individually against the specification produced during the design of the module and for their performance. Unit testing also involves code produced in the coding phase and hence the internal logic of the program. Each module is tested for different test cases design to check each specific combination of conditions handled by the program. Error handlers are included in each module for each event trap and handled the errors.

### Integration Testing

Integration testing takes as its input modules that have been checked out by unit testing, groups them in larger aggregates, applies tests defined in an Integration test plan to those aggregates, and delivers as its output the integrated system ready for system testing. The purpose of integration testing is to verify functional, performance and reliability requirements placed on major design items.

### System Testing

System testing of software or hardware is testing conducted on a complete, integrated system to evaluate the system's compliance with its specified requirements. System testing falls within the scope of black box testing, and as such, should require no knowledge of the inner design of the code or logic. As a rule, system testing takes, as its input, all of the "integrated" software components that have successfully passed integration testing and also the software system itself integrated with any applicable hardware system(s). The purpose of integration testing is to detect any inconsistencies between the software units that are integrated together (called *assemblages*) or between any of the assemblages and the hardware. System testing is a more limited type of testing; it seeks to detect defects both within the "inter-assemblages" and also within the system as a whole.

Quality Assurance: Ratings and reviews contribute to maintaining service quality. User Trust: Positive reviews build trust among potential users. Community Building: A sense of community is fostered as users share their experiences. Feedback Loop: User feedback is a vital component of our continuous improvement cycle. Regularly analyzing ratings and reviews guides our efforts in enhancing the service. User-Centric Approach: Our commitment to user feedback demonstrates our user-centric approach. Users' voices are valued and drive improvements in the bike rental experience.

### CHAPTER7:

### SYSTEM SECURITY

### CHECKS AND CONTROL:

The protection of computer based resources that includes hardware, software, data, procedures and people against unauthorized use or natural Disaster is known as System Security.

System Security can be divided into four related issues:

* Security
* Integrity
* Privacy
* Confidentiality

### SYSTEM SECURITY:

It refers to the technical innovations and procedures applied to the hardware and operation systems to protect against deliberate or accidental damage from a defined threat.

### DATA SECURITY:

It is the protection of data from loss, disclosure, modification and destruction. GPS tracking deters theft and enables quick recovery of lost or stolen bikes. Users can ride with confidence, knowing their rentals are secure.

### SYSTEM INTEGRITY:

It is a central hub for ensuring smooth operations, optimizing bike placement, and making informed decisions based on GPS data and user analytics. It empowers administrators to efficiently manage the bike rental service and maintain a high standard of quality and security

### PRIVACY:

It's important to note that user privacy is a top priority. Data is anonymized and aggregated to ensure individual user identities are not disclosed.

### CONFIDENTIALITY:

It is a special status given to sensitive information in a database to minimize the possible invasion of privacy. It is an attribute of information that characterizes its need for protection.

The user registration and authentication process is a crucial aspect of our GPS tracking system, ensuring that user data is protected and accessed securely while providing a seamless on boarding experience for users. Firebase Authentication simplifies this process and enhances security

### ENCRYPTION SECURE:

In your web application we have provided security such as session this Session is used all web forms.

Tokenization: User payment details are tokenized, ensuring that sensitive data is not stored on the app or server.

Authentication: Two-factor authentication and user verification processes enhance payment security.

Compliance: Payment systems adhere to industry standards and regulations for data security.

Security user data is encrypted and securely stored. Ease of use: Users can log in quickly using various methods. Scalability: Handles user authentication for growing user bases. Compatibility: Seamlessly integrates with other Firebase services.

### CHAPTER 8:

### CONCLUSION

### CONCLUSION:

To conclude the description about the project, the project, developed using JAVA with MySQL is based on the requirement specification of the user and the analysis of the existing system, with flexibility for future enhancement. This project focuses on how to track bike.Finally we conclude that our project bike rental tracking system is an efficient system to track rented bikes. These systems is tested on different test cases and showed positive result.TheBike rental tracking system portal is developed to facilitate easy processing of tracking bike procedures. Manually, this consumes a lot of time, effort and paper work. And also it is possible to freely submit the feedback without any hesitation. So, this portal overcomes all these limitations and offers a great deal of help at each and every stage in the whole process of availing a leave.So, bikes are the only solution that can come to the rescue here and help people reach their destination on time. Companies have started launching bike rental mobile apps as it is a great relief to the eco-system from vehicle pollution and the daily honking. Secondly, there will be considerably lesser cars seen on the roads, hence, less traffic.

### LIMITATIONS:

**Hardware/software:** This includes cost of sophisticated computer, GPS etc. **Coverage** **and** **Accuracy:** GPS signals may be weak or inaccurate in certain areas, such as densely populated urban areas, forests or tunnels. This can affect the precision of tracking.

**Battery Life:** Continous GPS tracking can drain the battery of the rental bike or the tracking device. This limits how long the bike can be used before recharge.

**Privacy Concerns:** GPS tracking raises privacy issues, as it can potentially record the location & movement of uses.

**Data Handling:** Collecting & managing GPS data requires robust infrastructure & data security measures to protect user information.

**User Experience:** Some users may find the idea of being tracked uncomfortable, leading to reduces adoption of bike rental services.

### FUTURE SCOPE:

We can use the EEPROM to store previous Navigating positions up to N number of locations by increasing its memory. We can reduce the size of the kit by using GPS+GSM on the same module. We can increase the accuracy up to 3m by increasing the cost of the GPS receivers. We can use our kit for detection of bomb by connecting to the bomb detector. With the help of high sensitivity vibration sensors, we can detect the accident. whenever bike unexpectedly had an accident on the road with the of vibration sensor, we can detect the detect the accident & we can send the location to the owner, hospital & police. We can use our kit to assist the traffic.

Smart Lock Integration Incorporate smart locks that can be controlled through the app, offering added security and convenience. Implement predictive maintenance algorithms to proactively identify and address bike maintenance needs, reducing downtime. Advanced geofencing Enhance geofencing capabilities to enable more precise area-based promotions and notifications. Multimodal integration explore partnerships with other urban transportation services, such as public transit, to provide seamless multimodal experiences for users. User rewards program introduce a rewards program to incentivize frequent users and foster brand loyalty. Voice commands develop voice-command features for a truly hands-free user experience, making it safer and more convenient for users on the go. Augmented reality (AR) navigation explore features to guide users to the nearest bikes and offer immersive experiences. Integration with sustainable initiatives collaborate with environmental and sustainability initiatives to further promote eco-friendly transportation. Gamification implement elements to engage users and encourage healthy competition.

### BIBLIOGRAPHY:

### Books:

* Parry, A. (2013). Event planners rejoice as a new type of event staff provider launches in London. Retrieved October 20, 2013 from http://www.eventindustrynews.co.uk/2013/03/20/event-planners-rejoiceas-a-new-type-of-event-staff-provider-launches-in-london/ .
* Freidson, E. (2007). Professionalism the third logic. Cambridge: Polity press in association with Blackwell publishers.
* East Anglia Childrens Hospice. (2013). Overview of EACH. Retrieved October 19, 2013 from http://www.each.org.uk/about\_us/overview\_of\_each .
* Lloyds Banking Group. (2012). The economic impact of the London 2012 Olympics and Paralympics games. Retrieved October 19, 2013
* From http://www.lloydsbankinggroup.com/media/pdfs/lbg/2012/Eco\_impact\_ report.pdf

### Websites:

### [1]. [http://en.wikipedia.org/wiki/ JAVA](http://en.wikipedia.org/wiki/%20JAVA) for JAVA

### [2].https://www.javapoint.com/java-tutorial/gor java

### [3].https://www.geeksforgeeks.org/lambda testing

### [4]. [http://w3schools.com/](http://w3schools.com/%20) for information on HTML.

[5].https://www.youtube.com/@freecodecamp. [6].https://codewithcurious.com/AndriodStudio.