

**SOCIAL COP**  
**A MINOR PROJECT I REPORT**

*Submitted by*

<b>THARUN RAJEEV</b>	<b>REG NO : RA1711008020056</b>
<b>MANOJ KUMAR</b>	<b>REG NO : RA17111008020054</b>
<b>ABHISHEK SOPANNA</b>	<b>REG NO : RA17110080200</b>

Under the guidance of

**MRS.LIZA**

**(Assistant Professor, Department of Information Technology)**

*in partial fulfilment for the award of the degree*

*of*

**BACHELOR OF TECHNOLOGY**

*in*

**INFORMATION TECHNOLOGY**



**FACULTY OF ENGINEERING AND TECHNOLOGY**  
**SRM INSTITUTE OF SCIENCE AND TECHNOLOGY**  
**RAMAPURAM CAMPUS, CHENNAI -600089**  
**NOVEMBER 2019**  
**SRM INSTITUTE OF SCIENCE AND TECHNOLOGY**  
**(Deemed to be University U/S 3 of UGC Act, 1956)**

## **BONAFIDE CERTIFICATE**

Certified that this project report “SOCIAL-COP” is the bonafide work of **THARUN RAJEEV (RA1711008020056)**, **MANOJ KUMAR (RA17111008020054)**, and **ABHISHEK SOPANNA (RA17110080200)** who carried out the project work under my supervision.

**MRS LIZA**

**ASSISTANT PROFESSOR**

**MRS R.M. RANI**

**HEAD OF THE DEPARTMENT**

Information Technology,

SRM Institute of Science and Technology

Information Technology,

SRM Institute of Science and Technology,

Ramapuram Campus, Chennai.

Ramapuram Campus, Chennai

Submitted for the Viva Voce Examination held on ..... at SRM Institute of Science and Technology, Ramapuram Campus, Chennai- 600089.

**INTERNAL EXAMINER I**

**INTERNAL EXAMINER II**

## **ABSTRACT**

## Social Cop

Social Cop application is a mobile-based solution to address the needs of removal of daily traffic chaos. Every day we face many issues like a traffic jam, illegal parking, road rages, accidents etc. These can be avoided and prevented to a larger extent if and only if they are reported on time to concerned authorities. The government officials as well as the citizen sign up using their Email ID where a feed is generated containing complaints by the people. They upload the image of their problem and write a short description about it along with it the problem type. The request to solve can be sent by the people as well as well the government.

In short can create a kind of competition to solve the issue between people and the government.

## **TABLE OF CONTENT**

<b>Chapter no.</b>	<b>Title</b>	<b>Page no.</b>
<b>1</b>	<b>Introduction</b>	
1.1	Introduction	
1.2	Overview	
1.3	API Modelling	
1.4	Authentication	
<b>2</b>	<b>Literature Survey</b>	
2.1	Existing System	
2.2	Proposed System	
<b>3</b>	<b>System Design</b>	
3.1	System Architecture	
3.2	Modules	
<b>4</b>	<b>System Requirements</b>	
4.1	Hardware Requirements	
4.2	System Requirements	
4.3	Backend	

**5      Methodology**

5.1            Android Studio

5.2            MongoDB

5.3            Implementation

**6      Database Design and tables**

**7      Screenshots**

**8      Testing**

**9      Conclusion and Future Enhancement**

**10     Reference**

**CHAPTER 1**

# INTRODUCTION

## 1.1 INTRODUCTION

Social cop app basically acts as a portal for people to lodge their complaints

Against basic traffic problems in the locality. Road accidents because of potholes, Traffic jams caused at roads where the cops are not present or not around, these problems are faced by lot of people until it gets solved or resolved.

These incidents including garbage on road, reducing the space to move, Faulty signals that creating a road chaos and damaged signal board that cause wrong information leading to parking at No parking thus causing traffic, such issues cops are needed to solve the problem. will be notified on the feed page that could be seen by the local users as well as the police officials.

The officials register using their Gmail as the Locals do to register, if the government officials take time to respond and act on the problem, there is a way for the locals to solve the issue if they have resources enough to resolve it , they can send a request to the profile who uploaded the issue.

Further facilities of commenting, upvoting, plusones can be done on a post.

## 1.2 Overview

Retrofit is a type-safe HTTP client for Android and Java – developed by Square ([Dagger](#), [Okhttp](#)).

In this article, we're going to explain how to use Retrofit, with a focus on its most interesting features. More notably we'll discuss the synchronous and asynchronous API, how to use it with authentication, logging, and some good modeling practices.

### 1.2.1 Setting up the Example

We'll start by adding the Retrofit library and the Gson converter:

```
<dependency>
    <groupId>com.squareup.retrofit2</groupId>
    <artifactId>retrofit</artifactId>
    <version>2.3.0</version>
```

```
3    </dependency>
4
5    <dependency>
6        <groupId>com.squareup.retrofit2</groupId>
7        <artifactId>converter-gson</artifactId>
8        <version>2.3.0</version>
9
10
```

For the latest versions, have a look at [Retrofit](#) and converter-gson on Maven Central repository.

## 1.3 API Modeling

Retrofit models REST endpoints as Java interfaces, making them very simple to understand and consume.

We'll model the user API from GitHub; this has a *GET* endpoint that returns this in JSON format:

```
1  {
2      login: "mojombo",
3      id: 1,
4      url: "https://api.github.com/users/mojombo",
5      ...
6  }
```

Retrofit works by modeling over a base URL and by making interfaces return the entities from the REST endpoint.

### 1.3.1 Synchronous/Asynchronous API

To construct an HTTP request call, we need to build our Retrofit object first:

```
OkHttpClient.Builder httpClient = new OkHttpClient.Builder();
Retrofit retrofit = new Retrofit.Builder()
1   .baseUrl("https://api.github.com/")
2   .addConverterFactory(GsonConverterFactory.create())
3   .client(httpClient.build())
```

```
4     .build();  
5  
6
```

Retrofit provides a convenient builder for constructing our required object. **It needs the base URL which is going to be used for every service call and a converter factory** – which takes care of the parsing of data we're sending and also the responses we get.

In this example, we're going to use the *GsonConverterFactory*, which is going to map our JSON data to the *User* class we defined earlier.

It's important to note that different factories serve different purposes, so keep in mind that we can also use factories for XML, proto-buffers or even create one for a custom protocol. For a list of already implemented factories, we can have a look [here](#).

The last dependency is *OKHttpClient* – which is an HTTP & HTTP/2 client for Android and Java applications. This is going to take care of connecting to the server and the sending and retrieval of information. We could also add headers and interceptors for every call, which we're going to see in our authentication section

### 1.3.2 Making a Reusable *ServiceGenerator* Class

Now that we saw how to construct our Retrofit object and how to consume an API, we can see that we don't want to keep writing the builder over and over again.

What we want is a reusable class that allows us to create this object once and reuse it for the lifetime of our application:

```
public class GitHubServiceGenerator {  
  
    1     private static final String BASE_URL = "https://api.github.com/";  
    2  
    3     private static Retrofit.Builder builder  
    4         = new Retrofit.Builder()  
    5             .baseUrl(BASE_URL)  
    6             .addConverterFactory(GsonConverterFactory.create());  
    7  
    8     private static Retrofit retrofit = builder.build();  
    9
```

```

10     private static OkHttpClient.Builder httpClient
11         = new OkHttpClient.Builder();
12
13     public static <S> S createService(Class<S> serviceClass) {
14         return retrofit.create(serviceClass);
15     }
}

```

All the logic of creating the Retrofit object is now moved to this *GitHubServiceGenerator* class, this makes it a sustainable client class which stops the code from repeating.

## 1.4 Authentication

Most APIs have some authentication to secure access to it.

Taking into account our previous generator class, we're going to add a create service method, that takes a JWT token with the *Authorization* header :

```

1     public static <S> S createService(Class<S> serviceClass, final String token ) {
2         if( token != null ) {
3             httpClient.interceptors().clear();
4             httpClient.addInterceptor( chain -> {
5                 Request original = chain.request();
6                 Request request = original.newBuilder()
7                     .header("Authorization", token)
8                     .build();
9                 return chain.proceed(request);
10            });
11            builder.client(httpClient.build());
12            retrofit = builder.build();
13        }
14        return retrofit.create(serviceClass);
15    }
}

```

To add a header to our request, we need to use the interceptor capabilities of *OkHttp*; we do this by using our previously defined builder and by reconstructing the Retrofit object.

Note that this is a simple auth example, but with the use of interceptors we can use any authentication such as OAuth, user/password, etc.

#### 1.4.1 Auth0 Overview

Auth0 provides authentication and authorization as a service. We are here to give developers and companies the building blocks they need to secure their applications without having to become security experts. You can connect any application (written in any language or on any stack) to Auth0 and define the identity providers you want to use (how you want your users to log in). Based on your app's technology, choose one of our SDKs (or call our API), and hook it up to your app. Now each time a user tries to authenticate, Auth0 will verify their identity and send the required information back to your app.

#### Why use Auth0?

Take a look at just a few of Auth0's use cases:

- You built an awesome app and you want to add user authentication and authorization. Your users should be able to log in either with username/password or with their social accounts (such as Facebook or Twitter). You want to retrieve the user's profile after the login so you can customize the UI and apply your authorization policies.
- You built an API and you want to secure it with [OAuth 2.0](#).
- You have more than one app, and you want to implement Single Sign-on (SSO).
- You built a JavaScript front-end app and a mobile app, and you want them both to securely access your API.
- You have a web app which needs to authenticate users using Security Assertion Markup Language (SAML).
- You believe passwords are broken and you want your users to log in with one-time codes delivered by email or SMS.
- If one of your user's email addresses is compromised in some site's public data breach, you want to be notified, and you want to notify the users and/or block them from logging in to your app until they reset their password.
- You want to act proactively to block suspicious IP addresses if they make consecutive failed login attempts, in order to avoid DDoS attacks.

- You are part of a large organization who wants to federate their existing enterprise directory service to allow employees to log in to the various internal and third-party applications using their existing enterprise credentials.
- You don't want (or you don't know how) to implement your own user management solution. Password resets, creating, provisioning, blocking, and deleting users, and the UI to manage all these. You just want to focus on your app.
- You want to enforce multi-factor authentication (MFA) when your users want to access sensitive data.
- You are looking for an identity solution that will help you stay on top of the constantly growing compliance requirements of SOC2, GDPR, PCI DSS, HIPAA, and others.
- You want to use analytics to track users on your site or application. You plan on using this data to create funnels, measure user retention, and improve your sign-up flow.

## Which industry standards does Auth0 use?

Once upon a time, when computers were standalone systems, all the authentication and user data lived in a single machine. Times have changed, and now you can use the same login information across multiple apps and sites. This has been achieved through widespread adoption of identity industry standards across the web.

These are a set of open specifications and protocols that specify how to design an authentication and authorization system. They specify how you should manage identity, move personal data securely, and decide who can access applications and data.

The identity industry standards that we use here in Auth0 are:

1. **Open Authorization (OAuth) 1:** the original standard for access delegation. Used as a way for a user to grant websites access to their information on other websites or apps, but without giving them the credentials.
2. **Open Authorization (OAuth) 2:** an authorization standard that allows a user to grant limited access to their resources on one site, to another site, without having to expose their credentials. You use this standard every time you log in to a site using your Google account and you are asked if you agree with sharing your email address and your contacts list with that site.
3. **OpenID Connect (OIDC):** an identity layer that sits on top of OAuth 2 and allows for easy verification of the user's identity, as well the ability to get basic profile information from the identity provider.

4. **JSON Web Tokens (JWT)**: an open standard that defines a compact and self-contained way for securely transmitting information between parties as a JSON object.
5. **Security Assertion Markup Language (SAML)**: an open-standard, XML-based data format that allows businesses to communicate user authentication and authorization information to partner companies and enterprise applications their employees may use.
6. **WS-Federation (WS-Fed)**: a standard developed by Microsoft, and used extensively in their applications. It defines the way security tokens can be transported between different entities to exchange identity and authorization information.

## CHAPTER 2

### LITERATURE SURVEY

## **2.1 EXISTING SYSTEM**

A mobile based solution to address the needs of removal of daily traffic chaos. Every day we face many issues like a traffic jam, illegal parking, road rages, accidents etc. These can be avoided and prevented to a larger extent if and only if they are reported on time to concerned authorities.

### **2.1.1 Features:**

The application can determine the location of the user very accurately and saves it in the database, so that police, respected team can immediately go there.

The app has the real-time database, every time when a police(admin) goes to see the complaints, he always sees the new ones i.e. recently updated.

It identifies the user by email id, so that false complaints and misuse of the app can be prevented from non-beneficial people.

### **2.1.2 Uses:**

- Easy reporting of faults
- More accurate information
- Less corruption
- Happy people

## **2.2 PROPOSED SYSTEM**

This is a mobile based application to address the need of removal of daily traffic problem. Everyday we face many issues like a traffic jam, illegal parking, road rages, accidents, path holes etc. These can be avoided and prevented to a larger extent if and only if they are reported on time to concerned authorities.

It identifies the user by gmail id, username where once registered data will be saved in the database so that false complaints and misuse of the app can be prevented from non-beneficial people.

The app can be used only when the gps is on to locate the exact location of the problem risen.

The difference between existing system is here we made some extra features which makes interface more smoother and easy to use ans some updates which helps to understand the problem and escalate it to the department which it is related.

Here if the officials are taking time to resolve the issue then basically the locals viewing the issue can also resolve the problem and get recognized as they did it before the government. This may create a competition between the people and government and capture user feelings using upvotes, plus ones and comments

### **2.2.1 Features:**

Some features which are add additionally are

- Here we can share the problem where people can share the issue which they already faced so that the problem will be in top of the app which is mostly shared and highlighted
- People can upvote the issue which they see on the app which they seem they also faced the same issue and can be shared.
- Comment section will be enabled to all the posts so that people can talk about the issue and share their views about the issue.

## **CHAPTER 3**

### **SYSTEM DESIGN**

### **3.1 System Architecture**

The system functions on android studio with a posting request and resolve request architecture considering all posts comments and requests as posts where comments can be commented. The user has to sign up first. Followed by login in to the portal. The portal consist of three key fragments. These fragments are held together by the navigation tool bar.

The first page is the posts recycler view that contains all the complaints posted by the people. The location, picture, a description, and a complaint tag is been put in as input details and the output is a feed data with picture tags location string text .followed by the write page where the user is allowed to launch his phone camera and is allowed to click a proof of the complaint along with the problem tag that the user has to select from a category list provided. With the description and a location the person can successfully post on to the recycler view.

This place the posts are seen with the user name and his profile picture to update which is not yet provided and can be done with time as not much of purpose.

The last fragment actually the page that shows all the posts posted by the concerned person. Actually a profile page with his name and profile picture along with a list or recycler view of his posts of complaints

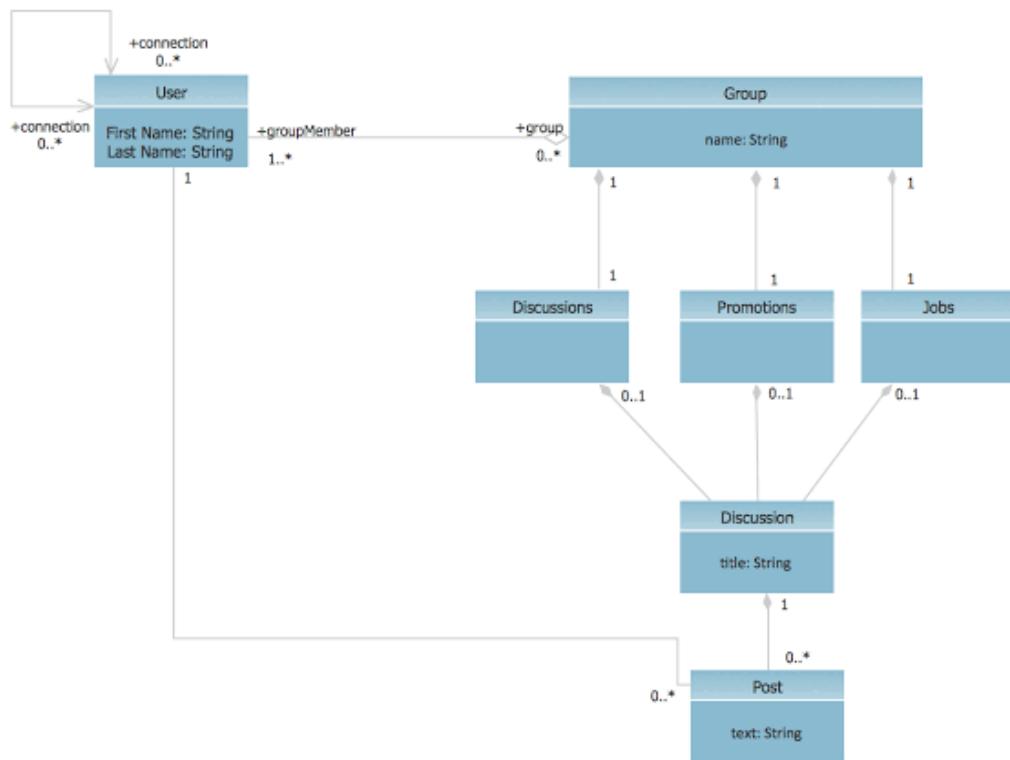
### **3.2 Modules**

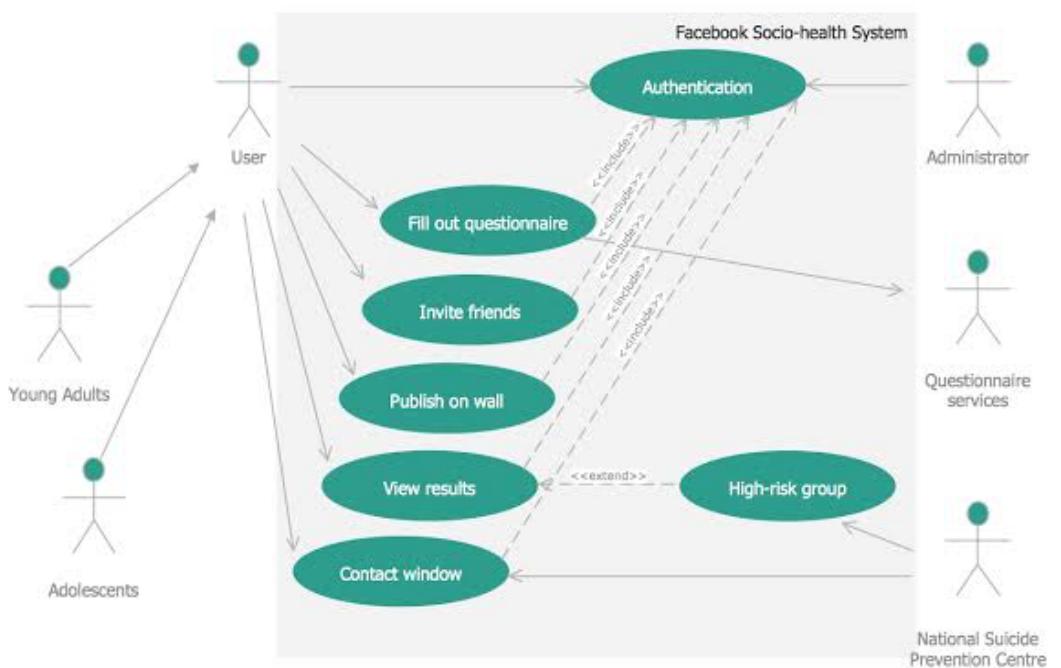
System Modules (originally known as System Building Blocks; the name was changed around 1961) are a DEC modular digital logic family which preceded the later FLIP CHIPS. They connect to the units they are plugged into via a set of 22 gold-plated discrete pins along one edge.

They use transistor inverter circuits, with the transistors operating saturated, to avoid dependence on tight tolerances; they use -3V and 0V as logic levels. Intended for prototyping as well as production, they include design features intended to avoid damage. They are provided with design advice which includes loading rules and wiring instructions.

System Modules (originally known as System Building Blocks; the name was changed around 1961) are a DEC modular digital logic family which preceded the later FLIP CHIPs. They connect to the units they are plugged into via a set of 22 gold-plated discrete pins along one edge.

They use transistor inverter circuits, with the transistors operating saturated, to avoid dependence on tight tolerances; they use -3V and 0V as logic levels. Intended for prototyping as well as production, they include design features intended to avoid damage. They are provided with design advice which includes loading rules and wiring instructions.





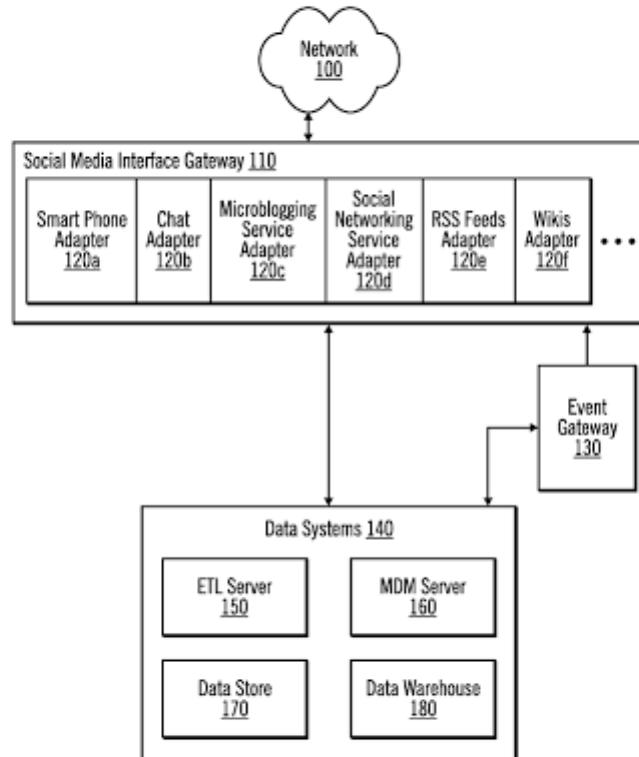


FIG. 1

## **CHAPTER 4**

### **SYSTEM REQUIREMENTS**

#### **4.1 Hardware Requirements:**

1. Smartphone
2. Camera
3. Active mobile network

#### **4.2 Software Requirements:**

1. Android version 4.2 or higher
2. Programming language: Java
3. Database : Mongodb

#### **4.3 Backend:**

1. My SQL
2. Retrofit
3. auth0

## CHAPTER 5

### METHODOLOGY

#### 5.1 Android Studio

Android studio is based on IntelliJ IDEA, which does all the functionality that Eclipse with ADT plug-in do, with lot more additional features. The initial version of android studio offers

1. Gradle-based build support
2. and quick fixes
3. Lint tools to catch performance, usability, version compatibility and other problems
4. ProGuard and app-signing capabilities
5. Template-based wizards to create common Android designs and components
6. **A rich layout editor:** it allows you to drag-and-drop UI components, preview layouts on multiple screen configurations. Preview appears instantly as you change in the layout editor. You can choose a language, and can see the preview of layout with that
7. **Rich Color Preview editor:** While adding colors as a resource, and we can see the color preview at the left hand side of the editor.
8. **Deep Code Analysis:** If you point to a line and it gives detailed explanation about an exception based on the annotation added. And you can also know which constants are allowed for which API. It also has the powerful code completion. You can also inspect code in whole project, IntelliJ lists all Lint errors during code inspection

#### What is XML?

1. XML (eXtensible Markup Language) is a mark up language.
2. XML is designed to store and transport data.
3. Xml was released in late 90's. it was created to provide an easy to use and store self describing data.

4. XML became a W3C Recommendation on February 10, 1998.
5. XML is not a replacement for HTML.
6. XML is designed to be self-descriptive.
7. XML is designed to carry data, not to display data.
8. XML tags are not predefined. You must define your own tags.
9. XML is platform independent and language independent.

## **What is Mark-Up Language?**

A **Mark Up Language** is a modern system for highlight or underline a document.

Students often underline or highlight a passage to revise easily, same in the sense of modern mark up language highlighting or underlining is replaced by tags.

## **5.2 MongoDB**

MongoDB is an object-oriented, simple, dynamic, and scalable NoSQL database. It is based on the NoSQL document store model. The data objects are stored as separate documents inside a collection — instead of storing the data into the columns and rows of a traditional relational database. The motivation of the MongoDB language is to implement a data store that provides high performance, high availability, and automatic scaling. MongoDB is extremely simple to install and implement. MongoDB uses JSON or BSON documents to store data. General distributions for MongoDB support Windows, Linux, Mac OS X, and Solaris.

### **5.2.1 Making the Choice**

Of course, your choice of database is always a decision based on pros and cons.

Pros

- Document oriented
- High performance
- High availability — Replication
- High scalability – Sharding
- Dynamic — No rigid schema.

- Flexible – field addition/deletion have less or no impact on the application
- Heterogeneous Data
- No Joins
- Distributed
- Data Representation in JSON or BSON
- Geospatial support
- Easy Integration with BigData Hadoop
- Document-based query language that's nearly as powerful as SQL
- Cloud distributions such as AWS, Microsoft, RedHat, dotCloud and SoftLayer etc:-. In fact, MongoDB is built for the cloud. Its native scale-out architecture, enabled by ‘sharding,’ aligns well with the horizontal scaling and agility afforded by cloud computing.

Cons

- A downside of NoSQL is that most solutions are not as strongly ACID-compliant (Atomic, Consistency, Isolation, Durability) as the more well-established RDBMS systems.
- Complex transaction
- No function or stored procedure exists where you can bind the logic

### **5.3 Implementation**

Good For:

- E-commerce product catalog.
- Blogs and content management.
- Real-time analytics and high-speed logging, caching, and high scalability.
- Configuration management.
- Maintaining location-based data — Geospatial data.
- Mobile and social networking sites.
- Evolving data requirements.
- Loosely coupled objectives — the design may change over time.

### **Not so Good For:**

- Highly transactional systems or where the data model is designed up front.
- Tightly coupled systems

And there you have it! Now, you've got a quick and easy overview of how MongoDB works, some use cases where it can shine, and how it relates to SQL technology.

## **CHAPTER 6**

### **DATABASE DESIGN**

Database design is the organization of data according to a database model. The designer determines what data must be stored and how the data elements interrelate. With this information, they can begin to fit the data to the database model.

Database design involves classifying data and identifying interrelationships. This theoretical representation of the data is called an ontology. The ontology is the theory behind the database's design.

In a majority of cases, a person who is doing the design of a database is a person with expertise in the area of database design, rather than expertise in the domain from which the data to be stored is drawn e.g. financial information, biological information etc. Therefore, the data to be stored in the database must be determined in cooperation with a person who does have expertise in that domain, and who is aware of what data must be stored within the system.

This process is one which is generally considered part of requirements analysis, and requires skill on the part of the database designer to elicit the needed information from those with the domain knowledge. This is because those with the necessary domain knowledge frequently cannot express clearly what their system requirements for the database are as they are unaccustomed to thinking in terms of the discrete data elements which must be stored. Data to be stored can be determined by Requirement Specification.

Once a database designer is aware of the data which is to be stored within the database, they must then determine where dependency is within the data. Sometimes when data is changed you can be changing other data that is not visible. For example, in a list of names and addresses, assuming a situation where multiple people can have the same address, but one person cannot have more than one address, the address is dependent upon the name. When provided a name and the list the address can be uniquely determined; however, the inverse does not hold - when given an address and the list, a name cannot be uniquely determined

because multiple people can reside at an address. Because an address is determined by a name, an address is considered dependent on a name.

(NOTE: A common misconception is that the relational model is so called because of the stating of relationships between data elements therein. This is not true. The relational model is so named because it is based upon the mathematical structures known as relations.)

In the field of relational database design, normalization is a systematic way of ensuring that a database structure is suitable for general-purpose querying and free of certain undesirable characteristics—insertion, update, and deletion anomalies that could lead to loss of data integrity.

A standard piece of database design guidance is that the designer should create a fully normalized design; selective denormalization can subsequently be performed, but only for performance reasons. The trade-off is storage space vs performance. The more normalized the design is, the less data redundancy there is (and therefore, it takes up less space to store), however, common data retrieval patterns may now need complex joins, merges, and sorts to occur - which takes up more data read, and compute cycles. Some modeling disciplines, such as the dimensional modeling approach to data warehouse design, explicitly recommend non-normalized designs, i.e. designs that in large part do not adhere to 3NF. Normalization consists of normal forms that are 1NF,2NF,3NF,BOYCE-CODD NF (3.5NF),4NF and 5NF

Document databases take a different approach. A document that is stored in such a database, typically would contain more than one normalized data unit and often the relationships between the units as well. If all the data units and the relationships in question are often retrieved together, then this approach optimizes the number of retrieves. It also simplifies how data gets replicated, because now there is a clearly identifiable unit of data whose consistency is self-contained. Another consideration is that reading and writing a single document in such databases will require a single transaction - which can be an important consideration in a Microservices architecture. In such situations, often, portions of the document are retrieved from other services via an API and stored locally for efficiency reasons. If the data units were to be split out across the services, then a read (or write) to

support a service consumer might require more than one service calls, and this could result in management of multiple transactions, which may not be preferred.

MongoDB is an object-oriented, simple, dynamic, and scalable NoSQL database. It is based on the NoSQL document store model. The data objects are stored as separate documents inside a collection — instead of storing the data into the columns and rows of a traditional relational database. The motivation of the MongoDB language is to implement a data store that provides high performance, high availability, and automatic scaling. MongoDB is extremely simple to install and implement. MongoDB uses JSON or BSON documents to store data. General distributions for MongoDB support Windows, Linux, Mac OS X, and Solaris.

The system functions on android studio with a posting request and resolve request architecture considering all posts comments and requests as posts where comments can be commented. The user has to sign up first. Followed by login in to the portal. The portal consist of three key fragments. These fragments are held together by the navigation tool bar.

The physical design of the database specifies the physical configuration of the database on the storage media. This includes detailed specification of data elements, data types, indexing options and other parameters residing in the DBMS data dictionary. It is the detailed design of a system that includes modules & the database's hardware & software specifications of the system. Some aspects that are addressed at the physical layer:

Security - end-user, as well as administrative security.

Replication - what pieces of data get copied over into another database, and how often. Are there multiple-masters, or a single one?

High-availability - whether the configuration is active-passive, or active-active, the topology, coordination scheme, reliability targets, etc all have to be defined.

Partitioning - if the database is distributed, then for a single entity, how is the data distributed amongst all the partitions of the database, and how is partition failure taken into account.

Backup and restore schemes.

At the application level, other aspects of the physical design can include the need to define stored procedures, or materialized query views, OLAP cubes, etc.

7:00 43%

cloud.mongodb.com/v2/5d9

Clusters SELF > PROJECT 0 > CLUSTERS Cluster0 VERSION 4.0.12 REGION

Overview Real Time Metrics Collections Profiler Performance Advisor Comm

DATABASES: 1 COLLECTIONS: 5 + Create Database

NAMESPACES

sociocop plusones

media posts upvotes users

sociocop.plusones

COLLECTION SIZE: TOTAL INDEXES TOTAL SIZE:

197B DOCUMENTS: 36KB 3

Find Indexes Aggregation

INSERT DOCUMENT

FILTER {"filter": "example"} Find Reset

QUERY RESULTS 1-3 OF 3

```
_id: ObjectId("5d9e088436c5171db4ded095")
pid: "5nhzp7kk1jh2efn"
uid: "1234"
```

```
_id: ObjectId("5d9e0995df25341f982fb5da")
pid: "5nhzp68ok1jhfvq7"
uid: "1234"
```

```
_id: ObjectId("5d9f8ef60dd23119686cab9a")
pid: "5nhzp50ok14sebs"
uid: "5nhzp6csk1kvuv8g"
```

Figure 1.1 xyz

7:00

cloud.mongodb.com/v2/5d9

Please set your time zone Usage This Month: \$0.00 / \$0.00

**Clusters**

**Cluster0**

Overview Real Time Metrics Collections Profiler Performance Advisor Comm

**DATABASES: 1 COLLECTIONS: 5**

+ Create Database NAMESPACES

- sociocop
  - media
  - pluses
  - posts
  - upvotes
  - users

**sociocop.media**

COLLECTION SIZE: 7.29KB TOTAL DOCUMENTS: 29 INDEXES TOTAL SIZE: 36KB

Find Indexes Aggregation

INSERT DOCUMENT Find Reset

FILTER <"filter": "example">

QUERY RESULTS 1-20 OF MANY

```

  {
    "_id": "ObjectID:5d9e3a76f49a4b0e1c27de22",
    "id": "Shnpzpack177e00",
    "url": "https://i.imgur.com/12345678901234567890.jpg",
    "type": "image"
  }

  {
    "_id": "ObjectID:5d9e3a76f49a4b0e1c27de23",
    "id": "Shnpzpack177e00",
    "url": "https://i.imgur.com/12345678901234567890.jpg",
    "type": "image"
  }

  {
    "_id": "ObjectID:5d9e3a76f49a4b0e1c27de24",
    "id": "Shnpzpack177e00",
    "url": "https://i.imgur.com/12345678901234567890.jpg",
    "type": "image"
  }

  {
    "_id": "ObjectID:5d9e3a76f49a4b0e1c27de25",
    "id": "Shnpzpack177e00",
    "url": "https://i.imgur.com/12345678901234567890.jpg",
    "type": "image"
  }

  {
    "_id": "ObjectID:5d9e3a76f49a4b0e1c27de26",
    "id": "Shnpzpack177e00",
    "url": "https://i.imgur.com/12345678901234567890.jpg",
    "type": "image"
  }

  {
    "_id": "ObjectID:5d9e3a76f49a4b0e1c27de27",
    "id": "Shnpzpack177e00",
    "url": "https://i.imgur.com/12345678901234567890.jpg",
    "type": "image"
  }

  {
    "_id": "ObjectID:5d9e3a76f49a4b0e1c27de28",
    "id": "Shnpzpack177e00",
    "url": "https://i.imgur.com/12345678901234567890.jpg",
    "type": "image"
  }

  {
    "_id": "ObjectID:5d9e3a76f49a4b0e1c27de29",
    "id": "Shnpzpack177e00",
    "url": "https://i.imgur.com/12345678901234567890.jpg",
    "type": "image"
  }

  {
    "_id": "ObjectID:5d9e3a76f49a4b0e1c27de2a",
    "id": "Shnpzpack177e00",
    "url": "https://i.imgur.com/12345678901234567890.jpg",
    "type": "image"
  }

  {
    "_id": "ObjectID:5d9e3a76f49a4b0e1c27de2b",
    "id": "Shnpzpack177e00",
    "url": "https://i.imgur.com/12345678901234567890.jpg",
    "type": "image"
  }

  {
    "_id": "ObjectID:5d9e3a76f49a4b0e1c27de2c",
    "id": "Shnpzpack177e00",
    "url": "https://i.imgur.com/12345678901234567890.jpg",
    "type": "image"
  }

  {
    "_id": "ObjectID:5d9e3a76f49a4b0e1c27de2d",
    "id": "Shnpzpack177e00",
    "url": "https://i.imgur.com/12345678901234567890.jpg",
    "type": "image"
  }

  {
    "_id": "ObjectID:5d9e3a76f49a4b0e1c27de2e",
    "id": "Shnpzpack177e00",
    "url": "https://i.imgur.com/12345678901234567890.jpg",
    "type": "image"
  }

  {
    "_id": "ObjectID:5d9e3a76f49a4b0e1c27de2f",
    "id": "Shnpzpack177e00",
    "url": "https://i.imgur.com/12345678901234567890.jpg",
    "type": "image"
  }

  {
    "_id": "ObjectID:5d9e3a76f49a4b0e1c27de2g",
    "id": "Shnpzpack177e00",
    "url": "https://i.imgur.com/12345678901234567890.jpg",
    "type": "image"
  }

  {
    "_id": "ObjectID:5d9e3a76f49a4b0e1c27de2h",
    "id": "Shnpzpack177e00",
    "url": "https://i.imgur.com/12345678901234567890.jpg",
    "type": "image"
  }

  {
    "_id": "ObjectID:5d9e3a76f49a4b0e1c27de2i",
    "id": "Shnpzpack177e00",
    "url": "https://i.imgur.com/12345678901234567890.jpg",
    "type": "image"
  }

  {
    "_id": "ObjectID:5d9e3a76f49a4b0e1c27de2j",
    "id": "Shnpzpack177e00",
    "url": "https://i.imgur.com/12345678901234567890.jpg",
    "type": "image"
  }

  {
    "_id": "ObjectID:5d9e3a76f49a4b0e1c27de2k",
    "id": "Shnpzpack177e00",
    "url": "https://i.imgur.com/12345678901234567890.jpg",
    "type": "image"
  }

  {
    "_id": "ObjectID:5d9e3a76f49a4b0e1c27de2l",
    "id": "Shnpzpack177e00",
    "url": "https://i.imgur.com/12345678901234567890.jpg",
    "type": "image"
  }

```

PREVIOUS 1-20 of many results NEXT >

System Status: All Good Last Logon: 48.202.1.23.14:8020 MongoDB Version: 4.0.12 Status Terms Privacy Other Help Contact Sales

Figure 1.5 xyz

SELF > PROJECT 0 > CLUSTERS

## Cluster0

- Overview
- Real Time
- Metrics
- Collections**
- Profiler
- Performance Advisor
- Comm

VERSION 4.0.12 REGION

DATABASES: 1 COLLECTIONS: 5

+ Create Database

Namespaces

sociocop

media

placonc

**posts**

upvotes

users

**sociocop.posts**

COLLECTION SIZE: 2.72KB DOCUMENTS: 14 INDEXES TOTAL SIZE: 30KB

Find Indexes Aggregation

FILTER: {"filter": "example"} **Find** Reset

QUERY RESULTS 1-14 OF 14

```

_id: ObjectId("5d9e125f0239417022b0e7")
parentId: null
pid: "https://sociocop.com"
uid: "5d9e125f0239417022b0e7"
description: "post"
image: "https://sociycop.com/api/post/5d9e125f0239417022b0e7"
category: null
location: null
type: "post"
status: null

_id: ObjectId("5d9e125f0239417022b0e8")
parentId: null
pid: "https://sociycop.com"
uid: "5d9e125f0239417022b0e8"
description: "first post"
image: "https://sociycop.com/api/post/5d9e125f0239417022b0e8"
category: null
location: null
type: "post"
status: null
accepted: true

_id: ObjectId("5d9e125f0239417022b0e9")
parentId: null
pid: "https://sociycop.com"
uid: "5d9e125f0239417022b0e9"
description: "2nd post"
category: "post"
location: "Barcelona"
type: "post"
status: null

_id: ObjectId("5d9e125f0239417022b0ea")
parentId: null
pid: "https://sociycop.com"
uid: "5d9e125f0239417022b0ea"
description: "3rd post"
category: "post"
location: null
type: "post"
status: null

_id: ObjectId("5d9e125f0239417022b0eb")
parentId: null
pid: "https://sociycop.com"
uid: "5d9e125f0239417022b0eb"
description: "4th post"
category: "post"
location: null
type: "post"
status: null

_id: ObjectId("5d9e36c757346b0190049c0e")
parentId: null
pid: "https://sociycop.com"
uid: "5d9e36c757346b0190049c0e"
description: "New user"
image: "https://sociycop.com/api/post/5d9e36c757346b0190049c0e"
category: null
location: null
type: "post"
status: null

_id: ObjectId("5d9e710155b204901fdd0e00")
parentId: null
pid: "https://sociycop.com"
uid: "5d9e710155b204901fdd0e00"
description: "new"
image: "https://sociycop.com/api/post/5d9e710155b204901fdd0e00"
location: null
type: "post"
status: null
createdat: 157064906292

_id: ObjectId("5d9e710155b204901fdd0e01")
parentId: null
pid: "https://sociycop.com"
uid: "5d9e710155b204901fdd0e01"
description: "new"
image: "https://sociycop.com/api/post/5d9e710155b204901fdd0e01"
location: null
type: "post"
status: null
createdat: 157064906294

_id: ObjectId("5d9e710155b204901fdd0e02")
parentId: null
pid: "https://sociycop.com"
uid: "5d9e710155b204901fdd0e02"
description: "new"
image: "https://sociycop.com/api/post/5d9e710155b204901fdd0e02"
location: null
type: "post"
status: null
createdat: 157064906295

_id: ObjectId("5d9e710155b204901fdd0e03")
parentId: null
pid: "https://sociycop.com"
uid: "5d9e710155b204901fdd0e03"
description: "new"
image: "https://sociycop.com/api/post/5d9e710155b204901fdd0e03"
location: null
type: "post"
status: null
createdat: 157064906296

_id: ObjectId("5d9e710155b204901fdd0e04")
parentId: null
pid: "https://sociycop.com"
uid: "5d9e710155b204901fdd0e04"
description: "new"
image: "https://sociycop.com/api/post/5d9e710155b204901fdd0e04"
location: null
type: "post"
status: null
createdat: 157064906298

_id: ObjectId("5d9e710155b204901fdd0e05")
parentId: null
pid: "https://sociycop.com"
uid: "5d9e710155b204901fdd0e05"
description: "new"
image: "https://sociycop.com/api/post/5d9e710155b204901fdd0e05"
location: null
type: "post"
status: null
createdat: 157064906300

_id: ObjectId("5d9e710155b204901fdd0e06")
parentId: null
pid: "https://sociycop.com"
uid: "5d9e710155b204901fdd0e06"
description: "new"
image: "https://sociycop.com/api/post/5d9e710155b204901fdd0e06"
location: null
type: "post"
status: null
createdat: 157064906302

_id: ObjectId("5d9e710155b204901fdd0e07")
parentId: null
pid: "https://sociycop.com"
uid: "5d9e710155b204901fdd0e07"
description: "new"
image: "https://sociycop.com/api/post/5d9e710155b204901fdd0e07"
location: null
type: "post"
status: null
createdat: 157064906304

```

Clusters

SELF > PROJECT 0 > CLUSTERS

**Cluster0**

VERSION 4.0.12 REGION

Overview Real Time Metrics Collections Profiler Performance Advisor Comm.

DATABASES: 1 COLLECTIONS: 5

+ Create Database

NAMESPACES

sociocop

- media
- plusones
- posts

upvotes

- users

**sociocop.upvotes**

COLLECTION SIZE: TOTAL DOCUMENTS: INDEXES TOTAL SIZE:

61B 1 36KB

Find Indexes Aggregation

INSERT DOCUMENT

FILTER {"filter": "example"}

Find Reset

QUERY RESULTS 1-1 OF 1

```
_id: ObjectId("5d9e087c36c5171db4ded094")
pid: "5nhzp7kk1jh2efn"
uid: "1234"
```

The screenshot shows the MongoDB Compass application interface. At the top, it says 'Clusters' and 'Please set your time zone'. It's connected to 'SELF > PROJECT 0 > CLUSTERS' with 'Cluster0'. The 'VERSION' is 4.0.12 and 'REGION' is listed. Below this is a navigation bar with tabs: Overview, Real Time, Metrics, Collections (which is underlined in green), Profiler, Performance Advisor, and Comm. Under 'Collections', it shows 'DATABASES: 1' and 'COLLECTIONS: 5'. A button '+ Create Database' is available. On the left, there's a sidebar for 'NAMESPACES' with 'sociocop' expanded, showing 'media', 'plusones', 'posts', and 'upvotes' (which is also expanded, showing 'users'). The main area is titled 'sociocop.upvotes' and shows collection statistics: 'COLLECTION SIZE: 61B', 'TOTAL DOCUMENTS: 1', and 'INDEXES TOTAL SIZE: 36KB'. It has tabs for 'Find', 'Indexes', and 'Aggregation'. Below these are buttons for 'INSERT DOCUMENT', 'FILTER {"filter": "example"}', 'Find', and 'Reset'. A section titled 'QUERY RESULTS 1-1 OF 1' shows a single document with its fields: '\_id', 'pid', and 'uid'.

Figure 1.3 xyz

**Clusters**

⚠ Please set your time zone    Usage This Month:\$0.00 del

SELF > PROJECT 0 > CLUSTERS

## Cluster0

VERSION	REGION
4.0.12	

Overview    Real Time    Metrics    **Collections**    Profiler    Performance Advisor    Comm.

DATABASES: 1    COLLECTIONS: 5

+ Create Database

NAMESPACES

- sociocop
  - media
  - plusones
  - posts
  - upvotes
  - users**

**sociocop.users**

COLLECTION SIZE: TOTAL INDEXES TOTAL SIZE:

839B DOCUMENTS: 36KB

3

Find    Indexes    Aggregation

INSERT DOCUMENT

FILTER {"filter": "example"}    Find    Reset

QUERY RESULTS 1-3 OF 3

```
_id: ObjectId("5d9c7d571c9d4400001ff9e")
uid: "1234"
username: "tharunrajeev"
firstname: "Tharun"
lastname: "Rajeev"
email: "tharunrajeev29@gmail.com"
gender: "m"
profilePic: "https://adayinthelifeimages.com/wp-content/uploads/2016/05/brisbane-he..."
```

```
_id: ObjectId("5d9f5450f8d537202c645401")
uid: "5nhzp6csk1kvuv8g"
firstname: "Tharun"
lastname: "Rajeev"
email: "tharunrajeev6543@gmail.com"
username: "tharun"
password: "$2b$10$zeK2Dk9GgeWOL./9Pi6jX.I.s168vAlkizxAdu
passw...: hpc7H90I6ey6lm"
gender: null
passwordChangedAt: 1570722896031
registeredAt: 1570722896031
```

```
_id: ObjectId("5d9f802dcc1f270f3407fc98")
uid: "5nhzp304k1l2jjn7"
firstname: "Tharun"
lastname: "Rajeev"
email: "ashutosh2726@gmail.com"
username: "ashutosh"
password: "$2b$10$7RTKTFU61bw3b/BYhtzQrO/ig1zH3ksq013E6BM
passw...: ZsnsDY9p4Zr5.2"
gender: null
passwordChangedAt: 1570734125106
registeredAt: 1570734125106
```

Figure 1.4 xyz

# **CHAPTER 8**

## **TESTING**

Testing is a process of executing a program with the intent of finding an error.

Testing is a crucial element of software quality assurance and presents ultimate review of specification, design and coding.

System Testing is an important phase. Testing represents an interesting anomaly for the software. Thus a series of testing are performed for the proposed system before the system is ready for user acceptance testing.

A good test case is one that has a high probability of finding an as undiscovered error. A successful test is one that uncovers an as undiscovered error.

### **8.1 Testing Objectives**

1. Testing is a process of executing a program with the intent of finding an error
2. A good test case is one that has a probability of finding an as yet undiscovered Error
3. A successful test is one that uncovers an undiscovered error

### **8.2 Testing Principles**

1. All tests should be traceable to end user requirements
2. Tests should be planned long before testing begins
3. Testing should begin on a small scale and progress towards testing in large
4. Exhaustive testing is not possible

To be most effective testing should be conducted by a independent third party

The primary objective for test case design is to derive a set of tests that has the highest likelihood for uncovering defects in software. To accomplish this objective two different categories of test case design techniques are used. They are

1. White box testing.
2. Black box testing.

### **8.2.1 White-box testing:**

White box testing focus on the program control structure. Test cases are derived to ensure that all statements in the program have been executed at least once during testing and that all logical conditions have been executed.

### **8.2.2 Black-box testing:**

Black box testing is designed to validate functional requirements without regard to the internal workings of a program. Black box testing mainly focuses on the Information domain of the software, deriving test cases by partitioning input and output in a manner that provides through test coverage. Incorrect and missing Functions, interface errors, errors in data structures, error in functional logic are The errors falling in this category.

## **8.3 Testing strategies:**

A strategy for software testing must accommodate low-level tests that are necessary to verify that all small source code segment has been correctly implemented as well as high-level tests that validate major system functions against customer requirements.

#### **8.4 Testing fundamentals:**

Testing is a process of executing program with the intent of finding error. A good test case is one that has high probability of finding an undiscovered error. If testing is conducted successfully it uncovers the errors in the software. Testing cannot show the absence of defects, it can only show that software defects present.

#### **8.5 Testing Information flow:**

Information flow for testing flows the pattern. Two class of input provided to test The process. The software configuration includes a software requirements Specification, a design specification and source code.

Test configuration includes test plan and test cases and test tools. Tests are Conducted and all the results are evaluated. That is test results are compared with Expected results. When erroneous data are uncovered, an error is implied and Debugging commences.

#### **8.6 Unit testing:**

Unit testing is essential for the verification of the code produced during the Coding phase and hence the goal is to test the internal logic of the modules. Using the detailed design description as a guide, important paths are tested to Uncover errors with in the boundary of the modules. These tests were carried out During the programming stage itself. All units of ViennaSQL were successfully Tested.

### **8.7 Integration testing :**

Integration testing focuses on unit tested modules and build the program

structure that is dictated by the design phase.

### **8.8 System testing:**

System testing tests the integration of each module in the system. It also tests to

find discrepancies between the system and it's original objective, current

specification and system documentation. The primary concern is the compatibility

of individual modules. Entire system is working properly or not will be tested

here, and specified path ODBC connection will correct or not, and giving output or

not are tested here these verifications and validations are done by giving input

values to the system and by comparing with expected output. Top-down testing

implementing here.

### **8.9 Acceptance Testing:**

This testing is done to verify the readiness of the system for the implementation.

Acceptance testing begins when the system is complete. Its purpose is to provide

the end user with the confidence that the system is ready for use. It involves

planning and execution of functional tests, performance tests and stress tests in

order to demonstrate that the implemented system satisfies its requirements.

Tools to special importance during acceptance testing include:

1. Test coverage Analyzer – records the control paths followed for each test case.
2. Timing Analyzer – also called a profiler, reports the time spent in various regions
3. of the code are areas to concentrate on to improve system performance.
4. Coding standards – static analyzers and standard checkers are used to inspect
5. code for deviations from standards and guidelines.

### **8.10 Test Cases:**

Test cases are derived to ensure that all statements in the program have been executed at least once during testing and that all logical conditions have been executed.

Using White-Box testing methods, the software engineer can drive test cases that

6. Guarantee that logical decisions on their true and false sides.
7. Exercise all logical decisions on their true and false sides.
8. Execute all loops at their boundaries and within their operational bounds.
9. Exercise internal data structure to assure their validity.

The test case specification for system testing has to be submitted for review before system testing commences

## **CHAPTER 9**

### **Conclusion and Future Enhancement**

This is a mobile based application to address the need of removal of daily traffic problem.

Everyday we face many issues like a traffic jam, illegal parking, road rages, accidents, path holes etc. These can be avoided and prevented to a larger extent if and only if they are reported on time to concerned authorities.

It identifies the user by gmail id,username where once registered data will be saved in the database so that false complaints and misuse of the app can be prevented from non-beneficial people.

The app can be used only when the gps is on to locate the exact location of the problem risen.

The difference between existing system is here we made some extra features which makes interface more smoother and easy to use and some updates which helps to understand the problem and escalate it to the department which it is related.

Here if the officials are taking time to resolve the issue then basically the locals viewing the issue can also resolve the problem and get recognized as they did it before the government. This may create a competition between the people and government and capture user feelings using upvotes, plus ones and comments

#### **2.2.1 Features:**

Some features which are add additionally are

- Here we can share the problem where people can share the issue which they already faced so that the problem will be in top of the app which is mostly shared and highlighted
- People can upvote the issue which they see on the app which they seem they also faced the same issue and can be shared.

- Comment section will be enabled to all the posts as that people can talk about the issue and share their views about the issue.

Enhancement that can be made in the future is that google api maps can be provided along with the issue to have quicker response from the officials and thus help the locality be free from daily small problems leading to hinderances in major problems. The advanced system should have a reward system based on the interests of people acting and using the application and looking into their needs and rewarding the properly so as to create a healthy competition between people and government. The people thus can solve their own problems without depending upon any officials or a third person.

The name of the person who resolves the problem will be given creditor recognition by which people will be motivated to do more such tasks.

2:25 P 53%

# SocialCop

Tharun Rajeev

post1

SEND REQUEST 01/01/1970

0 +1 0

Tharun Rajeev

post2

Home

Search

Pencil

Bell

User icon



3:00 P

58%

POST



Tharun Rajeev



post1

SEND REQUEST

01/01/1970



o

+1

o



Enter description

