Volume: 10 Issue: I Month of publication: January 2022

DOI: https://doi.org/10.22214/ijraset.2022.39797

Advanced and Secure Laboratory Management System

Harsh Ranjan[1], Manav Aggarwal[2], Gautam Gupta[3], Sanat Walia[4]
[1]Department of Computer Science Engineering, Dr. Akhilesh Das Gupta Institute of Technology and Management (affiliated to Guru Gobind Singh Indraprastha University), New Delhi, India

Abstract: Advanced & Secure Laboratory Information Management System, TRLIMS is the management system which has live tracking system for all the testing and research conducted at the laboratory. This system is developed to achieve diverse functionality for the disciplines such as mechanical, chemical, environmental, microbiology and non-destructive fields.

The basic features of this application are that it can manage the data related to client, employees and testing results of the laboratory. Apart from that since the application is fully hosted on server which offers flexibility, providing future scope for more hardware and operating system configuration. This application provides very enhanced turn-around-time (TAT) for the material testing laboratory It aims to manage the employees, clients and associated testing data to improve the lab productivity. The application allows clients to track their improvement in sample testing from time to time, the data is updated on server by employees who perform tests at the premises.

This paper could provide guidance to understanding the operation mechanism of Laboratory Information Management System.

I. INTRODUCTION

The paper is organized as follows:

1) In section 1 a literature overview of this application is given.
2) In section 2, the current applications related to LIMS have been mentioned.
3) In section 3, the requirements related to software and hardware as well as designed documents about the application has been mentioned.
4) In section 4, the methodologies for development including the algorithms have been mentioned.
5) In section 5, results have been discussed (including screenshots). Finally, the conclusion and future scope has been discussed in the last section of the paper.

II. OVERVIEW

The major unique valuation point of this app is to make it most secure and fast. All the data is on the server which means the data can be accessed and modified easily. To login as customer or employee very strong authentication system has been used. The client required unique password (OTP) every time for login whereas the employee must be within the premises of laboratory to login otherwise it will logout automatically. The customer gets unique code during the time of registration along with barcode. When the customer sign in, they can view the progress report (stage of testing) but not the actual data.

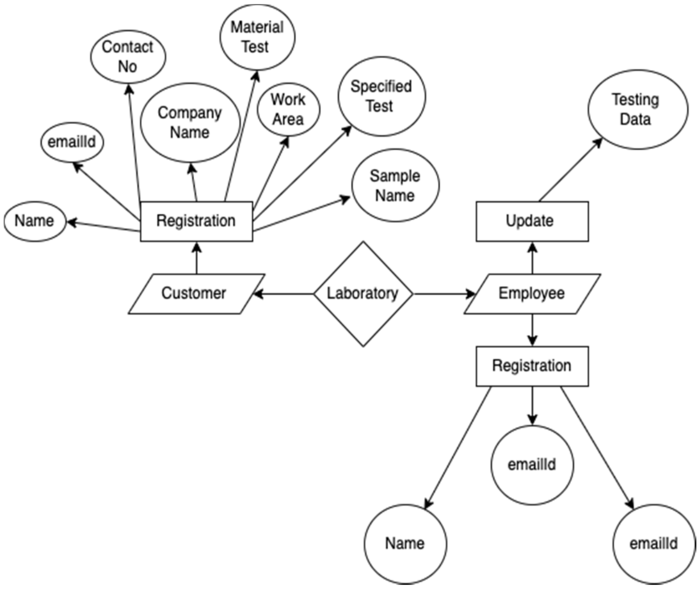
The employee can sign in using unique employee id and OTP along with enabled service location. They can view as well as update the data by just scanning the barcode or entering the value of customer id. Hence this way we have made it secure for both organization as well as client. Since as we have to make it easily accessible and fast. To deliver above mentioned system we have made android application and web-application with client ui as well as employee ui. To craft the application programming interface, we've used php. On the other hand

III. RECENT SIMILAR APPS

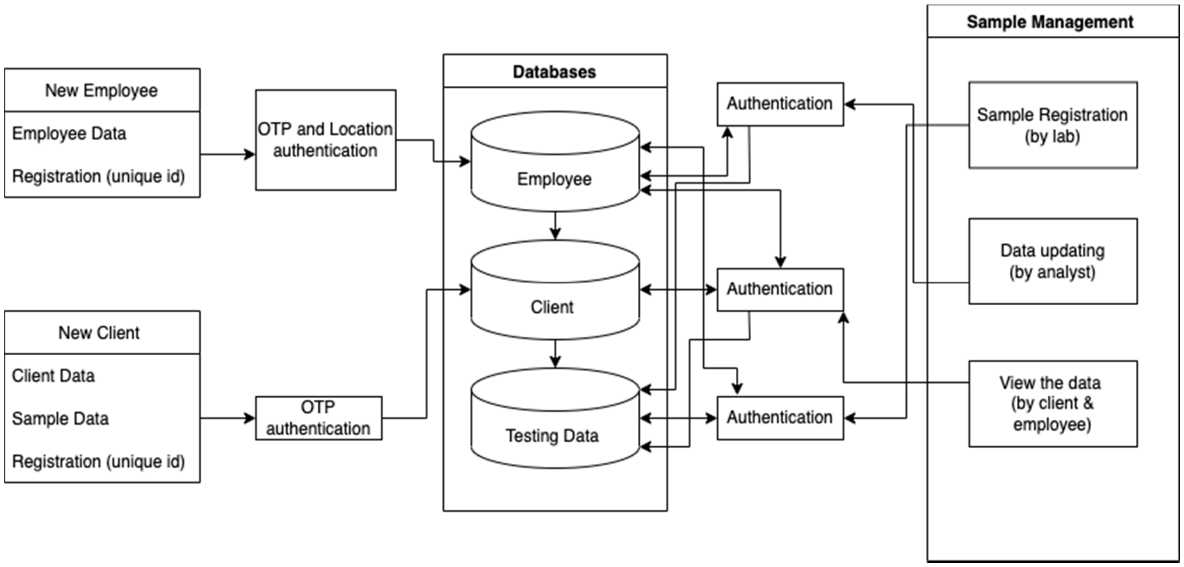
1) Starlims- Dr. Lal Path Labs
2) LIMSBIS- Bureau of Indian Standards
3) Thermo Scientific Sample Manager LIMS Software
4) Qualis LIMS

IV. BASIC REQUIREMENTS

Flow chart explaining the basic layout or the application



UML Diagram



A. *Software Requirements*

1) Android
2) SQL
3) Java
4) PHP
5) JavaScript
6) XML
7) Firebase

B. *Hardware Requirements*

1) Smartphone
2) Internet Connection

V. TECHNOLOGIES USED

A. Android

It is and operating system that optimises how we use our phone. It is software package and Linux based os. The modified Linux Kernel operates as the HAL, and provides device driver, process management, memory management, as well as networking functionalities respectively.

1) Why Android??

Because: The libraries in Android are interfaced through Java. It is in this layer that Android specific libc (Bionic) is located. The surface manager handles the user interface (UI) windows.

The Android runtime layer holds the Dalvik Virtual Machine (such as Java or IO).

In android architecture, there is application layer which include applications like camera, maps, browser, sms, calendar, contacts and native applications. These application works with end user with the help of application framework to operate. It has activity manager, resource manager, notification manager, location manager, window manager, telephony manager and android runtime where all the applications are executed.

Android has libraries that cannot be accessed directly. There is library for security in android and its main components are Privacy setting content provider, privacy setting manager and privacy aware components.

Android based system utilise their own virtual machine, known as Dalvik Virtual Machine. It uses the special byte-code, thus native Java bytecode cannot be directly utilised. To improve the efficiency, the DVM implements directly.

Android has some specific kernel enhancements that includes alarm driver, shared memory driver, binder, power management, low memory killer, kernel debugger and logger. The android linux kernel component first call the init process. The init process accesses the files in init.rc and init.device.rc. Out of the init.rc file, a process labelled zygote is started. These java classes can be reused by android application and hence this step expedites the overall start-up process.

2) The Android Libraries Used

a) *Lottie for Android(v3.0.7):* This dependency has been added because it parses Adobe After Effects animations exported as json with Bodymovin an renders them natively on mobile.
b) *Appcompat(1.3.1):* It is the static library added in order to use APIs that are not available for older platform versions or utility APIs that aren't a part of the framework APIs. In order to use this, Google Maven repository was added to the project.
c) *Material Component(v1.4.0):* It has been added in order to use APIs that provide implementations of the Material Design specification.
d) *Constraint Layout(v2.1.2):* It has been used to position and size the widgets in a flexible way.
e) *Play service location(v18.0.0):* It has been added so that location service can be enabled because our requires location to change or update the data.
f) *Junit(4.+):* It has been used to write the repeatable tests. It is an instance of the xUnit architecture for unit testing frameworks.
g) *OkHttp(v4.9.1):* It allows all the requests to the same host to share a socket. It preserves when the network is troublesome. It silently recovers from common connection problems.

B. Firebase

Google's firebase is mobile platform that lets us develop mobile apps. Since our application is geared towards business with intention to fasten, make it secure and easy to handle we have used firebase. It also makes it easy to add certain important features like cloud messaging, authentication, hosting, crash reporting, real time database.

C. JavaScript

It is a scripting language that allows to implements complex features on web pages. Helps in creating dynamically updating content, control multimedia, animate, images and much more. It can calculate, manipulate and validate data. It helps in adding interactive behaviour to webpages. We have used JS frameworks for developing and building web and mobile apps.

1) *Sass:* It is pre-processor scripting language that is compiled into CSS. It has been used to facilitate us to write clean, easy and less CSS in a programming construct. Sass has been used to bring variables to CSS. Acceptance values for variables including numbers, strin, colours, null, lists and maps. !global flag has enabled to set the variable globally from the declaration. !default flag has allowed to mae sure there is a default value for a variable in the event that one is not provided.

2) *Ajax:* Ajax stands for Asynchronous JavaScript And XML. It has helped to use the XMLHttpRequest object to communicate with the servers. Basically we required to send and receive the information in various formats, including JSON, XML, HTML and text files. Also it could make request to the server without reloading the page and receive/work with data from the server.
3) *Owl Carousel:* We needed slider in the homepage for our website for impressing user experience and to provide ease by the site admin. Owl Carousel made this possible as it is simple jQuery plugin which lets us create responsive and touch enabled carousel slider. Also, it could handle custom events and call backs.
4) *Anime.js:* It is simple and flexible to use. We used anime to add small animations to our project.

D. MySQL

To store the data of the clients, employees and testing results we needed client-server architecture. MySQL which is relational database management system served the purpose. It has been used for maintaining the customer-facing application. This has allowed better optimise actions like data retrieval and updating information. A logical model has been defined overall of the contents of the database. MySQL has provided the ultimate in scalability, sporting the capacity to handle our deeply embedded application. Also MySQL is the de-facto standard for high-traffic web sites because of its high-performance query engine, tremendously fast data insert capability and strong support for specialized web functions. MySQL also offers exceptional security feature that ensured absolute data protection. It provides comprehensive support for our development process. MySQL provides connector and drivers that allow all forms of application to make
use of MySQL as preferred by management server.

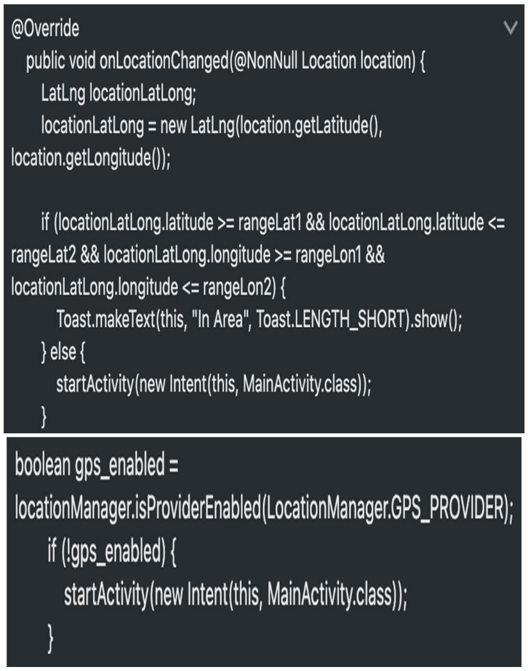
E. PHP

PHP has been used for application programming interface(API) which defines the classes, methods, functions and variables that our application needs to call in order to carry out the desired task. We have made representational state transfer API (REST) which is the backbone our application. We have preferred PHP ecosystem for API because it provides great tools to process the requests and generate JSON responses with the correct HTTP status codes. They also make it easy to handle common issues like authentication/authorization, request validation and data transformation.

VI. SOME IMPORTANT ALGORITHMS USED

A. Location Service

The location service in the application has been used to make it more secure. As to update the data we have specified certain co-ordinates outside which the person won't be able to login to update the data.



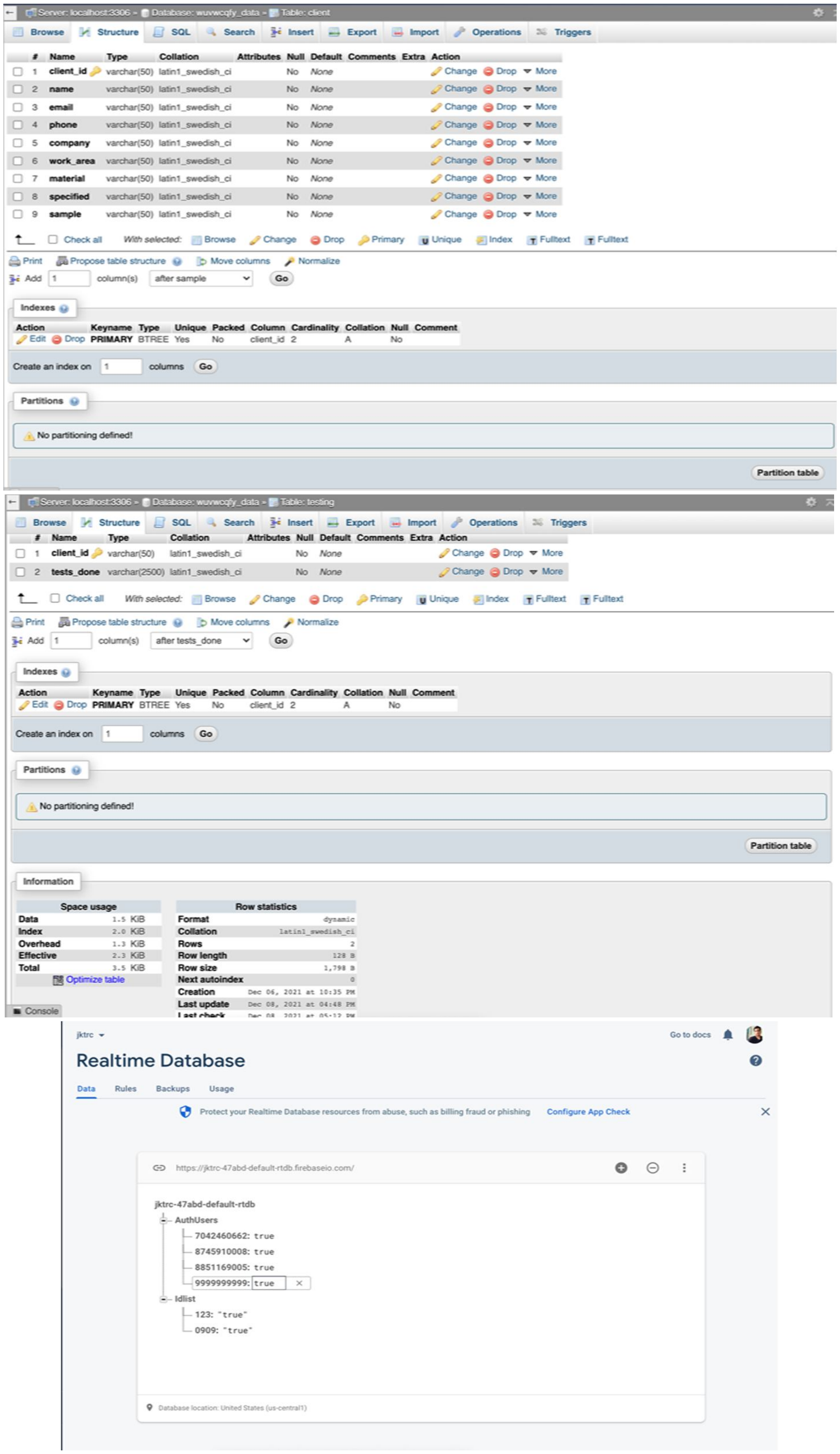
```
@Override
   public void onLocationChanged(@NonNull Location location) {
       LatLng locationLatLong;
       locationLatLong = new LatLng(location.getLatitude(),
location.getLongitude());

       if (locationLatLong.latitude >= rangeLat1 && locationLatLong.latitude <=
rangeLat2 && locationLatLong.longitude >= rangeLon1 &&
locationLatLong.longitude <= rangeLon2) {
           Toast.makeText(this, "In Area", Toast.LENGTH_SHORT).show();
       } else {
           startActivity(new Intent(this, MainActivity.class));
       }
```

```
boolean gps_enabled =
locationManager.isProviderEnabled(LocationManager.GPS_PROVIDER);
      if (!gps_enabled) {
          startActivity(new Intent(this, MainActivity.class));
      }
```

Fig: GPS Authentication algorithm

B. Database Algorithm

We have also designed our real time database in such a way that it can be easily used to access and centralise the data. Also, it would make very easy to add new scope and modules in our application.


Server: localhost:3306 » Database: wuvwcqfy_data » Table: client
Browse
Structure
SQL
Search
Insert
Export
Import
Operations
Triggers

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	client_id	varchar(50)	latin1_swedish_ci		No	*None*			Change Drop More
2	name	varchar(50)	latin1_swedish_ci		No	*None*			Change Drop More
3	email	varchar(50)	latin1_swedish_ci		No	*None*			Change Drop More
4	phone	varchar(50)	latin1_swedish_ci		No	*None*			Change Drop More
5	company	varchar(50)	latin1_swedish_ci		No	*None*			Change Drop More
6	work_area	varchar(50)	latin1_swedish_ci		No	*None*			Change Drop More
7	material	varchar(50)	latin1_swedish_ci		No	*None*			Change Drop More
8	specified	varchar(50)	latin1_swedish_ci		No	*None*			Change Drop More
9	sample	varchar(50)	latin1_swedish_ci		No	*None*			Change Drop More

□ Check all
With selected: Browse Change Drop Primary Unique Index Fulltext Fulltext
Print
Propose table structure
Move columns
Normalize
Add 1 column(s) after sample Go
Indexes

Action	Keyname	Type	Unique	Packed	Column	Cardinality	Collation	Null	Comment
Edit Drop	PRIMARY	BTREE	Yes	No	client_id	2	A	No	

No partitioning defined!

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	client_id	varchar(50)	latin1_swedish_ci		No	*None*			Change Drop More
2	tests_done	varchar(2500)	latin1_swedish_ci		No	*None*			Change Drop More

□ Check all
With selected: Browse Change Drop Primary Unique Index Fulltext Fulltext
Print
Propose table structure
Move columns
Normalize
Add 1 column(s) after tests_done Go
Indexes

Action	Keyname	Type	Unique	Packed	Column	Cardinality	Collation	Null	Comment
Edit Drop	PRIMARY	BTREE	Yes	No	client_id	2	A	No	

No partitioning defined!

Space usage	
Data	1.5 KiB
Index	2.0 KiB
Overhead	1.3 KiB
Effective	2.3 KiB
Total	3.5 KiB
Optimize table	

Row statistics	
Format	dynamic
Collation	latin1_swedish_ci
Rows	2
Row length	128 B
Row size	1,798 B
Next autoindex	0
Creation	Dec 06, 2021 at 10:35 PM
Last update	Dec 08, 2021 at 04:48 PM
Last check	Dec 08, 2021 at 05:12 PM

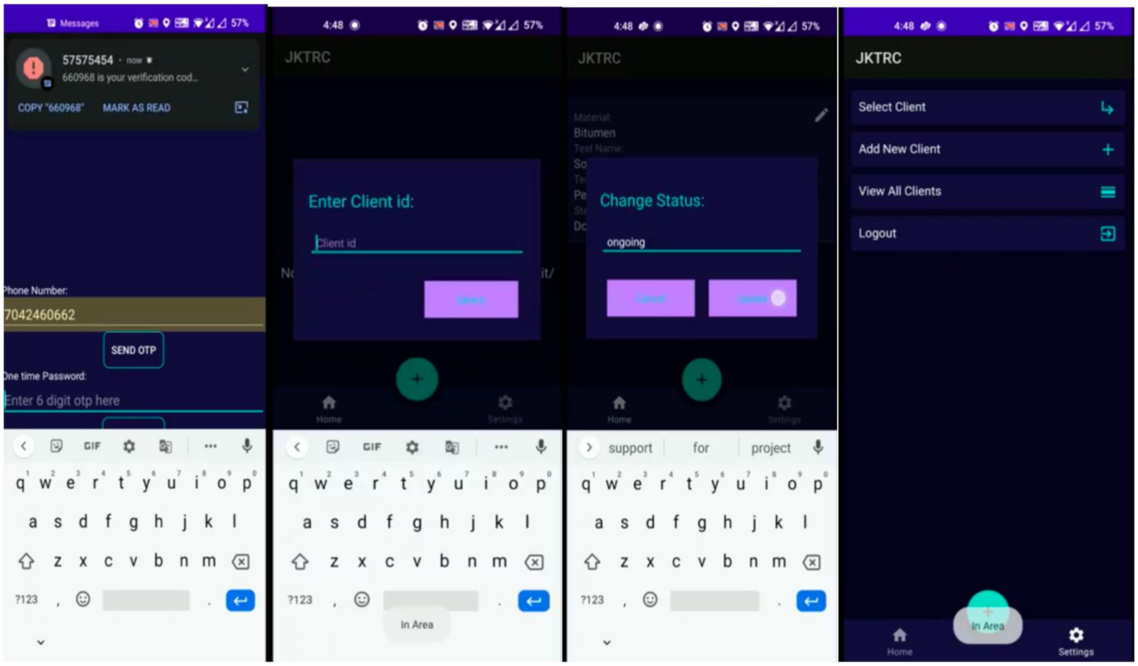
Protect your Realtime Database resources from abuse, such as billing fraud or phishing

```
https://jktrc-47abd-default-rtdb.firebaseio.com/

jktrc-47abd-default-rtdb
  AuthUsers
    7042460662: true
    8745910008: true
    8851169005: true
    9999999999: true
  Idlist
    123: "true"
    0909: "true"
```

Database location: United States (us-central1)

Fig: Database design

VII. RESULTS(WITH SCREENSHOTS OF UI)



57575454 · now
660968 is your verification cod...

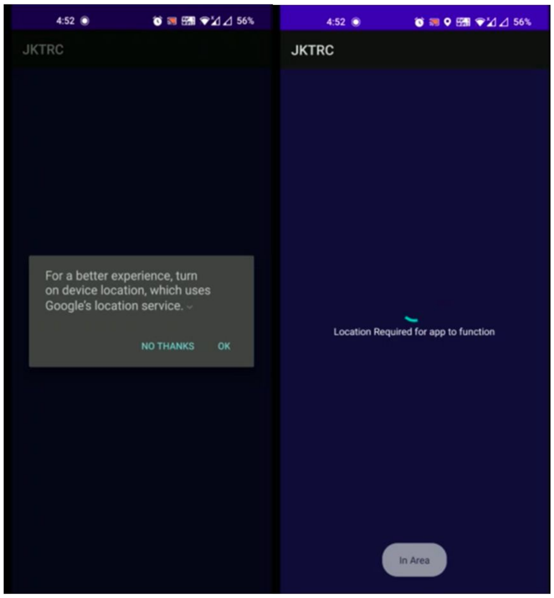
Phone Number: 7042460662

One time Password: Enter 6 digit otp here

Enter Client id: Client id

Material: Bitumen

Change Status: ongoing



For a better experience, turn on device location, which uses Google's location service.

Location Required for app to function

VIII. FUTURE SCOPE

A. In today's time the on-premises system fails most of the time to offer flexibility, our application is completely online which offers more flexibility, efficiency and decreases the cost while at same time improves the quality.

B. Most of the time many laboratories have sensitive data. We have GPS, OTP and owner password authentication levels to update the data which makes the more secure than existing applications.

C. We have impeccable design of our database which will help our application to also integrate with testing machines. Hence, it would help to keep track of the machine accuracy.

D. Since the data and results can be stored online, we can easily add module for internal audit. Internal audit helps to stay consistent with work.

E. Our application will promote and give fair opportunity to small scale laboratories also to manage easily.

IX. CONCLUSION

A. Hence we successfully completed our research on how to build advanced and secure LIMS using limited technologies.

B. We are successful in making it easy for the laboratory to handle and organise the data easily.

C. As the demand in R&D field is developing rapidly the laboratories must switch to online-based system to increase the efficiency.

D. We are successful in providing simple and interactive user interface which can easily be customised according to the need and requirements of the laboratories.

E. The unblemished design of our application is in such a way that it could adapt to future changes as we've planned and mention it in future scope.

F. The application has potential to automate the data entry process, manage inventory, streamline the workflows, and assure quality. It reduces time spent on locating the sample and records.

REFERENCES

[1] Improving Efficiency in Delivering Clinical Laboratory Results Using a Cloud-Based LIMS
[2] https://www.researchgate.net/publication/299394606_Android_Operating_System_Architecture_Security_Challenges_and_Solutions
[3] https://www.researchgate.net/publication/236221555_Laboratory_information_and_management_system_A_tool_to_increase_laboratory_productivity
[4] https://ieeexplore.ieee.org/document/6240739