A MINI PROJECT REPORT ON

WOMEN SAFETY APP

A dissertation submitted in partial fulfilment of the

Requirements for the award of the degree of

BACHELOR OF TECHNOLOGY

in

INFORMATION TECHNOLOGY

Submitted by

N.Bhavana (18B81A1267)

V.Laharika (18B81A1277)

T.Rishitha (18B81A1286)

Under the esteemed guidance of

S.Anupkant

Assistant Professor, IT Department

CVR College of Engineering

DEPPARTMENT OF INFORMATION TECHNOLOGY

CVR COLLEGE OF ENGINEERING

ACCREDITED BY NBA, AICTE & Affiliated to JNTU-H

Vastunagar, Mangalpally (V), Ibrahimpatnam (M), R.R. District, PIN-501 510

2020-2021

Cherabuddi Education Society's

CVR COLLEGE OF ENGINEERING

(An Autonomous Institution)

ACCREDITED BY NATIONAL BOARD OF ACCREDITATION, AICTE

(Approved by AICTE & Govt. of Telangana and Affiliated to JNT University)

Vastunagar, Mangalpalli (V), Ibrahimpatan (M), R.R. District, PIN - 501 510

Web : http://cvr.ac.in, email : info@cvr.ac.in

Ph : 08414 - 252222, 252369, Office Telefax : 252396, Principal : 252396 (O)

DEPARTMENT OF INFORMATION TECHNOLOGY

CERTIFICATE

This is to certify that the Project Report entitled **"WOMEN SAFETY APP" is** submitted by **N.Bhavana (18B81A1267)**, **V.Laharika (18B81A1277)**, **T.Rishitha (18B81A1286)** during the academic year 2020-2021, in partial fulfilment of requirement for the degree of Bachelor of Technology in Information Technology from Jawaharlal Nehru Technological University Hyderabad, is a bonafide record of work carried out by them under my guidance and supervision.

Certified further that to my best of the knowledge, the work in this dissertation has not been submitted to any other institution for the award of any degree or diploma.

INTERNAL GUIDE	**HEAD OF THE DEPARTMENT**
S.Anupkant	**Dr.Bipin Bihari Jayasingh**
Assistant Professor, IT Department	Professor, IT Department
PROFESSOR INCHARGE	**PROJECT COORDINATOR**
Prof. R.Seetharamaiah	**G.Sunitha Rekha**
Professor, IT Department	Assistant Professor, IT Department

EXTERNAL EXAMINER

DECLARATION

We hereby declare that the project report entitled “**WOMEN SAFETY APP**” is an original work done and submitted to IT Department, CVR College of Engineering, affiliated to Jawaharlal Nehru Technological University Hyderabad, Hyderabad in partial fulfilment of the requirement for the award of degree of Bachelor of Technology in **Information Technology,** it is a record of bonafide project work carried out by us under the guidance of S.**Anupkant, Assistant Professor, Department of Information Technology.**

We further declare that the work reported in this project has not been submitted, either in part or in full, for the award of any other degree or diploma in this institute or any other Institute or University.

Signature of the Student

(N.Bhavana)

(18B81A1267)

Signature of the Student

(V.Laharika)

(18B81A1277)

Signature of the Student

(T.Rishitha)

(18B81A1286)

ACKNOWLEDGEMENT

The satisfaction of completing this project would be incomplete without mentioning our gratitude towards all the people who have supported us. Constant guidance and encouragement have been instrumental in the completion of this project.

First and Foremost, We thank the Chairman, Principal, Vice Principal for availing infrastructural facilities to complete the mini project in time.

We offer our sincere gratitude to our internal guide S.**Anupkant,** Assistant Professor, IT Department, CVR College of Engineering for his immense support, timely co-operation and valuable advice throughout the course of our project work.

We would like to thank the Professor In-Charge of Projects, **Prof.R.Seetharamaiah**, Professor, Information Technology for his valuable suggestions in implementing the project.

We would like to thank the Head of Department, Professor **Dr.Bipin Bihari Jayasingh**, for his meticulous care and cooperation throughout the project work.

We are thankful to G.**Sunitha Rekha,** Project Coordinator, Assistant Professor, IT Department, CVR College of Engineering for her supportive guidelines and for having provided the necessary help for carrying forward this project without any obstacles and hindrances.

We also thank the **Project Review Committee Members** for their valuable suggestions.

ABSTRACT

Women's safety is a big concern which has been the most important topic till date. Few crimes against ladies were terribly dread and fearful. In today’s world, it is not safe for a person to travel alone at night especially for women.

Our motto in developing this app is to provide a safe environment to women through smart phone as today most of the people are carrying smart phones to wherever they go.

This is a user-friendly application that can be accessed by anyone who has installed it in their smart phones. Our intention is to provide you with fastest and simplest way to contact your nearest help.

TABLE OF CONTENTS

CONTENTS **PAGE NUMBER**

LIST OF FIGURES

SCREENSHOTS

LIST OF TABLES

1. INTRODUCTION

Women often work across ethnic, religious, political, and cultural divides to promote peace. We all are aware of importance of women's safety ,but we must realize that they should be properly protected.

The best way to minimize your chances of becoming a victim of violent crime (robbery, sexual assault, domestic violence) is to identify and call on resources to help you out of dangerous situations. Whether you're in immediate trouble or get separated from friends during a night out and don't know how to get home, having these apps on your phone can reduce your risk and bring assistance when you need it. Here we introduce an app which ensures the safety of women.

I. EXISTING SYSTEM

Existing system app does not activate when they are in background, only when app is opened.

Limited contacts to be added.

Managing the contact details and message to be send is not exists.

DISADVANTAGES OF EXISTING SYSTEM

- Time Consuming
- Less Efficient
- Less Accurate
- Not User Friendly

II. PROPOSED SYSTEM

To bring in centralized system for women safety process. To make alert notifications to all registered contacts .

Reduce the risk and damage for women. And to save the precious life of n women.

Women can send the message to number by using this app along with location.

It provides the alarm with police siren so that other people around may hear.

ADVANTAGES OF PROPOSED SYSTEM

- User Friendly
- Reporting to contacts made easy
- Accurate location Trace
- Updating of Data is easy

1.1 LITERATURE SURVEY

Anup CJ et al designed a gadget like a typical belt. This design consisting of GPS, GSM, Zapper, Buzzer circuit. Controller in that activates the GPS, GSM in order to identify the location. Zapper circuit produce shock which has high voltage. Buzzer also gets activated which produces boisterous yelling sound.

SnehaLokesh et al proposed a system consist of android application, main device, and portable camera. Android application uses Phone GPS or GPS of the main device to locate the victim in the critical situation they were pressing the emergency button. The camera will be added through the photo will be captured and it will be sent to the server with ensure the data security, Main device is also attached with manually operating in pepper spray.

2.SOFTWARE REQUIREMENT SPECIFICATION

HARDWARE REQUIRMENTS

Processor	:	Intel Core
Hard Disk	:	1TB
Ram	:	4GB

SOFTWARE REQUIREMENTS

Operating system	:	Windows 7/10
User interface	:	Java, PHP
IDE/workbench	:	Android Studio
Data base	:	MySQL

- **Functional Requirements :**

  The system must send SMS to registered mobile numbers.

  It must be able to make a call to primary numbers in the contacts.

- **Non-Functional Requirements:**

  We should double click the power button.

  Internet and GPS location of the user should be on.

3. PROJECT DESIGN

System design is transition from a user oriented document to programmers or database personnel. The design is a solution, how to approach to creation of a new system. This is composed of several steps. It provides the understanding and procedural details necessary for implementing the system recommended in the feasibility study. Designing goes through logical and physical stages of development, logical design reviews the present the present system, prepare input and output specification, details of implementation plan and prepare a logical design walkthrough.

The system tables are designed by analyzing functions involved in the system and format of the fields is also designed. The fields in the database tables should define their role in the system. The design should made user friendly. The menu should be precise and compact.

3.1 UML CONCEPTS

The Unified Modelling Language (UML) is a standard language for writing software blue prints. The UML is a language for

- Visualizing
- Specifying
- Constructing
- Documenting

The uml is a language which provides vocabulary and the rules for contributing words in that vocabulary for the purpose of communication. A modelling language focus on the conceptual and physical representation of a system. Modelling yields an understanding of a system.

BULDING BLOCKS OF THE UML

The vocabulary of the UML encompasses three kinds of building blocks.

- THINGS
- RELATIONSHIPS
- DIAGRAM

THINGS IN THE UML

There are four kinds of things in UML

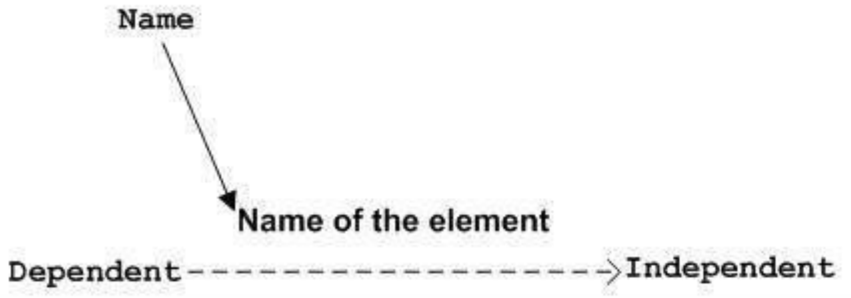
- Structural things
- Behavioral things
- Grouping things
- Annotational things

RELATIONSHIPS IN THE UML

There are four types of relationships in the UML

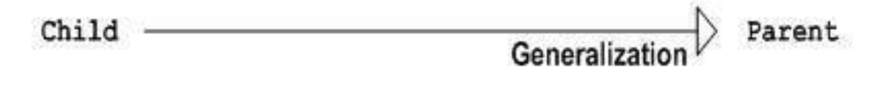
- Dependency
- Association
- Generalization
- Realization

A Dependency is a semantic relationship between two things in which a change to one thing many affect the semantic of the other thing (the dependent thing).



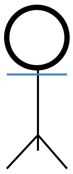
An Association is a several relationship that describes a link being a connection among objects.

A generalization is a specification/generalization relationship in which objects of the specialized element (the child) are substitutable for objects of the generalized element (the parent).



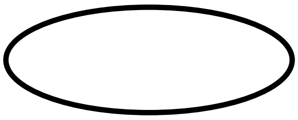
USE CASE DIAGRAMS

A use case diagram is a graph of actor set of use cases enclosed by a system boundary, communication association between the actors set of use cases enclosed by a system among use cases. The use case model defines the outside (actors) and inside (use case) of the system's behavior.



Actor

A use case is shown as an eclipse in a use case diagram



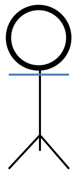
Use case

SEQUENCE DIAGRAMS

UML sequence diagrams are used to represent the flow of messages, events and between the object or components of a system. Time is represented in the vertical direction showing the sequence of interaction of the header elements, which are displayed horizontally at the diagram.

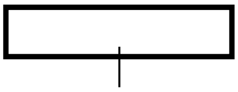
Sequence Diagram are used primarily to design, document and validate the architecture, interfaces and logic of the system by describing the sequence of actions that need to be performed to complete a task or scenario. UML sequence diagrams are useful design tools because they provide a dynamic view of the system behavior which can be difficult to extract from static diagrams or specification.

ACTOR



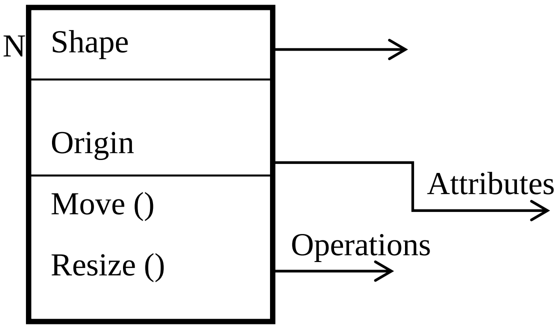
User

OBJECT



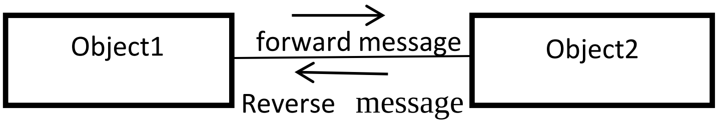
CLASS DIAGRAM

A class is a description of a set of objects that share the same attributes, operations, relationship, and semantics. Class implements one or more interfaces graphically a class is rendered as rectangle, usually including its name, attributes, and operations.



COLLABORATION

Collaboration defines an interaction and is a society of roles and other elements that work together to provide some cooperative behavior that's bigger that the sum of all the elements. Collaborations have structural as well as behavioral dimensions. A given class might participate in several collaborations.

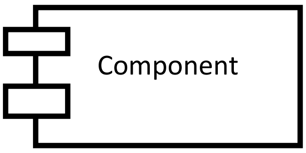


ACTIVITY DIAGRAM

An activity diagram is a special kind of state chart diagram that shows the flow from activity to activity within a system. Activity diagrams address the dynamic view of a system. They are especially important in modelling the function of a system and emphasize the flow of control among object.

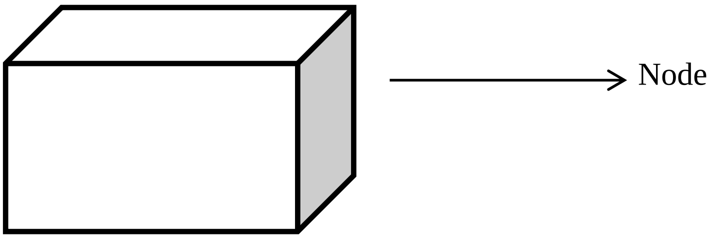
COMPONENT DIAGRAM

A component diagram shows the organizations and dependencies among a set of components. Component diagrams address the static implementation view of a system. They are related to class diagrams in that a component typically maps to one or more classes, interfaces, or collaborations.



DEPLOYMENT DIAGRAM

A deployment diagram shows the configuration of runtime processing nodes and the components that live on them. Deployment diagrams address the static development view of architecture. They are related to component diagrams in that a node typically encloses one or more components.



UML DIAGRAMS

3.1.1 USE CASE DIAGRAM

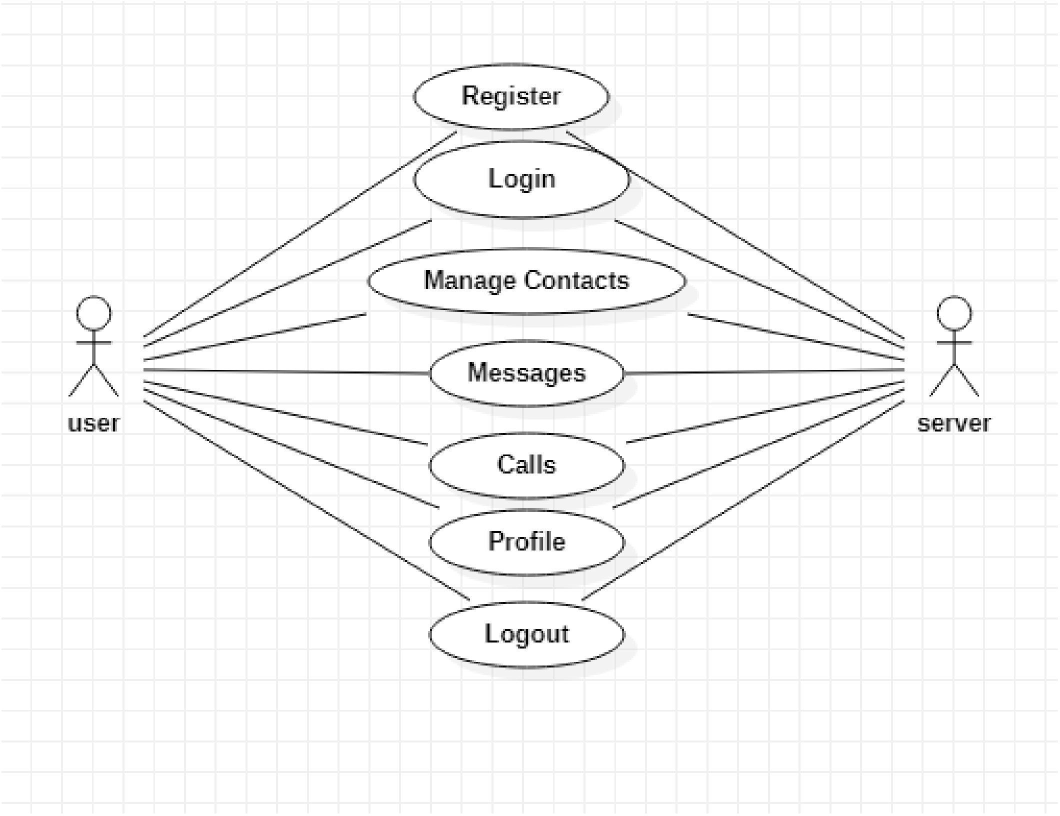


Figure 3.1.1 Use Case Diagram

3.1.2 CLASS DIAGRAM

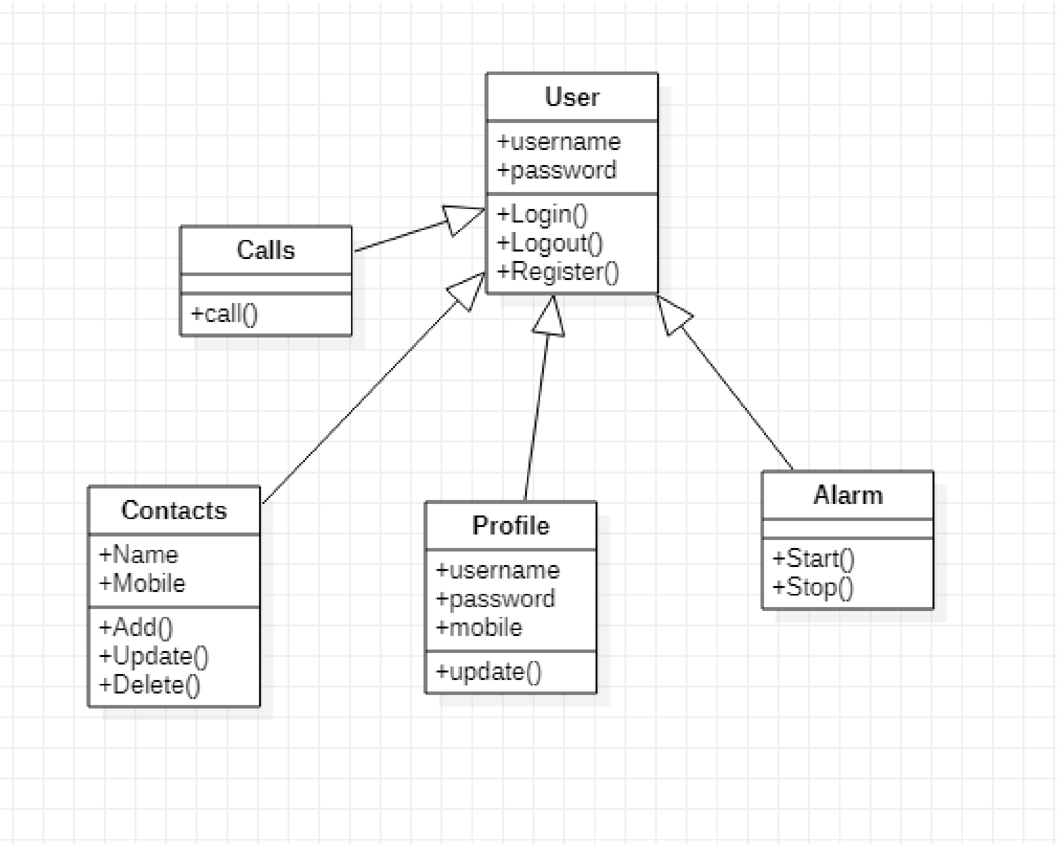


Figure 3.1.2 Class Diagram

3.1.3 SEQUENCE DIAGRAM

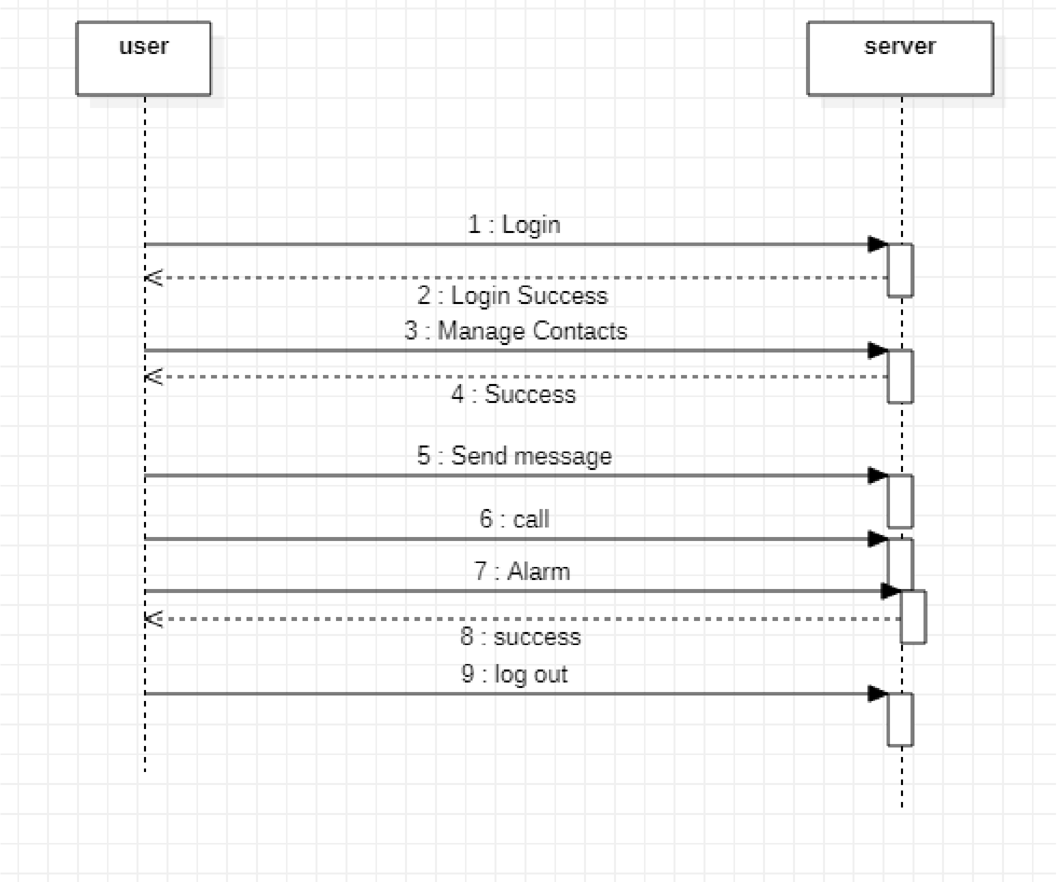


Figure 3.1.3 Sequence Diagram

3.1.4 ACTIVITY DIAGRAM

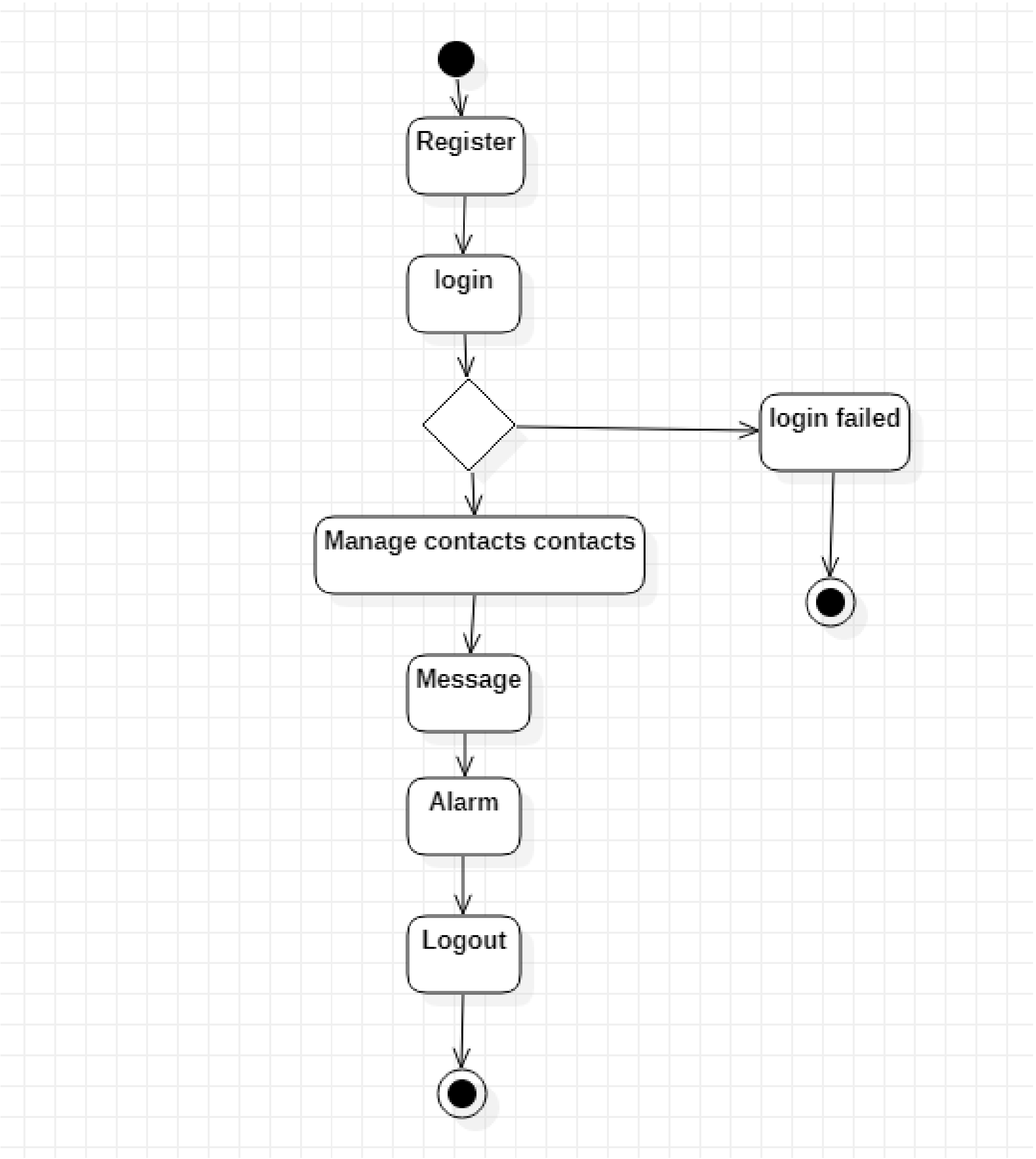


Figure 3.1.4 Activity Diagram

3.1.5 COLLABORATION DIAGRAM

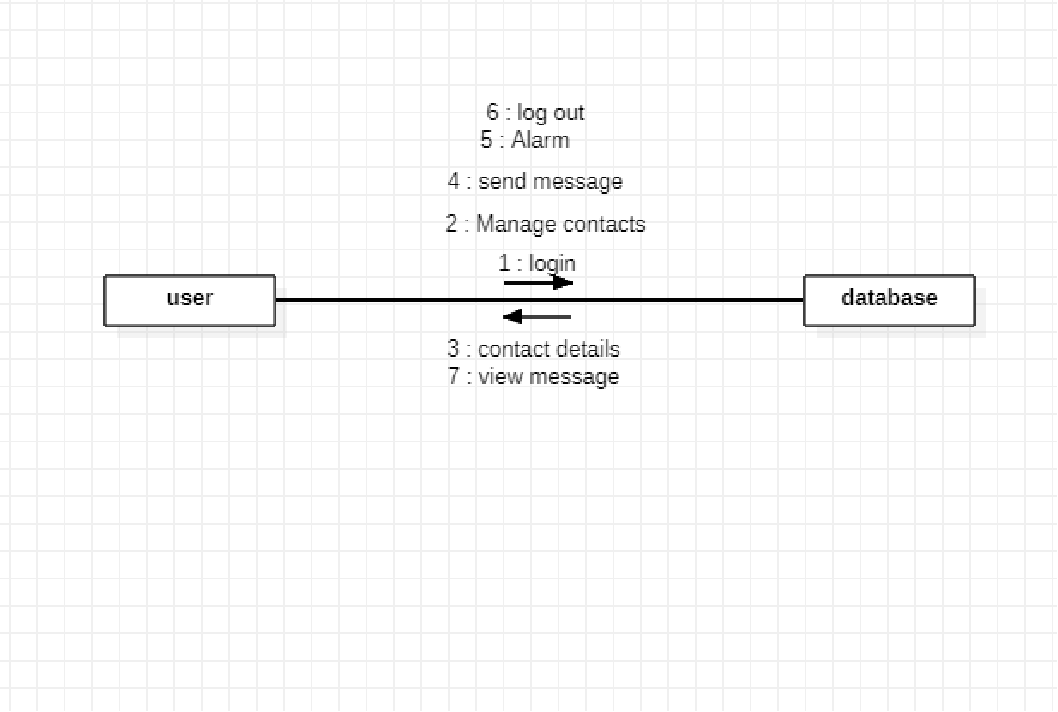


Figure 3.1.5 Collaboration Diagram

3.1.6 COMPONENT DIAGRAM

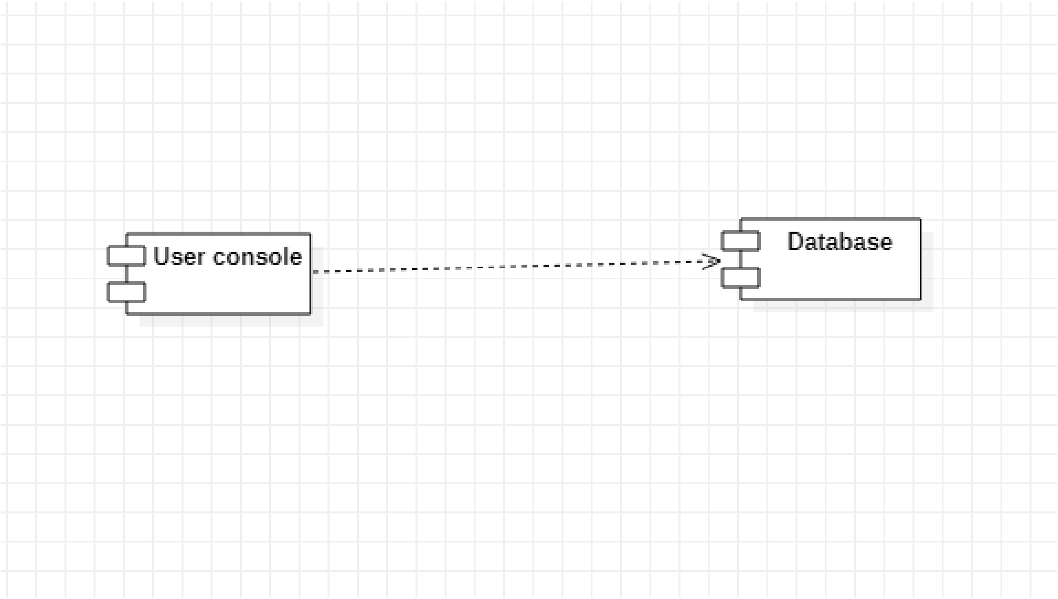


Figure 3.1.6 Component Diagram

3.1.7 DEPLOYMENT DIAGRAM

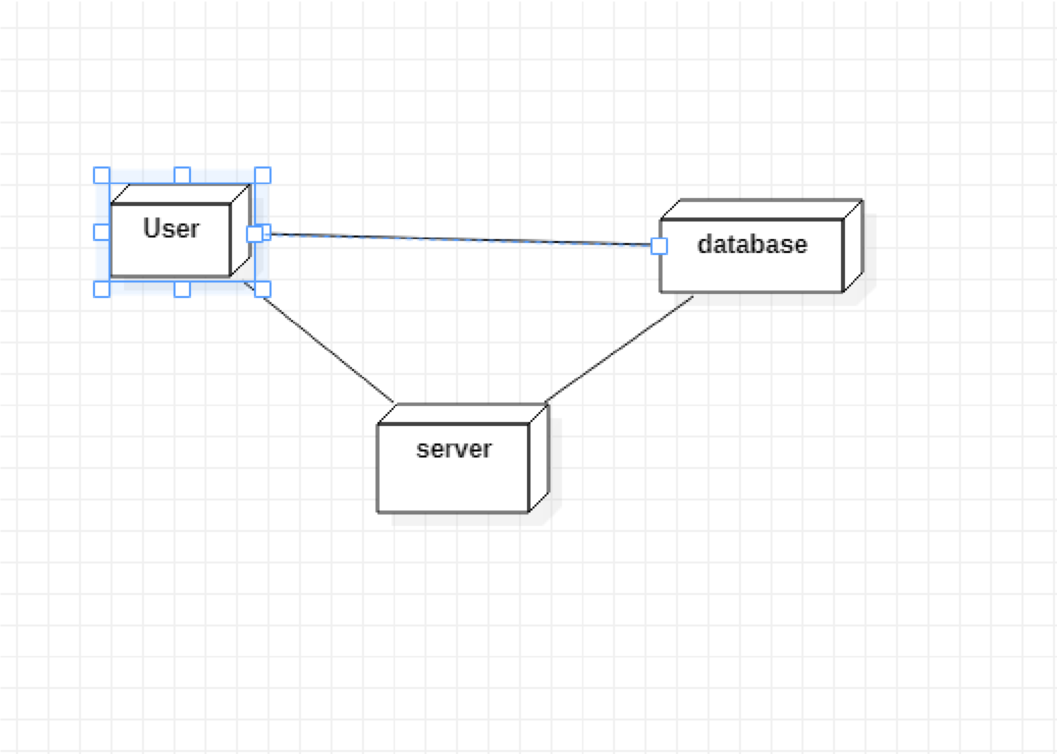


Figure 3.1.7 Deployment Diagram

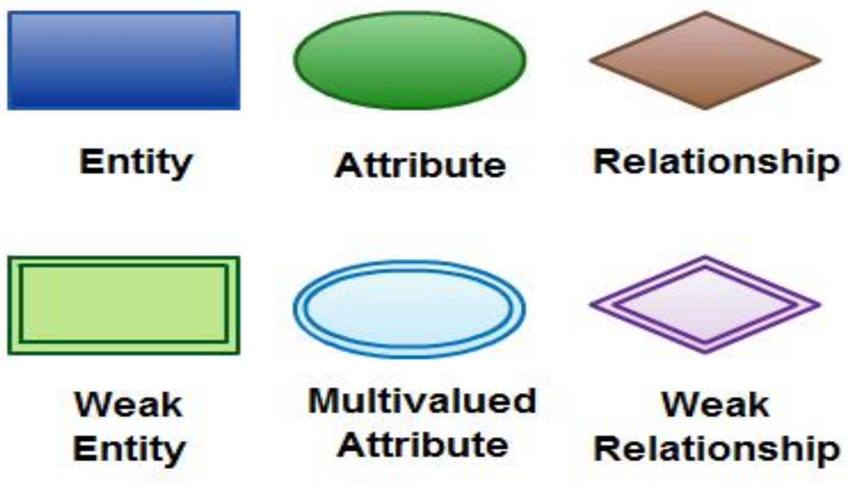
3.2 ER DIAGRAM

An Entity Relationship Diagram (ERD) is a visual representation of **different entities within a system and how they relate to each other**. For example, the elements writer, novel, and a consumer may be described using ER diagrams the following way:

ER MODELS IN DATABASE DESIGN

They are widely used to design relational databases. The entities in the ER schema become tables, attributes and converted the database schema. Since they can be used to visualize database tables and their relationships it’s commonly used for database troubleshooting as well.

ER DIAGRAM SYMBOLS AND NOTATIONS



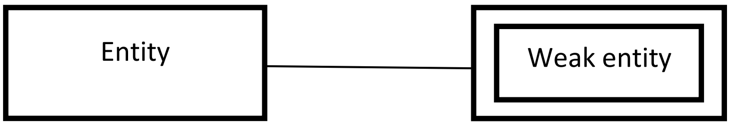
There are three basic elements in an ER Diagram: entity, attribute, relationship. There are more elements which are based on the main elements. They are weak entity, multi valued attribute, derived attribute, weak relationship, and recursive relationship. Cardinality and ordinality are two other notations used in ER diagrams to further define relationships.

ENTITY

An entity can be a person, place, event, or object that is relevant to a given system. For example, a school system may include students, teachers, major courses, subjects, fees, and other items. Entities are represented in ER diagrams by a rectangle and named using singular nouns.

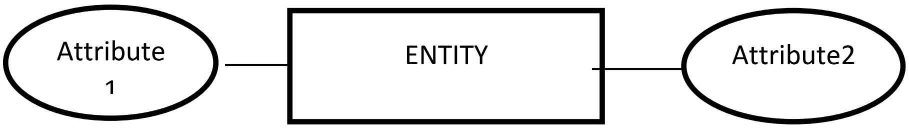
WEAK ENTITY

A weak entity is an entity that depends on the existence of another entity. In more technical terms it can be defined as an entity that cannot be identified by its own attributes. It uses a foreign key combined with its attributed to form the primary key. An entity like order item is a good example for this. The order item will be meaningless without an order so it depends on the existence of the order.



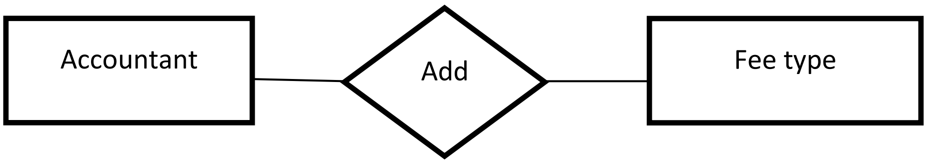
ATTRIBUTE

An attribute is a property, or characteristic of an entity, relationship, or another attribute. An entity can have as many attributes as necessary. Meanwhile, attributes can also have their own specific attributes. Note that some top level ER diagrams do not show attributes for the sake of simplicity. However, attributes are represented by oval shapes.



RELATIONSHIP

A relationship describes how entities interact. Relationships are represented by diamond shapes and are labeled using verbs.



ENTITY RELATIONSHIP DIAGRAM

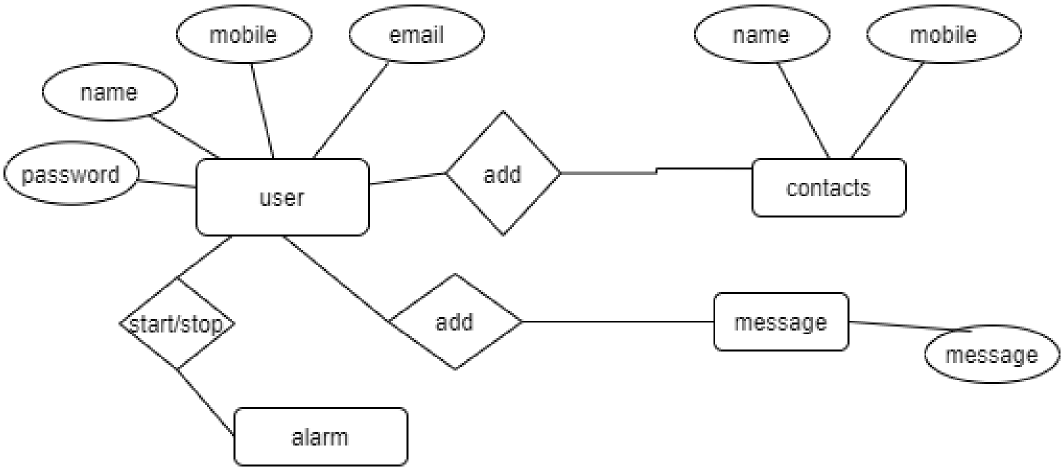


Fig 3.2.1 ER Diagram

3.3 DATA DICTIONARY

A data dictionary is a file or a set of files that contains a database's metadata. The data dictionary contains records about other objects in the database, such as data ownership, data relationships to other objects, and other data.

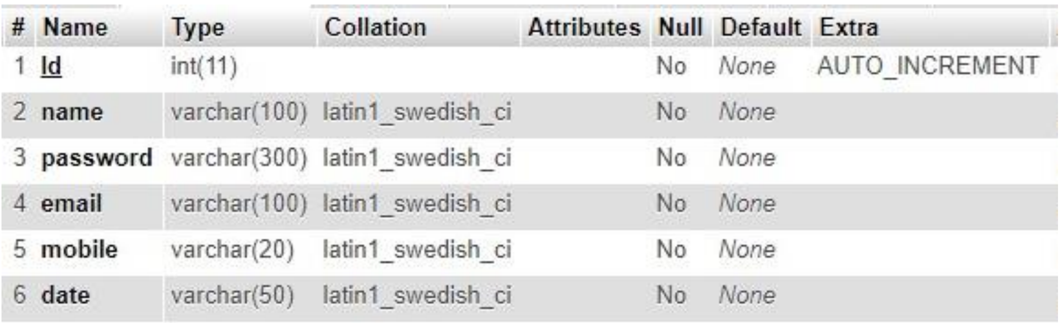
The data dictionary is a crucial component of any relational database. It is invisible to most database users. Typically, only database administrators interact with the data dictionary.

In a relational database, the metadata in the data dictionary includes the following:

- Names of all tables in the database and their owners
- Names of all indexes and the columns to which the tables in those indexes relate
- Constraints defined on tables, including primary keys, foreign-key relationships to other tables, and not-null constraints

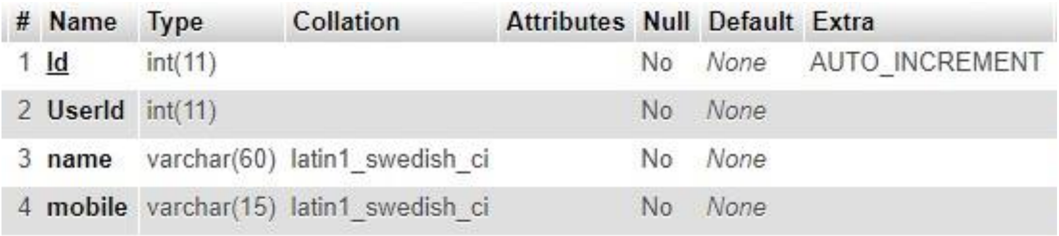
For most relational database management systems (RDBMS), the database management system software needs the data dictionary to access the data within a database.

3.3.1 USER TABLE



#	Name	Type	Collation	Attributes	Null	Default	Extra
1	**Id**	int(11)			No	*None*	AUTO_INCREMENT
2	**name**	varchar(100)	latin1_swedish_ci		No	*None*	
3	**password**	varchar(300)	latin1_swedish_ci		No	*None*	
4	**email**	varchar(100)	latin1_swedish_ci		No	*None*	
5	**mobile**	varchar(20)	latin1_swedish_ci		No	*None*	
6	**date**	varchar(50)	latin1_swedish_ci		No	*None*	

3.3.2 USER CONTACTS TABLE



#	Name	Type	Collation	Attributes	Null	Default	Extra
1	**Id**	int(11)			No	*None*	AUTO_INCREMENT
2	**UserId**	int(11)			No	*None*	
3	**name**	varchar(60)	latin1_swedish_ci		No	*None*	
4	**mobile**	varchar(15)	latin1_swedish_ci		No	*None*	

4. IMPLEMENTATION

The process of putting a decision or plan into effect, execution is implementation. Implementation is the utilization of an application or execution of a plan, idea, model, design, specification, standard, algorithm or policy.

4.1 EXPLANATION OF KEY FUNCTION

4.1.1 XML

Html is a language which is used to create web pages with html marking up a page to indicate its format, telling the web browser where you want a new line to begin or how you want text or images aligned and more are possible.

Some of the tags we used are

4.1.2 FORMS

The purpose of FORM is to create an html form; used to enclose XML, controls, like buttons and text fields.

ATTRIBUTES

- ACTION: Gives the URL that will handle the form data.
- NAME: Gives the name to the form so you can reference it in code set to an alphanumeric string.

METHOD: method or protocol is used to sending data to the target action URL. The GET method is the default, it is used to send all forms name/value pair information in an URL.

Using the POST method, the content of the form are encoded as with the GET method, but are sent in environment variables.

CONTROLS IN XML

< BUTTON>

Creates an html button in a form.

<EDITTEXT>

Creates a password text field, which makes typed input.

< EITTEXT TYPE=RADIOBUTTON>

Creates a radio button in a form.

< BUTTON>

Creates a submit button that the user can click to send data in the form back to the web server.

ATTRIBUTES

NAME: Gives the element a name. Set to alphanumeric characters.

VALUE: Gives this button another label besides the default, Submit Query. Set to alphanumeric characters.

< EDITTEXT >:

Creates a text field that the user can enter or edit text in.

4.2 SAMPLE CODE

```
packagecom.example.womensafetyapp;

importandroidx.appcompat.app.AppCompatActivity;
importandroidx.fragment.app.Fragment;
importandroidx.fragment.app.FragmentManager;
importandroidx.fragment.app.FragmentPagerAdapter;
importandroidx.viewpager.widget.ViewPager;

importandroid.app.AlertDialog;
importandroid.content.Context;
importandroid.content.DialogInterface;
importandroid.content.Intent;
importandroid.content.SharedPreferences;
importandroid.content.res.Configuration;
importandroid.os.Bundle;
importandroid.os.Handler;
importandroid.os.Vibrator;
importandroid.util.Log;
importandroid.widget.Toast;

importcom.google.gson.Gson;

importjava.io.Serializable;
importjava.util.ArrayList;

public class MainActivity extends AppCompatActivity {
publicViewPagerviewPager;
publicAuthenticationPagerAdapterpagerAdapter;
private User user;
SharedPreferencessharedpreferences;
SharedPreferences.Editor editor;
   @Override
protected void onCreate(Bundle savedInstanceState) {
super.onCreate(savedInstanceState);
setContentView(R.layout.activity_main);
sharedpreferences = getSharedPreferences("UserData", Context.MODE_PRIVATE);
editor = sharedpreferences.edit();
Gsongson = new Gson();
        String json = sharedpreferences.getString("User", "");
Log.d("user",json);
user= !json.equals("")?gson.fromJson(json, User.class):new User();
```

```
if(user.isLoggedIn()){
            Intent intent = new Intent(getApplicationContext(), HomeActivity.class);
startActivity(intent);
        }
viewPager = findViewById(R.id.viewPager);
pagerAdapter = new AuthenticationPagerAdapter(getSupportFragmentManager());
pagerAdapter.addFragmet(new login());
pagerAdapter.addFragmet(new register());
viewPager.setAdapter(pagerAdapter);
    }
public void FragmentMethod() {
viewPager.setCurrentItem(0);
    }
public void MoveToHome(String UserData) {
Gsongson = new Gson();
user=gson.fromJson(UserData,  User.class);
user.setLoggedIn(true);
        String json = gson.toJson(user);
editor.clear();
editor.putString("User", json);
editor.commit();
        Intent intent = new Intent(getApplicationContext(), HomeActivity.class);
startActivity(intent);
    }
public void AlertMedthod(String Message, Context context){
AlertDialog.Builder builder = new AlertDialog.Builder(
context,android.R.style.Theme_Material_Light_Dialog_Alert);
builder.setTitle("Alert");
builder.setMessage(Message);
builder.setCancelable(false);
builder.setPositiveButton("Ok", new DialogInterface.OnClickListener() {
            @Override
public void onClick(DialogInterfacedialogInterface, int i) {
dialogInterface.cancel();
            }
        });
AlertDialog alert = builder.create();
alert.show();
    }
    @Override
public void onBackPressed() {
        Intent a = new Intent(Intent.ACTION_MAIN);
```

```
a.addCategory(Intent.CATEGORY_HOME);
a.setFlags(Intent.FLAG_ACTIVITY_NEW_TASK);
startActivity(a);
    }

}
classAuthenticationPagerAdapter extends FragmentPagerAdapter {
privateArrayList<Fragment>fragmentList = new ArrayList<>();

publicAuthenticationPagerAdapter(FragmentManagerfm) {
super(fm);
    }

    @Override
public Fragment getItem(int i) {
returnfragmentList.get(i);
    }
    @Override
publicintgetCount() {
returnfragmentList.size();
    }
voidaddFragmet(Fragment fragment) {
fragmentList.add(fragment);
    }
    }
    packagecom.example.womensafetyapp;

    importandroid.app.Activity;
    importandroid.content.BroadcastReceiver;
    importandroid.content.Context;
    importandroid.content.Intent;
    importandroid.content.SharedPreferences;
    importandroid.os.Build;
    importandroid.telephony.SmsManager;
    importandroid.util.Log;
    importandroid.widget.Toast;

    importandroidx.annotation.RequiresApi;

    importcom.android.volley.Request;
    importcom.android.volley.RequestQueue;
    importcom.android.volley.Response;
    importcom.android.volley.VolleyError;
    importcom.android.volley.toolbox.StringRequest;
    importcom.android.volley.toolbox.Volley;
    importcom.google.gson.Gson;

    importorg.json.JSONArray;
    importorg.json.JSONObject;

    importjava.util.HashMap;
    importjava.util.List;
    importjava.util.Map;

    public class MyReceiver extends BroadcastReceiver {
    staticintcountPowerOff = 0;
    private Activity activity = null;
    privateSmsManagersm;
    RequestQueue queue;
        User user;

    publicMyReceiver(Activity activity,User user) {
    this.activity = activity;
    this.user=user;
    sm=SmsManager.getDefault();
    //        queue = Volley.newRequestQueue(activity.getApplicationContext());
        }

    @RequiresApi(api = Build.VERSION_CODES.M)
        @Override
    public void onReceive(Context context, Intent intent) {
    Log.d("onReceive", "Power button is pressed.");
    //
    //        Toast.makeText(context, "power button clicked", Toast.LENGTH_LONG)
    //                .show();

    //        if (intent.getAction().equals(Intent.ACTION_SCREEN_OFF)) {
    if(countPowerOff>=3){

    GPSTrackermGPS = new GPSTracker(activity);
    mGPS.getLocation();
    if(mGPS.canGetLocation){
    sendSMS(mGPS.getLatitude() + "," + mGPS.getLongitude());
    countPowerOff=0;
```

```
                }
            } else{
countPowerOff++;
            }
//       }
//       else if (intent.getAction().equals(Intent.ACTION_SCREEN_ON)) {
//           if (countPowerOff == 3) {
//           }
//       }

    }
public void sendSMS(String location){
Log.d("loc","calling");
Log.d("loc",location);
        List<Contact> contacts=user.getContacts();
for(int i=0;i<contacts.size();i++){
Log.d("contacts",contacts.get(i).getMobile());
sm.sendTextMessage(contacts.get(i).getMobile(),null,"I'm in danger..My current location
is http://maps.google.com/?q="+location,null,null);
        }
Toast.makeText(activity,"Messages sent",Toast.LENGTH_LONG).show();
//       SMSService(contacts,location);
    }
public void SMSService(final List<Contact>contacts,final String location){
        String url = new Connect().geturl();
StringRequestpostRequest = new StringRequest(Request.Method.POST,
"http://"+url+"/WomenSafety/SendSms.php",
newResponse.Listener<String>()
                {
                    @Override
public void onResponse(String response) {
                        // response
Log.d("Response", response);
if(response.equals("1")){
Toast.makeText(activity,"Messages sent",Toast.LENGTH_LONG).show();
                            }
                    }
                },
newResponse.ErrorListener()
                {
                    @Override
public void onErrorResponse(VolleyError error) {
```

```
                    // error
Log.d("Error.Response", String.valueOf(error));
                }
            }
        ) {
            @Override
protected Map<String, String>getParams()
            {
                Map<String, String>params = new HashMap<>();
params.put("Contacts", new Gson().toJson(contacts));
params.put("Location", location);
returnparams;
            }
        };
queue.add(postRequest);
    }
}
```

5. TESTING

Software testing is a process of executing a program or application with the intent of finding the softwarebugs. It can also be stated as the process of validating and verifying that a software program or application or product: Meets the business and technical requirements that guided its design and development

5.1 TESTING OBJECTIVES

- To ensure that during operation the system will perform as per specification.
- To make sure that system meets the user requirements during operation.
- To make sure that during the operation, incorrect input, processing and output will be detected.
- To see that when correct inputs are fed to the system the outputs are correct
- To verify that the controls incorporated in the same system as intended.
- Testing is a process of executing a program with the intent of finding an error.
- A good test case is one that has a high probability of finding an undiscovered error.

The software developed has been tested successfully using the following testing strategies and any errors that are encountered are corrected and again the part of the program or the procedure or function is put to testing until all the errors are removed.

Note that the result of the system testing will prove that the system is working correctly. It will give confidence to system designer, users of the system, prevent frustration during implementation process etc.,

5.2 DESIGN OF TEST CASES AND SCENARIOS

WHITE BOX TESTING

White box testing is a testing case design method that uses the control structure of the procedure design to derive test cases. All independents path in a module are exercised at least once, all logical decisions are exercised once, execute all loops at boundaries and within their operational bound exercise internal data structure to ensure their validity. Here the customer is given three chances to enter a valid choice out of the given menu. After which the control exists the current menu.

- It is also called as STRUCTURAL TESTING or GLASS BOX TESTING.
- Testers use the knowledge of internal logic of the system.
- Mostly verification techniques are used.
- It does not ensure that the user requirement had been met.
- The test may not mimic the real-world situations.
- Cost is very high since expert testers are required.
- Example: feasibility review, designer review, code inspection and code walk through.

BLACK BOX TESTING

Black box testing attempts to find errors in following areas or categories, incorrect or missing functions, interface error, errors in data structures, performance error and initialization and termination error. Here all the input data must match the data type to become a valid entry.

- It is also called as FUNCTIONAL TESTING. These tests are conducted at interface.
- Testers do not have information about the internal functionality of the system.
- Mostly validation techniques are used.
- It stimulates the actual system usage.
- They have potential of not detecting the logical errors.
- The possibility that efforts are duplicated is high.
- Example: unit testing, integration testing, system testing, and acceptance testing.

The following are the differ MM 2`ent tests at various levels

UNIT TESTING

Unit testing is essentially for the verification of the code produced during the coding phase and the goal is test the internal logic of the module/program. In the Generic code project, the testing is done during coding phase of data entry forms whether the functions are working properly are not. In this phase all the drivers are tested they are rightly connected are not.

The various tests that are conducted during the unit test are described below:

i. Module interfaces are tested for proper information flow in and out of the program.
ii. Local data are examined to ensure that integrity is maintained.

iii. Boundary conditions are tested to ensure that the module operates properly at boundaries established to limit or restrict processing

iv. All the basis (independent) paths are tested for ensuring that all statements in the module have been executed only once.

v. All errors handling paths should be tested.

INTEGRATION TESTING

All the tested modules are combined into sub systems, which are then tested. The goal is to see if the modules are properly integrated, and the emphasis being on the testing interfaces between the modules. In the generic code integration testing is done mainly on table creation module and insertion module.

The integration testing can be carried out using two approaches.

i. The non-incremental integration.

ii. Incremental integration.

VALIDATION TESTING

This testing concentrates on conforming that the software is error-free in all respects. All the specified validations are verified, and the software is subjected to hard-core testing. It also aims at determining the degree of deviation that exists in the software designed from the specification; they are listed out and corrected.

SYSTEM TESTING

This testing is a series of different test whose primary is to fully exercise the computer-based system. This involves:

- Implementing the system in a simulated production environment and testing it.
- Introducing errors and testing for error handling.

Various types of system tests are:

I. **Recovery testing.**

II. **Security testing.**

III. **Stress testing.**

IV. **Performance testing**

Acceptance testing

Once the system tests have been satisfactory completed, the system is ready for acceptance testing. Acceptance testing is the process whereby actual users test a completed information system in the environment where it will eventually be used, the end result of which is the user’s acceptance or rejection. The admin and staff at Company accepted proposed system after testing.

Incremental Integration Testing

Bottom up approach for testing i.e. continuous testing of an application as new functionality is added. This Application functionality and modules are independent enough to test separately. The functionality like view/create group, view,/edit profile, view/post notification, view/post news and add/show event are independent to each other. These functionalities are added separately and tested after the implementation of each. The distributed nature of client/server systems pose a set of unique problems while conduct tests we noted following areas while testing:

- Client GUI considerations
- Target environment and platform diversity considerations
- Distributed processing considerations
- No robust target environment

Nonlinear performance relationships Many different types of tests are conducted at each of this level of detail the following tests are conducted

Application function tests

The functionality of client applications is tested using the methods discussed below.

Server tests

The coordination and data management functions of the server are tested. Server performance is also considered.

Database tests

The accuracy and integrity of data stored by the server is tested. Transactions posted by client applications are examined to ensure that data are properly stored, updated and retrieved. Archiving is also tested.

Network communication tests

These tests verify that communication among the nodes of the network occur are correct and that message passing, transactions and related network traffic occurs without error. Networks tests are also being conducted. The strategy for testing c/s architecture is analogous to testing of other architecture

5.3 VALIDATION

From various types of testing, we need very few methods of testing for database validation. Those are listed below.

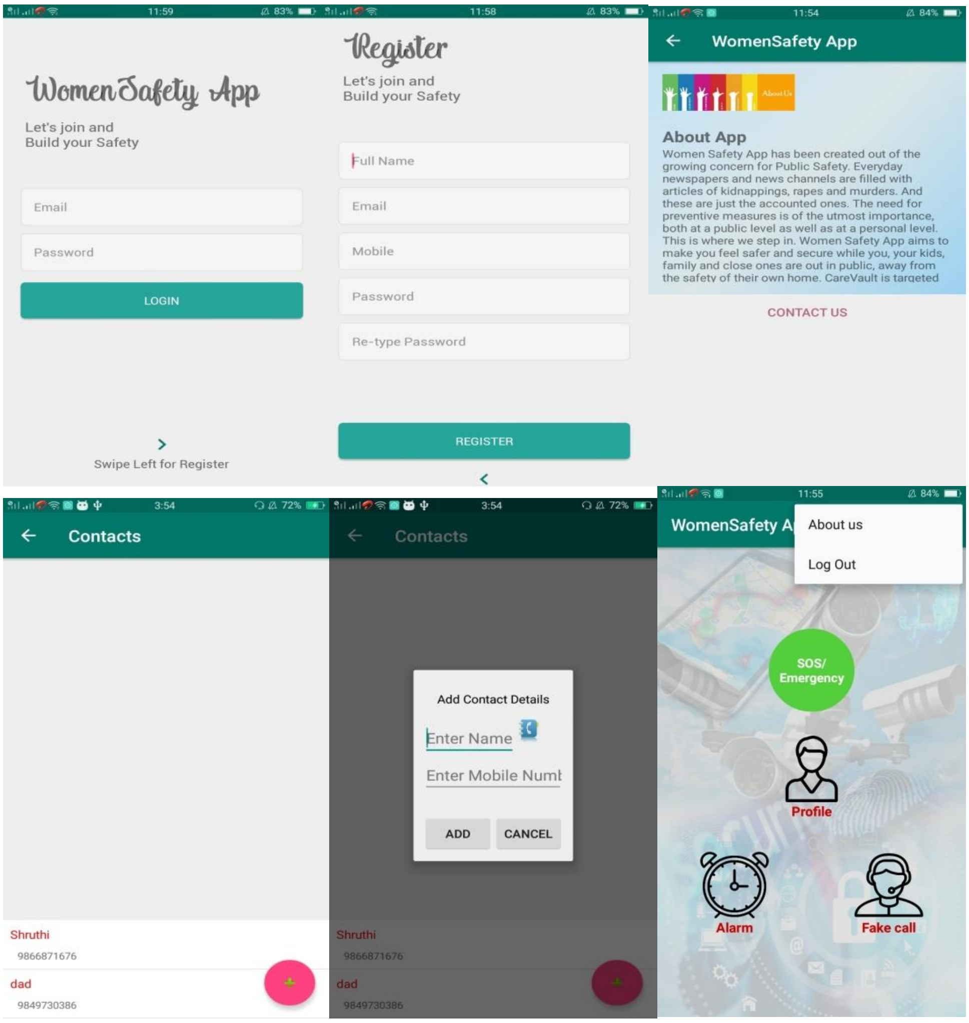
Interfaces consistency should be validated in order to guarantee that applications have a stable structure for data access. This testing is required for the mobile store management system application to ensure we have a properly constructed database and are concentrated properly.

Data availability and authorization tests are similar to interface consistency tests, but are more focused on who can get data from the database than how the data should be retrieved. This testing is also required for the application in order to assure data connectivity.

TEST CASES

Sl.no	Test case Name	Test Procedure	Pre Condition	Post Condition	Status
1	User login case	Is user provided true username and password	Redirect to next screen	Redirect to next screen	Success
2	Invalid username and password case	Is username and password wrong then	Please enter correct username and password	Please enter correct username and password	Fail
3	Add Contact Case	If null enter in mandatory fields	Please fill out the field	Please fill out the field	Fail
	If the data entered is correct	Added successfully	If the data entered is correct	If the data entered is correct	Success
4	View Contact screen case	Is represented page available	Successfully view that screen	Successfully view that screen	Success
5	Contact add	If null enter in mandatory fields	Contact added successful	Contact sent successful	Success

6. RESULTS



About App

Women Safety App has been created out of the growing concern for Public Safety. Everyday newspapers and news channels are filled with articles of kidnappings, rapes and murders. And these are just the accounted ones. The need for preventive measures is of the utmost importance, both at a public level as well as at a personal level. This is where we step in. Women Safety App aims to make you feel safer and secure while you, your kids, family and close ones are out in public, away from the safety of their own home. CareVault is targeted

Shruthi
9866871676

dad
9849730386

7. CONCLUSION

7.1 CONCLUSION

The “Women Safety App” has been developed to satisfy all the proposed requirements. The process is maintained more simple and easy in ensuring the women safety. The system is highly scalable and user friendly. Almost all the system objectives have been met. The system has been tested under all criteria. The system minimizes the problem arising in the existing manual system and it ensures the immediate action to be taken when an unfavourable situation is encountered. The design of the database is flexible ensuring that the system can be implemented. It is implemented and gone through all validation. All phases of development were conceived. User with little training can get the required report. The software executes successfully by fulfilling the objectives of the project. Further extensions to this system can be made required with minor modifications.

7.2 FUTURE ENHANCEMENTS

This mobile application is helpful in future when any problem arises in travelling or any kind of situations. As the technology emerges, it is possible to upgrade the system and can be adaptable to desired environment

- Currently we are creating android app in future we will develop ios app.
- Based on the future security issues, security can be improved using emerging technologies.
- We are going to implement a fake call module, where we can get an incoming call by clicking the module.

7.3 REFERENCES

1. Pasha S., Kavana J., Mangala G.K.R., Nischitha K., Surendra B.K., Rakshitha M.S. (2016). BSecure for women: an android application, International Journal of Innovative Research in Computer and Communication Engineering, Vol. 4, No. 5, pp. 8073- 8080.
2. Saranya N., Karthik K. (2015). Women safety application using android mobile, International Journal of Engineering Science and Computing, pp. 1317-1319.
3. Pawar V., Wankhade N.R., Nikam D., Jadhav K., Pathak N. (2014). SCIWARS android app for women safety, International Journal of Engineering Research and Application, Vol. 4, No. 3 (Version 1), pp. 823- 826.
4. Mandapati S., Pamidi S., Ambati S. (2015). A mobile based women safety application (I safe apps), IOSR Journal of Computer Engineering, Vol. 17, No. 1 (Version 1), pp. 29-34.
5. Uma D., Vishakha V., Ravina R., Rinku B. (2015). An android application for women safety based on voice recognition, International Journal of Computer Science and Mobile Computing, Vol. 4, No. 3, pp. 216-220.
6. Abhaya: An Android App for the safety of women by Ravi ShekharYarabothu at 12th IEEE International Conference 2015
7. BSAFE-PERSONAL SAFETY APP, Android app developed by Bipper. Inc., March 6, 2015 http://getbsafe.com/
8. A Mobile Based Women Safety Application (I Safe Apps)by Dr. Sridhar Mandapati1 , Sravya Pamidi2 , SriharithaAmbati IOSR Journal of Computer Engineering (IOSR-JCE) eISSN: 2278-0661,p-ISSN: 2278-8727, Volume 17,Issue 1, Ver. I (Jan Feb. 2015).

8. APPENDIX

8.1 ABBREVATIONS

S.NO	ABBREVATION	EXPANSION
1.	UML	Unified modeling language
2.	XML	Extensible markup language
3.	SMS	Short message service
4.	GPS	Global positioning system
5.	PHP	Hypertext preprocessor
6.	GSM	Global system for mobile communications
7.	ERD	Entity relationship diagram

8.2SOFTWARE INSTALLATION PROCEDURE

- **ANDROID INSTALLATION:**

Before downloading and installing Android Studio, the following requirements are essential.

- Operating System Version - Microsoft Windows 7/8/10 (32-bit or 64-bit).
- Random Access Memory (RAM) - Minimum 4 GB RAM and 8 GB RAM recommended.
- Free Disk Space - Minimum 2 GB and 4 GB recommended.
- Minimum Required JDK Version - Java Development Kit (JDK) 8.
- Minimum Screen Resolution - 1280 * 800.resolution

Step 1

To download the Android Studio, visit the official Android Studio website in your web browser.

Step 2

Click on the "Download Android Studio" option.

Step 3

Double click on the downloaded "Android Studio-ide.exe" file.

Step 4

"Android Studio Setup" will appear on the screen and click "Next" to proceed.

Step 5

Select the components that you want to install and click on the "Next" button.

Step 6

Now, browse the location where you want to install the Android Studio and click "Next" to proceed.

Step 7

Choose a start menu folder for the "Android Studio" shortcut and click the "Install" button to proceed.

Step 8

After the successful completion of the installation, click on the "Next" button.

Step 9

Click on the "Finish" button to proceed.
Now, your Android studio welcome screen will appear on the screen.

Android Studio Setup Configuration

Step 10

"Android Studio Setup Wizard" will appear on the screen with the welcome wizard. Click on the "Next" button.

Step 11

Select (check) the "Standard" option if you are a beginner and do not have any idea about Android Studio. It will install the most common settings and options for you. Click "Next" to proceed.

Step 12

Now, select the user interface theme as you want. (I prefer Dark theme (Dracula) that is most liked by the coders). Then, click on the "Next" button.

Step 13

Now, click on the "Finish" button to download all the SDK components.
And, the downloading and installation process of components gets started.

Step 14

After downloading all the necessary components, click on the "Finish" button.

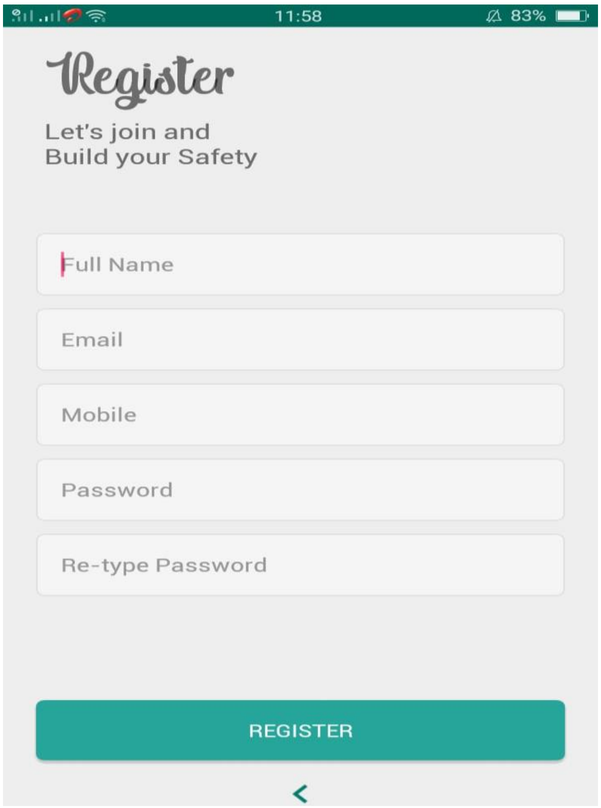
- **XAMPP INSTALLATION:**

https://www.ionos.com/digitalguide/server/tools/xampp-tutorial-create-your-own-local-test-server/

8.3 SOFTWARE USAGE PROCESS

The following steps describe the procedure to use the software

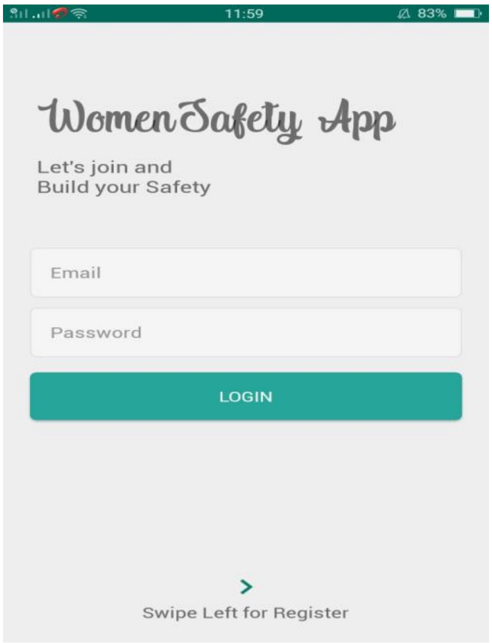
1.Firstly, register to the app



Register

Let's join and
Build your Safety

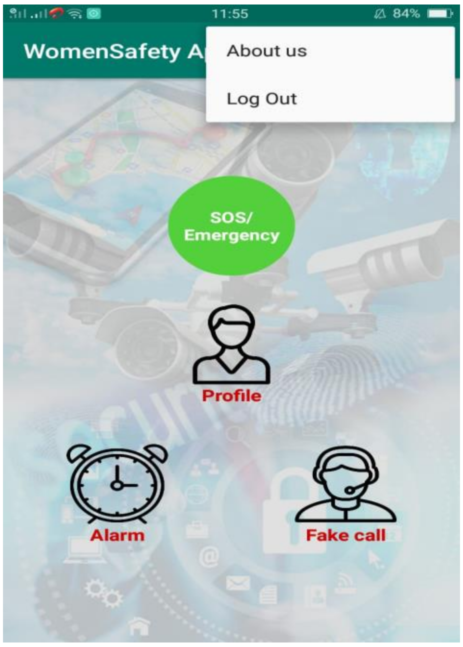
1. After becoming registered user, you can login to the app by using username and password.



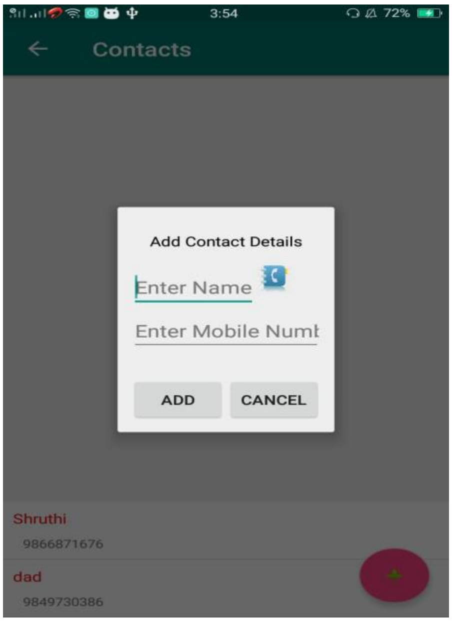
WomenSafety App

Let's join and
Build your Safety

2. We can see a home page screen after successful registration.



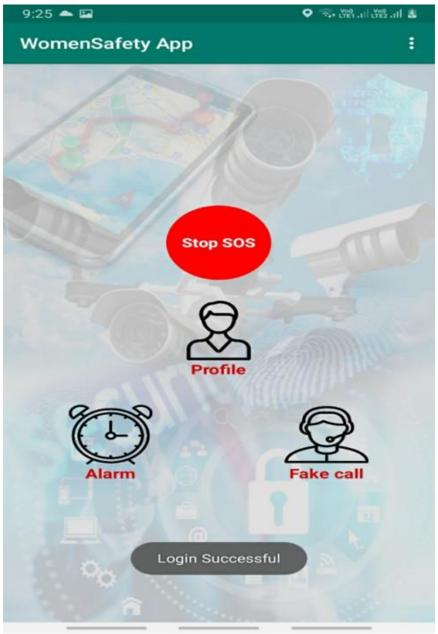
3. We can register some numbers, to whom the emergency message as to be sent.



Shruthi
9866871676
dad
9849730386

4. When we click on alarm icon, we can hear an alarm sound so that other people around may hear.

5. If power button doesn’t work, then we need to manually click on SOS/emergency button, to send sms to the registered numbers with exact location.


9:25
WomenSafety App
Stop SOS
Profile
Alarm
Fake call
Login Successful