

**Tribhuvan University**

**Faculty of Humanities and Social Sciences**

Smart contact Manager

# A PROJECT REPORT

**Submitted to Department of Computer Application**

**Damak Multiple College**

***In partial fulfillment of the requirements for the Bachelors in Computer Application***

Submitted by **Sandip chapagain[]**

**2020523**

June, 2022

Under the Supervision of

**Mr. Nabin Khatiwada**



**Tribhuvan University**

**Faculty of Humanities and Social Sciences**

**Damak Multiple College**

**Supervisor’s Recommendation**

I hereby recommend that this project prepared under my supervision by **Sandip Chapagain** entitled “**Smart Contact Management.**” in partial fulfillment of the requirements for the degree of Bachelor of Computer Application is recommended for the final evaluation.

**Mr. Nabin khatiwada**

**SUPERVISOR**

Damak Multiple College

Department of BCA



**Tribhuvan University**

**Faculty of Humanities and Social Sciences**

**Damak Multiple College**

## LETTER OF APPROVAL

This is to certify that this project prepared by **Sandip Chapagain** entitled “**Smart Contact Management System**” in partial fulfillment of the requirements for the degree of Bachelor in Computer Application has been evaluated. In our opinion, it is satisfactory in the scope and quality as a project for the required degree.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Mr.  External Examiner |  | | | | | | |
| Tribhuvan University |  | Signature |  |  |  | Date |  |
| Mr. Abinash Jha HOD/Coordinator |  |  |  |  |  |  |  |
| DMC |  | Signature |  |  |  | Date |  |

Mr. Nabin Khatiwada

Supervisor

DMC Signature Date

# Acknowledgement

This project is prepared in the partial fulfillment of the requirement for the degree of Bachelor in Computer Application (BCA). The satisfaction and success of completion of this task would be incomplete without heartfelt thanks to people whose constant guidance, support and encouragement made this work successful. On doing this undergraduate project we have been fortunate to have help, support and encouragement from many people we would like to acknowledge them for their cooperation.

Our first thanks goes to Tribhuvan University for designing such a worthy syllabus and making us do this project. Our next batch of thanks goes to the faculty of Management of DMC without whose help our project would have been impossible. This list includes chief of DMC, **Mrs. Netra Budhathoki** Our very sincere and heartfelt thanks go to **Mr. Nabin Khatiwada** our project supervisors who constantly guided us through the project time period. Without his guidance, our project would have been impossible. Last but not the least we want to thank every direct and indirect hands that were involved in completion of this project.

This project has been a wonderful experience where we have learnt and experienced many beneficial things.

**Sandip chapagain[ ]**

# Abstract

The main aim of this project is to provide simple platform to the User to Save Their contact details of relatives or staff of the company. The proposed of the systems is simplify store the information of the people and. It provides the backup of the contact details of useful people or staff of the company The proposed system was designed and implemented using Java,js,html,css and MySQL

Table of Contents

Acknowledgement………………………………………………..i

Abstract………………………………………………………......ii

[Chapter 1: Introduction 1](#_Toc82805525)

[1.1. Introduction 1](#_Toc82805526)

[1.2. Problem Statement 1](#_Toc82805527)

[1.3. Objectives 1](#_Toc82805528)

[1.4. Scope and Limitations 2](#_Toc82805529)

[1.5. Report Organization 2](#_Toc82805530)

[Chapter 2: Background Study and Literature Review 2](#_Toc82805531)

[2.1. Background Study 2](#_Toc82805532)

[2.2 Literature Review 3](#_Toc82805533)

[Chapter 3: System Analysis and Design 3](#_Toc82805534)

[3.1 System Analysis 3](#_Toc82805535)

[3.1.1 Requirement Analysis 4](#_Toc82805536)

[1. Functional Requirements 4](#_Toc82805537)

[2. Non-Functional Requirements 4](#_Toc82805538)

[3.1.2 Feasibility Study 4](#_Toc82805539)

[3.1.2.1 Operational Feasibility 4](#_Toc82805540)

[3.1.2.2 Economic Feasibility 5](#_Toc82805541)

[3.1.2.3 Technical Feasibility 5](#_Toc82805542)

[3.1.2.4 Time Feasibility 5](#_Toc82805543)

[3.1.2.5 Schedule Feasibility 6](#_Toc82805544)

[3.1.3Data Modeling (ER- Diagram) 7](#_Toc82805545)

[3.1.4 Process Modeling (DFD) 7](#_Toc82805546)

[3.1.4.1 Zero Level DFD 7](#_Toc82805547)

[3.1.4.2 One Level DFD 8](#_Toc82805548)

[3.1.4.3 Two Level DFD 9](#_Toc82805549)

[3.2 System Design 10](#_Toc82805550)

[3.2.1 Architectural Design 10](#_Toc82805551)

[3.2.2 Database Schema Design 11](#_Toc82805552)

[3.2.3 Interface Design 12](#_Toc82805553)

[3.2.4 Physical DFD 13](#_Toc82805554)

[Chapter 4: Implementation and Testing 14](#_Toc82805555)

[4.1 Implementation 14](#_Toc82805556)

[4.1.1 Tools Used 14](#_Toc82805557)

[Drawing tools 14](#_Toc82805558)

[Programming Languages 14](#_Toc82805559)

[4.1.2 Implementation details of modules 15](#_Toc82805560)

[4.2 Testing 16](#_Toc82805561)

[4.2.1 Test cases for unit testing 16](#_Toc82805562)

[4.2.2 Test Cases for System Testing 18](#_Toc82805563)

[Chapter 5: Conclusion and Future Recommendation 19](#_Toc82805564)

[5.1 Lesson Learnt/Outcome 19](#_Toc82805565)

[5.2 Conclusion 19](#_Toc82805566)

[5.3 Future Recommendations 19](#_Toc82805567)

[Appendices 20](#_Toc82805568)

[Screenshots 20](#_Toc82805569)

[References 21](#_Toc82805570)

[Websites: 31](#_Toc82805571)

[Books: 31](#_Toc82805572)

Table of figures

[Figure 1: Gantt Chart 6](#_Toc82808516)

[Figure 2: E-R Diagram 7](#_Toc82808517)

[Figure 3: Zero Level DFD 7](#_Toc82808518)

[Figure 4: One Level DFD 8](#_Toc82808519)

[Figure 5: Two Level DFD 9](#_Toc82808520)

[Figure 6: Architectural Design 10](#_Toc82808521)

[Figure 7: Schema Diagram 11](#_Toc82808522)

[Figure 8: User Interface Structure Diagram 12](#_Toc82808523)

[Figure 9: Physical DFD 13](#_Toc82808524)

**List of tables**

[Table 1: Login module test 16](#_Toc82808528)

[Table 2:Contact module test 16](#_Toc82808529)

[Table 3: Account info module test 18](#_Toc82808530)

[Table 4: System Test 18](#_Toc82808532)

# Chapter 1: Introduction

## Introduction

Smart Contact manager is the Web application software to store the Contact records of a user or company. It maintains the records in systematic and scientific procedures. Efficiency is the core thing of this software.

This project will play a vital role to keep customer detail it helps for a company fully digital and well organized. The main aim of this project is to keep or store the contact details of the user or employee of the company..

Smart Contact manager System is the web application software use to store contact detail. Smart Contact Manager System is a web application to maintain employee contact detail. This software help to register user details as well as company employee contact details etc.,  

Problem Statement

Traditional method of management is the main problem of the today modern world. This is the age of modern computers and digitalization, so all the system present are digital. According to the modern world requirement, online system and application software are mostly used. The main problems are listed below:

* Traditional methods
* Much time consuming to keep record in book-keeping system
* Slow search of record in bookkeeping system
* High chance of data lose

## Objectives

The objectives of project are as follows:

* To keep the contact detail of the company’s employee or user’s relative.
* Multiple user can access
* Operated from anywhere from any device

## Scope and Limitations

Smart Contact Manager system is the web application software designed and developed for store the contact details of the company’s staff. Its scopes are given below:

* Keep contacts details of staff of company or user’s relative

Limitations of this application software are listed below:

* Internet Reliance
* Security
* Reduced Speed
* Browser Support
* No GPS Tracking

## Report Organization

**Chapter 1:** Introduction of the project along with project scope limitations and objectives are described.

**Chapter 2:** Background study related to the project along with general descriptions of project functions and components. Literature review in order to have broader understanding of the project concepts based on research done previously and analyze similar systems for comparison with project.

**Chapter 3**: System Analysis and Design of the system using various charts and figures. Functional requirements defined using use cases and other techniques. Database schema, interface design and deployment diagram are included.

**Chapter 4:** Tools and techniques used for project implementation along with algorithms used in the project and creation of test cases to test the system as unit and as a whole. **Chapter 5:** Lessons learn from start to finishing the project, future recommendations for other projects and project conclusion.

# 

# Chapter 2: Background Study and Literature Review

## 2.1. Background Study

In this 21st century most of the people are familiar with technology most of the people are depend on technology especially Transportation and communication are the most two very useful in our day to day life .obviously communication play significant role in our daily life .We know that , basically in most of the office or company the role of manager is to keep all the staff contact details ,to maintain the communication between staff so it should necessary to store the all the staff contact details .In previous days people use to store the contact details of their relative in notebook .. Searching contact details of any particular person becomes lengthy and time consuming Paper can be damaged or note book can be loss , now to reduce this problem my project "smart contact manager" was introduce

Now Mobile phone is introduce to store the contact details as well as for phones calls .in some situation mobile phone is not much efficient ,may be it would loss/ battery low/ or some technical issues take place so to overcome the ..my project " smart contact manager " was introduce it can store much more information of staff , Any user can easily store and search specific persons details with in minimum time , including high level security cannot be access by unauthorized persons ,multiple user can access the system from any environment

## 2.2 Literature Review

All the company have old paper filing method to maintain the records of contact details as well as image, description . Record Book are maintained but it is not good to keep record in Record book .because of data loss and searching a particular person takes more time. So, it is better to store details of the employee in cloud. Also, Low level of security can cause the fraud issue as well as loss of actual record.

# 

# Chapter 3: System Analysis and Design

## 3.1 System Analysis

System analysis is the process of studying the each and every thing of a system in detail way. Critical analysis is done to know the activities of the system. It helps in the perspective of developing of the required software. Smart Contact Manager system is analyzed in various perspectives to get the required output. It helps to implement the software easily and efficiently.

## 3.1.1 Requirement Analysis

## 1. Functional Requirements

These are the major requirements of the software. User defined requirements are called functional requirements. In this project, functional requirements are:

* To keep Description of the person (notepad).
* To store image of the person.
* Each user has profile base system.
* Searching functionality.
* To display details of the employee.

## Non-Functional Requirements

These are the requirement that user don't define but should be there in software. In this project, non-functional requirements are:

* Validations on different fields
* Availability
* Security
* Performance
* Reliable
* Efficient

## 3.1.2 Feasibility Study

Feasibility study is the process of determining the necessary requirements for the developer. It determines how much time does the project needed to complete it through available resources. It is done before the starting phase of project. Possibility of project is determined here by studying operational, economic, technical, time and schedule feasibility. Comparison of real work and logical work is done to achieve the best result of the project development with systematic and scientific as well as efficient way.

### 3.1.2.1 Operational Feasibility

The manpower in the developing site is skilled enough to perform the general operations. Every problem can be handled with ease by performing the necessary operations required. Testing and redefining can be done by manpower of the project. Operational issues can be fixed easily. Quality is maintained by operating the system testing.

### 3.1.2.2 Economic Feasibility

Economic Feasibility is directly determined by calculating the total cost required for the development of the project. The project will be economically feasible if the development cost can be bear by developer in development phase. This project is simple and easily the cost of development can be bear in development phase. So this project is economically flexible.

### 3.1.2.3 Technical Feasibility

During the study of project the technical requirements are collected to know its feasibility. It means manpower are used to or not for the technology used to in this project. This project is built on SPRING BOOT FRAMEWORK as front and MYSQL as Database. So, developers are used to for these programming languages and available at developing site. So this project is technically feasible.

### 3.1.2.4 Time Feasibility

This project will be completed in given time period. We know it by studying the above mentioned feasibility study. As developer focuses on this project to give customer satisfaction, it is feasible respect to time.

### 3.1.2.5 Schedule Feasibility

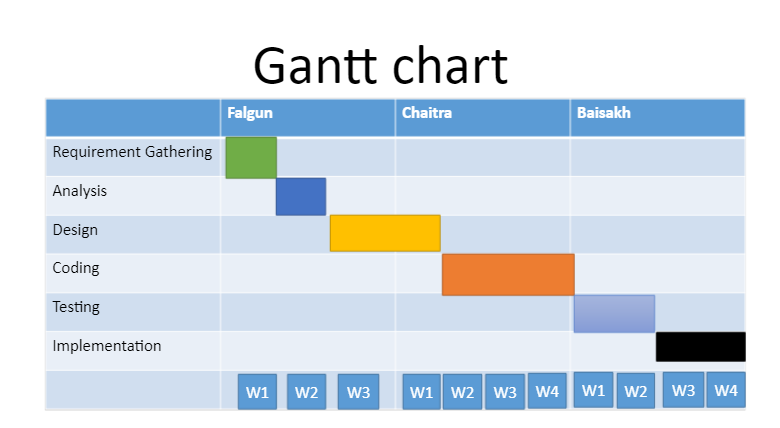


Figure 1: Gantt Chart

## 3.1.3Data Modeling (ER- Diagram)

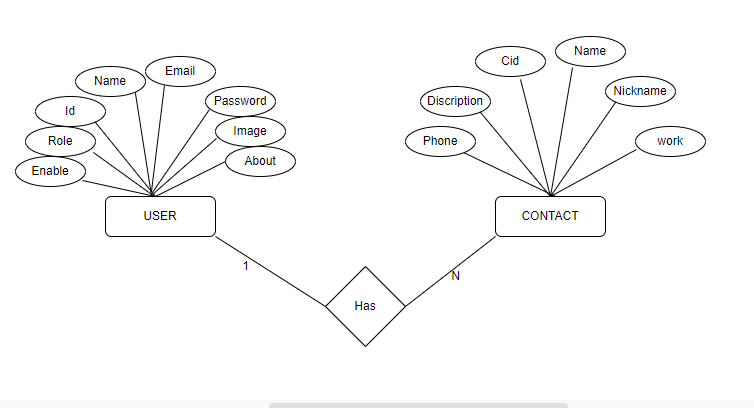


Figure 2: E-R Diagram

## 3.1.4 Process Modeling (DFD)

### 3.1.4.1 Zero Level DFD

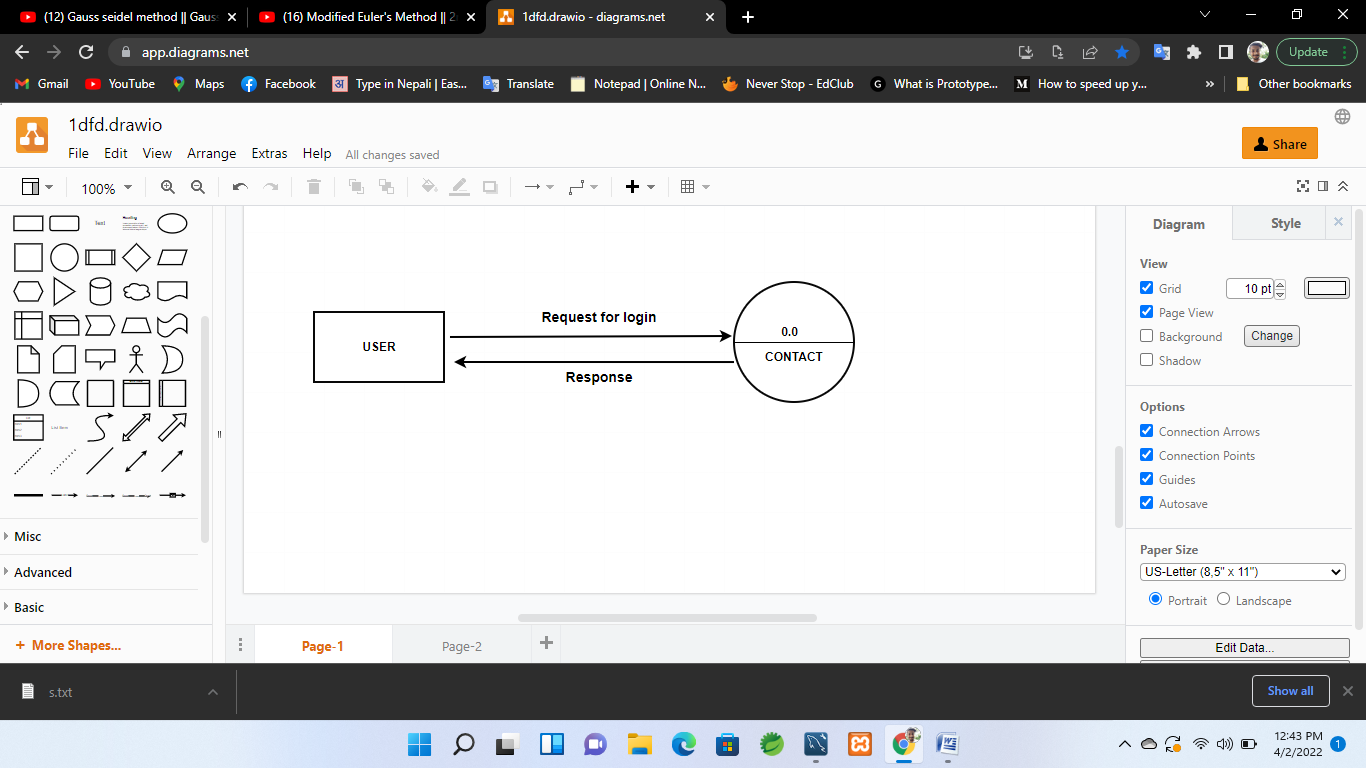


Figure 3: Zero Level DFD

### 3.1.4.2 One Level DFD

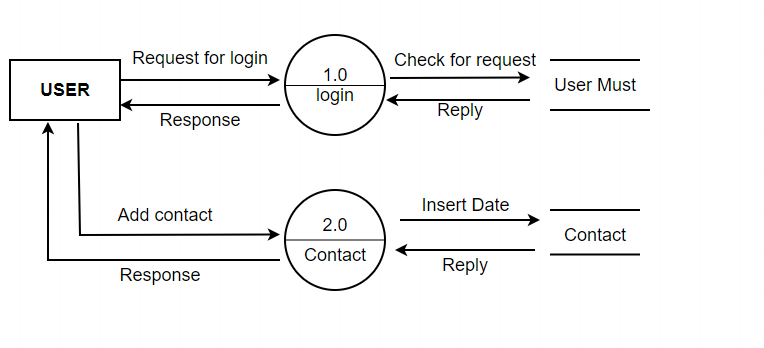


Figure 4: One Level DFD

### 3.1.4.3 Two Level DFD

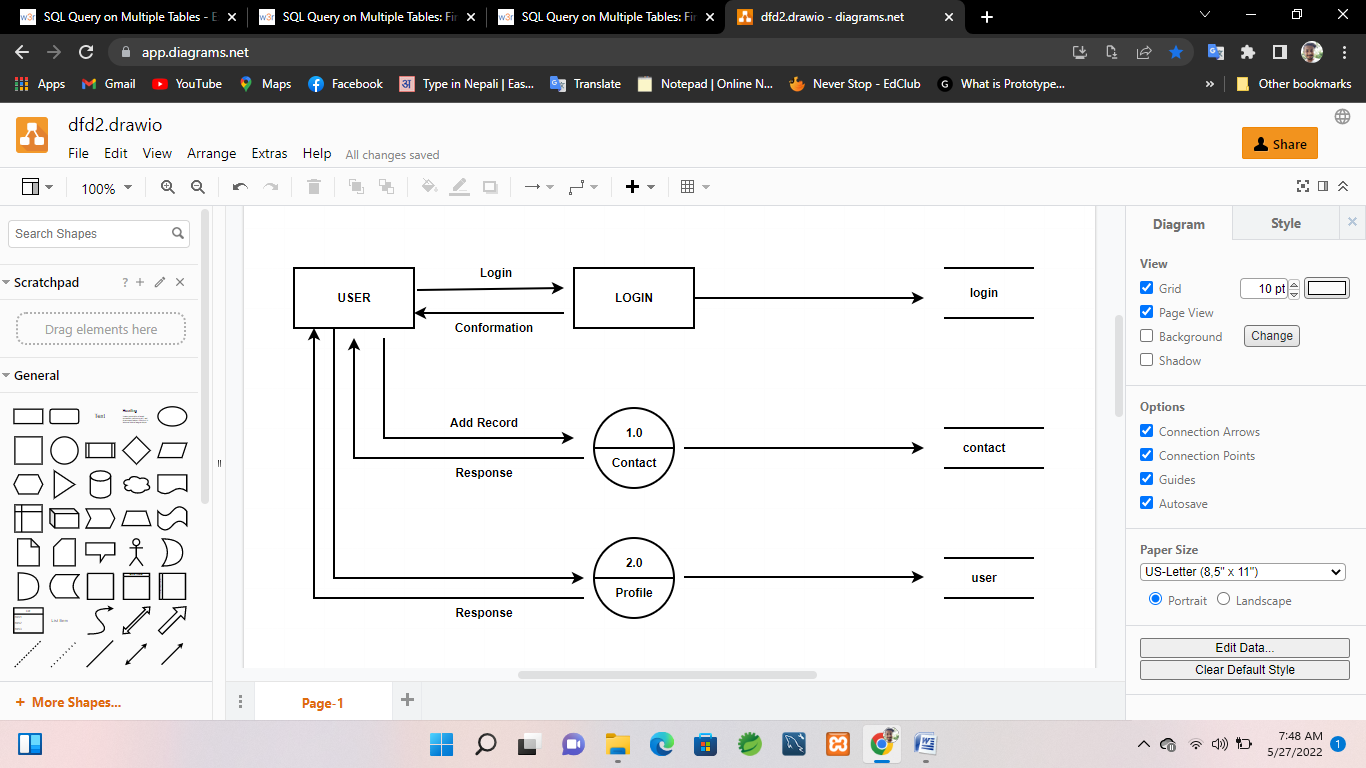


Figure 5: Two Level DFD

## 3.2 System Design

System design is the process of structuring the planned system. It is the way to visualize the concepts of system that how it works. It defines the graphical format or representation of system model process. To have complete design of project we must have architectural design, database schema design, interface design and physical DFD.

### 3.2.1 Architectural Design

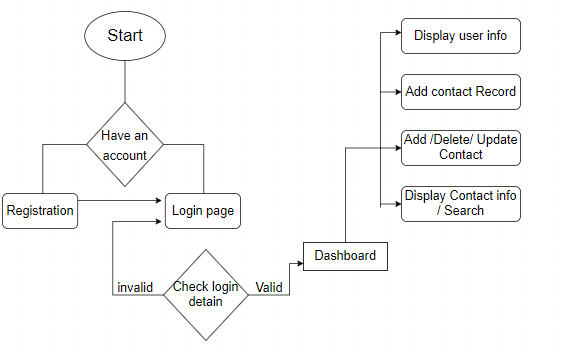


Figure 6: Architectural Design

### 

### 3.2.2 Database Schema Design

Schema is the collection of database created for the project. It determines the simplest form of data flow inside the project. It contains the databases and tables inside the database. It covers the whole project diagram of data flow.

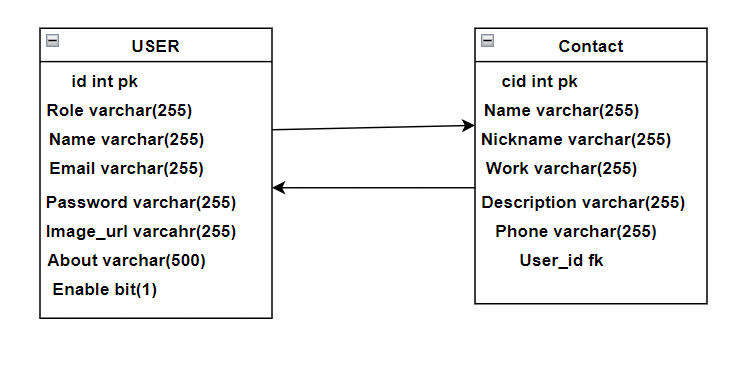


Figure 7: Schema Diagram

### 3.2.3 Interface Design

Interface is the medium to interact with software. Interface should be designed in such a way that user must feel comfortable to feel it and contains the most of necessary information. Interface Structure Design is the process of structuring the interface in raw format. It is not the actual interface but it is the frame to develop the interface.

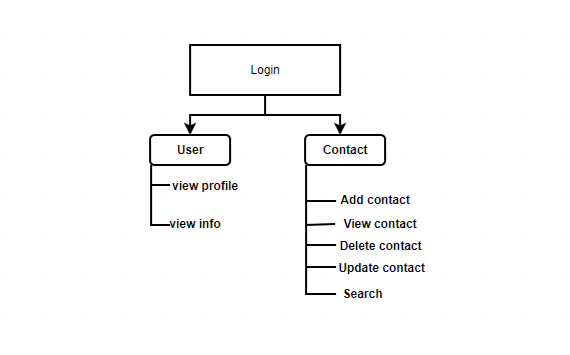


Figure 8: User Interface Structure Diagram

### 3.2.4 Physical DFD

A physical data flow diagram show how the system will be implemented, including the hardware, shftware, files , and people in the system. It is developed such that the processes descibed in the logical data flow diagrams are implemented correct to achieve the goal of the business.

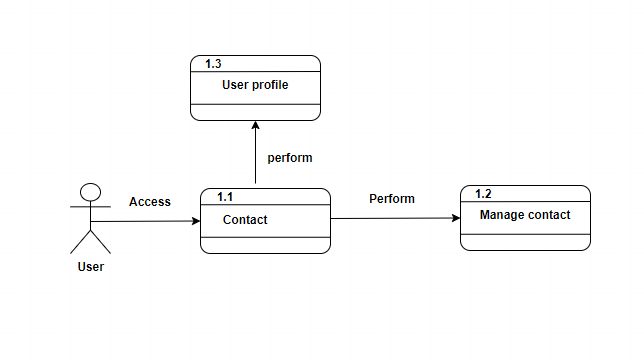


Figure 9: Physical DFD

# Chapter 4: Implementation and Testing

After planning and designing phase we must implement the project and test it accordingly. There are different types of testing; unit testing and system testing.

## 4.1 Implementation

Implementation is the process of writing codes for the designed system to run in the hardware over the operating system. It consist the techniques and tools to write the program and use it. We implement the developed software here and check it. Modules are created and implemented by unit to unit. After the completion of whole project then only whole project is implemented.

### 4.1.1 Tools Used

Different types of tools are used while implementing the project; CASE Tools, Programming Tools and Database Platforms. CASE tools are used to design the system as well. To generate the code automatically we use code generator. In this project we use the following tools, languages and database platforms.

### Drawing tools

We use draw.io (Draw) to draw the different figures while designing the system. We draw different DFD and ER Diagrams, schema diagrams, interface diagrams, architectural diagrams and physical DFD. It makes the project easy to represent in graphical format.

### Programming Languages And Dependencies

This project is developed in spring boot framework. Html and bootstrap for front end and JAVA is for back end. I use JRE v1.8 to compile the program and run. This is web application software developed for general purpose of users. SpringToolSuite is used as Java Development Kit. Window builder is used to design the graphical user interface and that automatically generate the code makes easy to program.

For database, we use MYSQL Database version 8.0.26 to store the data in backend. It has its own database server to serve the data according to the request sent by the user using JDBC driver of JAVA programming language.

Dependencies which are used in the project:-

* Spring-boot-stater-data-spa
* Spring-boot-starter-web
* Spring-boot-starter-web
* Spring-boot-starter-thymeleaf
* Mysql-connecter-java
* Validator-api
* Hibernate-validator
* Spring-boot-devtools
* bootstrap

### 4.1.2 Implementation details of modules

There are many panels used in this Web application program for user convenience. Those panels have different functions inside it. There are total nine modules in this project. They are:

1. Contact module:

It provides the interface to enter the contact record of the user. It also display the record enter and helps search of different contact .

1. Dashboard Module:

It displays the buttons to navigate to any another modules and act as Home page of software.

1. Login Module:

It is the module used to get inside the application software. It requests the user to input the user name and password to get access inside the software.

1. Database Module:

It is the class to establish the connection between MYSQL server and user interface or application. It is used all over the program to make connection to database.

## 4.2 Testing

Testing is the process of determining the faults of programs. It is actually done to check different conditions and scenarios that may occur in the program while performing any kind of operation. It is used also for quality assurance. All the programs that are developed should be tested properly. So, testing takes long time rather than development. Critical condition should be applied while testing to make software run in critical condition. There are different types of testing. Here we perform only two types of testing; unit testing and system testing.

### 4.2.1 Test cases for unit testing

It is basically the testing of modules inside the project. Certain cases are test under it to make sure of same problem overlaps or not.

Some test cases are given below:

Login Module:

Table 1: Login module test

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Test ID** | **Test Scenario** | **Test Steps** | **Test Data** | **Expected Result** | **Actual Result** | **Pass/Fail** |
| T01 | Login by  User | Enter username and password | username: ngarc  Passowrd: sandip@123 | Dashboard should be opened | As Expected | Pass |
| T02 | Login By other with wrong credentials | Enter username and password | username: ngarc  Passowrd: 12345 | Display message and stay in same page | As Expected | Pass |

Contact Modules:

Table 2: Contact module test

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Test ID** | **Test Scenario** | **Test Steps** | **Test Data** | **Expected Result** | **Actual Result** | **Pass/Fail** |
| T03 | Adding contact records | Enter name  nickname  phone num  Email  work  description | sandip  sndp  9800000000  Sndp1@g.c  Doctor  Good doc | New record should be added and displayed in table | As excepted | Pass |
| T04 | Adding  Contact records by blank fields | Enter name  nickname  phone num  Email  work  discription | (blank)  ramu  98000000  Ramu1@  business  ramu busin.. | Dialog box should be displayed with "please enter your name" | As excepted | Pass |
| T05 | Searching details by any credentials of contact record | Enter anything that to be searched from contacts T06record | Sandip | Display all the records where name= sandip | As excepted | Pass |
| T06 | Deleting the selected  contact | select the contact to be deleted | sndip  sndp  98000000  sndp@12  doctor  sndp good doctor | Deleted record should be removed from table | As  Excepted | Pass |
| T07 | Updating the  Selected  Contact | Select the contact to be edit | Ram  Ramu  98000000  ram@12  doctor  sam good doctor | Updated record should be displayed over the previous record | As  Excepted | Pass |

Account info module test:

Table 3: Account info Module Test

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Test ID** | **Test Scenario** | **Test Steps** | **Test Data** | **Expected Result** | **Actual Result** | **Pass/Fail** |
| T12 | Account Register | Name  email  password  about  Image | sandip  sandp@13  pas@123  jnsja  img | On clicking profile display the  Profile of each user | As expected | Pass |

### 4.2.2 Test Cases for System Testing

System testing is the process of checking the whole system compatibility. It is done after completing the final project after unit testing. After integrating all the units in system the testing is done to find out whether it could run or not. There are some cases of system testing are:

Table 4: System Test

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Test ID | Test Scenario | Test Steps | Test Data | Expected result | Actual Result | Pass/ Fail |
| ST01 | * valid login credentials * loading dashboard * loading different windows on clicking button on dashboard * displaying accurate data in tables * after clicking back button loading dashboard and dispose the current page | Enter valid login Credentials  Check dashboard  Check Different windows  Check Table data  Check Back button | Valid details  click  click  click  click | Login successful  load dashboard  load different windows  display accurate data  redirect to dashboard window | As excepted | Pass |

# Chapter 5: Conclusion and Future Recommendation

## 5.1 Lesson Learnt/Outcome

While developing this project a lot of things happened. Different types of problems arises, challenges faced and a lot of hard work is done. From zero level to high level of development activities; lot of experience is gained. Some of the lessons learn are given below:

* High level of analysis is necessary to design and develop the project
* Accurate requirements should be collected to get the maximum productivity
* Simple and clear design should be done
* Good approach of programming should be followed to make it reliable
* Professional skills are most important factor

Some outcomes of the project are given below:

* It consume less time in record keeping
* Searching becomes more easier
* Easy to get accurate data
* Easy to maintain the contact details

## 5.2 Conclusion

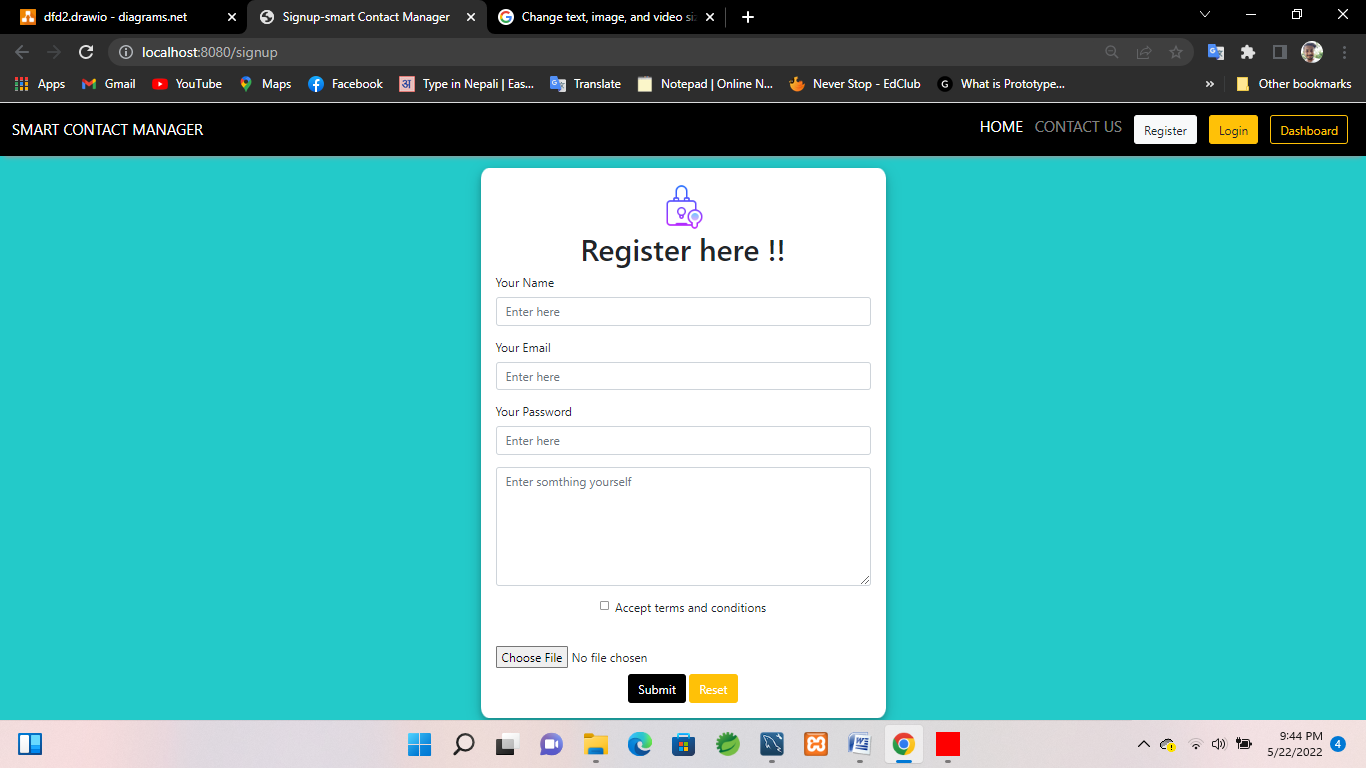
I conclude that, this web application software is the basic requirements of each and every normal person to track their contact details remotely. This software contains most basic things that help in accuracy of the result.. These types of web applications helps to digitize the whole country and play the vital role in development.

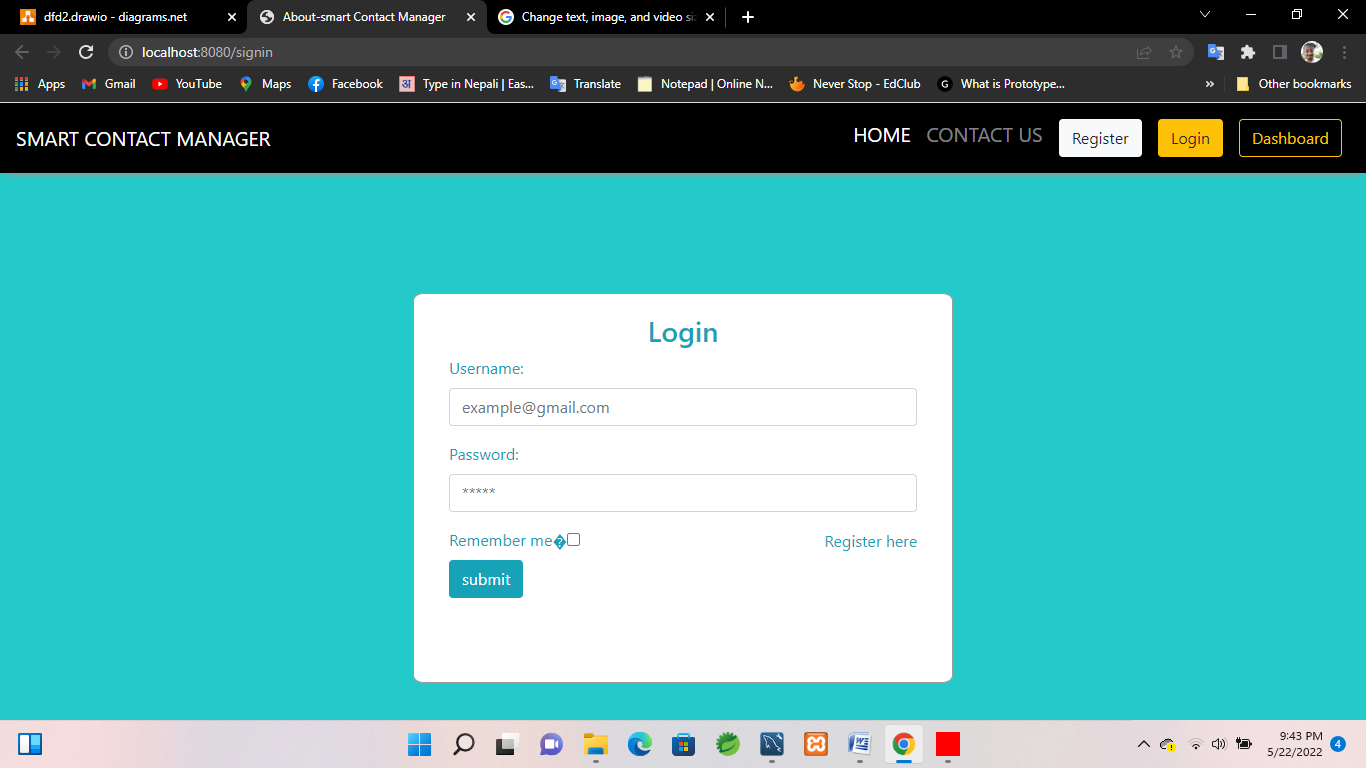
## 5.3 Future Recommendations

This is the basic application only. It can be expand further to any extend on requirement. location tracking etc.. It helps to fully digitalize the any company.

# Appendices

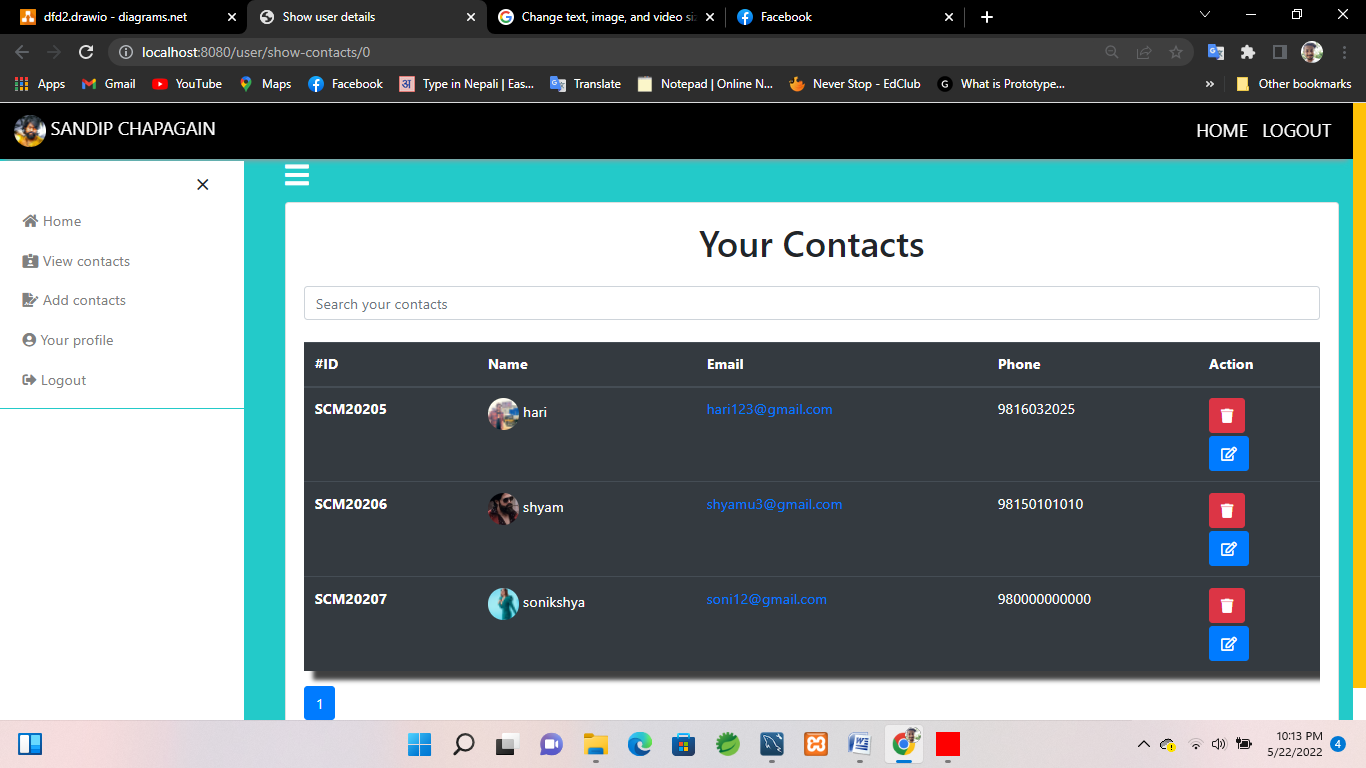
## Screenshots

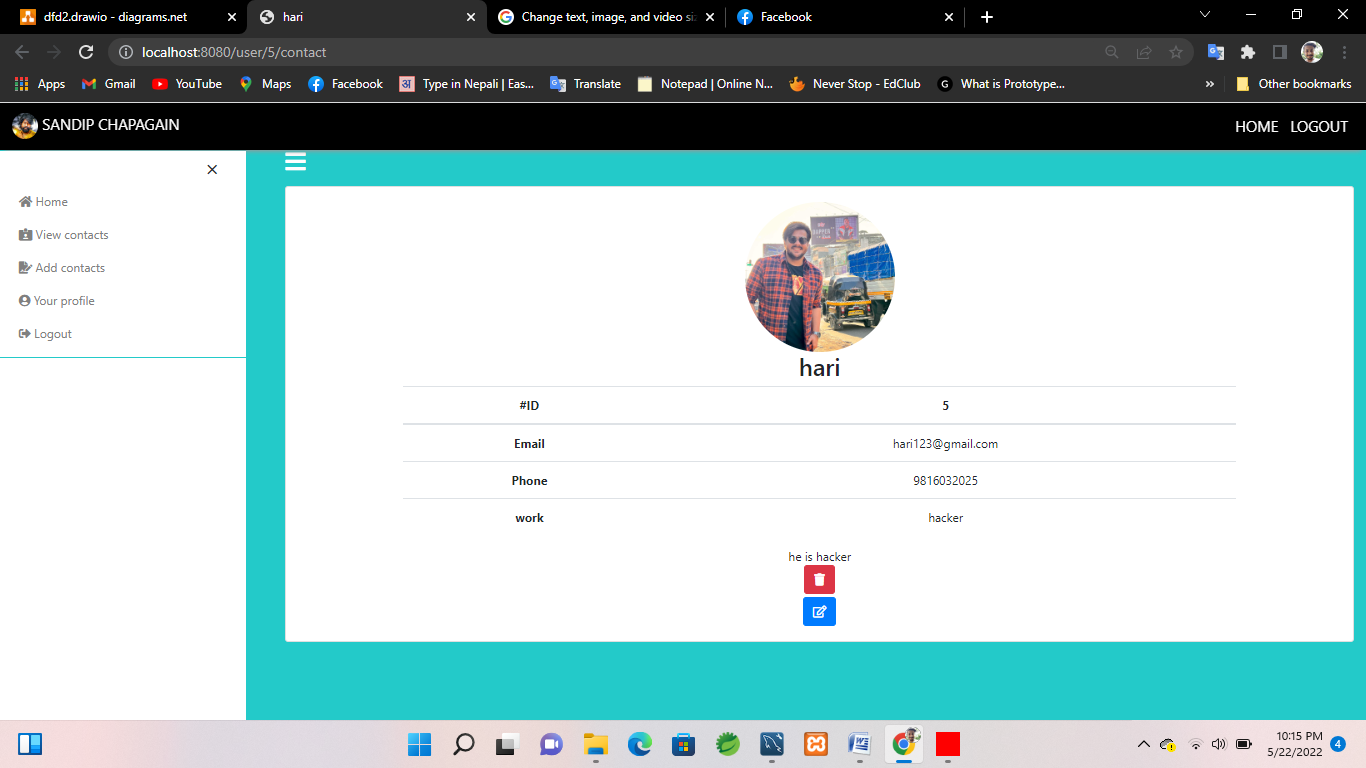




# Screenshot (67).png

# Screenshot (68).png





# Screenshot (71).png

**Source code**

package com.smart.entities;

import java.util.ArrayList;

import java.util.List;

import javax.persistence.CascadeType;

import javax.persistence.Column;

import javax.persistence.Entity;

import javax.persistence.FetchType;

import javax.persistence.GeneratedValue;

import javax.persistence.GenerationType;

import javax.persistence.Id;

import javax.persistence.OneToMany;

import javax.persistence.Table;

import javax.validation.constraints.NotBlank;

import javax.validation.constraints.Size;

@Entity

@Table(name="USER")

public class User {

@Id

@GeneratedValue(strategy = GenerationType.AUTO)

private int id;

@NotBlank(message = "Name field is required")

@Size(max = 20, min = 4, message = "min 4 and max 20 characters are allowed !")

private String name;

@Column(unique = true)

private String email;

private String password;

private String role;

private boolean enabled;

private String imageUrl;

@Column(length = 500)

private String about;

@Column(name = "verification\_code", updatable = false)

private String verificationCode;

@OneToMany(mappedBy = "user", cascade = CascadeType.ALL, orphanRemoval = true)

private List<Contact> contacts = new ArrayList<>();

public User() {

super();

// TODO Auto-generated constructor stub

}

public int getId() {

return id;

}

public void setId(int id) {

this.id = id;

}

public String getName() {

return name;

}

public void setName(String name) {

this.name = name;

}

public String getEmail() {

return email;

}

public void setEmail(String email) {

this.email = email;

}

public String getPassword() {

return password;

}

public void setPassword(String password) {

this.password = password;

}

public String getRole() {

return role;

}

public void setRole(String role) {

this.role = role;

}

public boolean isEnabled() {

return enabled;

}

public void setEnabled(boolean enabled) {

this.enabled = enabled;

}

public String getImageUrl() {

return imageUrl;

}

public void setImageUrl(String imageUrl) {

this.imageUrl = imageUrl;

}

public String getAbout() {

return about;

}

public void setAbout(String about) {

this.about = about;

}

public String getVerificationCode() {

return verificationCode;

}

public void setVerificationCode(String verificationCode) {

this.verificationCode = verificationCode;

}

public List<Contact> getContacts() {

return contacts;

}

public void setContacts(List<Contact> contacts) {

this.contacts = contacts;

}

@Override

public String toString() {

return "User [id=" + id + ", name=" + name + ", email=" + email + ", password=" + password + ", role=" + role

+ ", enabled=" + enabled + ", imageUrl=" + imageUrl + ", about=" + about + ", contacts=" + contacts

+ "]";

}

**}**

Contact.java

package com.smart.entities;

import java.util.Objects;

import javax.persistence.Column;

import javax.persistence.Entity;

import javax.persistence.GeneratedValue;

import javax.persistence.GenerationType;

import javax.persistence.Id;

import javax.persistence.ManyToOne;

import javax.persistence.Table;

import com.fasterxml.jackson.annotation.JsonIgnore;

@Entity

@Table(name = "CONTACT")

public class Contact {

@Id

@GeneratedValue(strategy = GenerationType.AUTO)

private int cId;

private String name;

private String secondName;

private String work;

private String email;

private String phone;

private String image;

@Column(length = 5000)

private String description;

@ManyToOne

@JsonIgnore

private User user;

public int getcId() {

return cId;

}

public void setcId(int cId) {

this.cId = cId;

}

public String getName() {

return name;

}

public void setName(String name) {

this.name = name;

}

public String getSecondName() {

return secondName;

}

public void setSecondName(String secondName) {

this.secondName = secondName;

}

public String getWork() {

return work;

}

public void setWork(String work) {

this.work = work;

}

public String getEmail() {

return email;

}

public void setEmail(String email) {

this.email = email;

}

public String getPhone() {

return phone;}

public void setPhone(String phone) {

this.phone = phone;

}

public String getImage() {

return image;

}

public void setImage(String image) {

this.image = image;

}

public String getDescription() {

return description;

}

public void setDescription(String description) {

this.description = description;

}

public User getUser() {

return user;

}

public void setUser(User user) {

this.user = user;

}

@Override

public int hashCode() {

return Objects.hash(cId);

}

@Override

public boolean equals(Object obj) {

return this.cId ==((Contact)obj).getcId();}}

# References

## Websites:

* [www.draw.io](http://www.draw.io)

It is used to draw different diagrams such as Data Flow diagram, ER diagram, Architecture diagram, Interface Diagram and Physical DFD diagram.

* [www.javatpoint.com](http://www.javatpoint.com) , [www.tutorialspoint.com](http://www.tutorialspoint.com) ,[www.stackoverflow.com](http://www.stackoverflow.com)
* https://www.youtube.com/LearnCodeWithDurgesh

These sites are used as alternatives for the core development of project. i.e programming techniques and ideas.

## Books:

Head First Java, 2nd Edition

by Kathy Sierra, Bert Bates

Released February 2005

Publisher(s): O'Reilly Media, Inc. ISBN: 9780596009205