**Table of Contents**

|  |  |  |
| --- | --- | --- |
| **Sr. No.** | **Topic** | **Page No.** |
| 1 | Introduction:  1.1 Project Overview  1.2 Aim  1.3 Scope  1.4 Users of System |  |
| 2 | Literature Review |  |
| 3 | Problem Definition |  |
| 4 | Requirements analysis:  4.1 Functional Requirements  4.2 Non-Functional Requirements  4.3 Hardware & Software Requirements  4.4 Technology to be Used |  |
| 5 | Proposed System |  |
| 6 | System Block Diagram |  |
| 7 | Methodology |  |
| 8 | System Design:  8.1 Use Case Diagram  8.2 ER Diagram  8.3 Sequence Diagram  8.4 Timeline Chart |  |
| 9 | Implementation  9.1 Flowchart  9.2 Screens |  |
| 10 | Limitations |  |
| 11 | Conclusion |  |
| 12 | References |  |

**CHAPTER 1:**

**INTRODUCTION**

# INTRODUCTION

## Project Overview

1. The assigned project is to display the details of the employees on an android platform from the server. The android application will display all the details of the employees.
2. The details are provided by the company and then this detail is in the form of a database that will be used to retrieve data from the server.
3. The application will contain the details of the employees from various departments, sectors, offices, former-employees.
4. Last but not least is the offline feature that will enable the user to store the information of their favourite employee in their mobile offline and can be viewed even without the internet.

## Aim

To display the information of the employees working in Tata Institute of Fundamental Research present or former and display it on an android device also giving them the offline feature to store information and display without internet.

## Objectives

1. The first foremost thing is to display the details of the employees of the company on an android device.
2. The second thing in line is to retrieve information from the database which is suitable for the android platform.
3. Thus, the main objective is to display the information into a JSON format to fetch the data from the database to display on the android platform.
4. Once the JSON array is passed the next objective is to fetch the JSON array in the android device.
5. Thus, the android code for JSON with volley library is implemented.
6. The last and the important factor is to display the details with proper UI features which cannot be unhesitant to the user.

## Scope

### Scope Statement

To develop an android application for the Tata Institute of Fundamental Research that will display the information of the employees of the company. The application is an android application that fetches the information from the server of the company. Information includes all the details of the employees of the company which includes former, retire as well as the present with their current status. With the simple and user-friendly User Interface, the application is made for the motive to show the information of all the people involved in the institution. On searching the name, the details are also fetched and then on clicking on the name the new activity opens which shows all the display of the desired employee which the user wants to see. The other important feature of the application is, the details of the employees searched can also be kept offline by pressing on the favourite button This will be stored separately and can be seen offline which is the favourite bar.

### Scope Boundary

#### Within the Scope

1. Searching the employee details.
2. All the details can be fetched of the desired employee.
3. The details can be stored offline in a favourite bar.
4. Secured way of details transferred from the server to the application.
5. User-friendly interface.
6. Available on all the devices.

#### Outside the Scope

1. Cannot insert data from the application.
2. Authentication of the user.
3. Update the details from the application.

## Users of the System

1. TIFR Employees.
2. Visitors to TIFR.
3. Security team.
4. H.O.D.s and Admins of different departments.

**CHAPTER 2:**

**LITERATURE REVIEW**

# LITERATURE REVIEW

## Successful strategies of older people for finding information

**Author:** Paul Curzon, Judy Wilson, Gill Whitney

**Abstract:**

Older people have successful search strategies for finding practical information in everyday situations but, increasingly, traditional information sources are being supplemented or replaced by web-based ones. However, there are wider issues than just making information available if people are to replace existing strategies by new web-based ones. In this paper we use three studies on the information usage of older people to explore the issues surrounding why they favour specific search strategy and technology combinations. The studies each investigate different aspects of information search in a natural setting and concern tasks relevant to older people as their lives change: finding e-government information and planning travel. Results suggest that a variety of measures are important in the choice of strategy. Furthermore, interface mechanisms are needed that complement existing strategies, reinforce the existence and crossing of boundaries, and support interactive use of landmarks.

**Inference:**

1. In this research paper the author focuses on the searching techniques used by various companies and groups of people in society.
2. The author tries to find various strategies for people finding a problem and uses different approaches to get the optimal solution.
3. As in the example given in the 3rd point of the paper it says that the authors try to enhance the search option for housing societies and they try to find people living in it.
4. Thus, it pictures a scenario about the searching techniques of the author while using the technologies.
5. The author uses web technologies for searching for this particular situation.
6. The other scenario is where the authors check the finding techniques of the other people and how they try to find nearby places and people.
7. Thus, we can say that people use different strategies for finding the route and people which may include directories, web searches, etc.
8. Last but not least the author mentions the searching options on the web-technologies.
9. The web-searches which are used for searching the places and the people also about the approaches they apply for finding various places.

**Lacuna:**

1. The system as discussed by the author has many limitations concerning the finding options and their approach towards the searching techniques.
2. The first thing in this approach for the people finder system is the collection of the data for searching. The collection of the data is not that authenticated and properly searched as the data may include many redundant data.
3. As the authors mention the approach of getting the data from directories and web-searches may lead to redundant data. In our system, the data is authenticated and given by the company itself which will not have any intertwining in it.
4. The next thing is the approach of the author to find. As the author uses web-searches for finding the solution, our approach is to create an android application that displays the results and also provides the offline feature.
5. Lastly, the author gives their approach which has many limitations compared to the People Finder System of TIFR.

## Comparative Study of Syntactic Search Engine and Semantic Search Engine: A Survey

**Author:** A.C. Santha Sheela, Dr.C.Jayakumar

**Abstract:**

The search engine symbolizes an extremely powerful and valuable tool for fetching any sort of information from the Internet. There have been numerous researches carried on search engine techniques, the major ones are syntactic and

semantic. Referring to the Syntactic web, the results obtained are pure as per the keyword match. That is the query outputs numerous web pages against the keyword match that may not even be relevant or meaningful. Whereas, unlike the syntactic web, the semantic web is a revised or upgraded version of the

web which produces quite meaningful and specific output as it has the potential to comprehend the query effectively. Few examples of Semantic-based search engines include Kosmix, Hakia, Cognition, Swoogle and Lexxe. Whereas syntactic based search engines are Google, Yahoo, Ask. The work performs a comparison amidst the performance of semantic and syntactic based search engine and evaluates them by employing certain queries.

**Inference:**

1. The authors propose a system that reveals the similarity in the sentence based on the verification of repeated words via semantic analysis.
2. The proposed system carries out such operations to search similarity in various sentences within the same document.
3. The author differentiates between semantic search engines and syntactic based search engines.
4. As with the rapid growth and popularity of WWW, many search engines have also spread its roots like Google, Yahoo, Hakia, Ask, Kosmix, Lexxe, cognition, Swoogle and so on. Amidst all prevailing search engines today, there are a handful of ones that offer relevant and specific information as desired by the user.
5. The research work takes into account 6 different sorts of search engines namely Google, Yahoo!, Wikipedia for syntactic based and Hakia, Bing and DuckDuckGo for semantic-based to determine the performance of these search engines and the one resulting in the most desired information.

**Lacuna:**

1. From the given research paper which focuses on the searching techniques and the semantic and syntactic search engines.
2. The searching techniques in the various search engines differ, in our system there is no need for any web-search engines as the data is uploaded into the database.
3. The data is fetched with the help of the JSON format and displayed in an android application.
4. Hence, there is no need for any search engines in this system.
5. Secondly, the main difference in this is that our system includes the offline feature which stores the data offline into the mobile database which other system lacks.
6. Lastly, all the differences are shown between the authors' system and the People Finder System of TIFR.

The above research papers are the papers relevant to our project as the project is given by the TIFR so there are not many related research papers on this topic apart from these papers.

|  |  |  |
| --- | --- | --- |
| People Finder Application: TIFR | People Finder Application: Microsoft Corporation. | People Finder Application: Other Companies. |
| In our proposed system is built for looking into the employee details of TIFR. | In the Microsoft People Finder Application is built to connect and collaborate with the employees. | In other people finder applications like Easy Connect, or LTI people finder application the purpose is to display employee details. |
| The system is simple and user friendly and for the employees and the users of TIFR company. | This system is available for all Employees as well as for the public. | This depends on application to application mostly it is for the employees and the HODs. |
| In this system, only details are provided and there are no other operations performed. | In this system of Microsoft, they provide the facility to the users to directly contact the employee but using skype. | This system only provides the details and no other operations are performed. |
| There is a facility to store the thing offline for future use without the internet. | There is no offline feature for using the system offline without internet. | There is no feature for storing the details offline for using it without the internet. |
| This system offers flexibility in searching the employee details by sorting the employees in various sections like department, ex-employee, branch. | The system offers segregation with the help of the department as there are various sectors in the corporation. | It does not have any segregation facility to the users. |
| The offline section of the system is another system that stores the data into the mobile database and can be easily deleted after the data is utilized. | There is no system for deleting the offline system. It simply gets stored into the system memory and will be deleted when the users delete the application. | There is no such feature. |

*Figure 2.1 – Represents the comparison between People Finder Application and various systems*

The above table represents the differentiation between the system of TIFR compared with the Microsoft People Finder System and other systems. The comparison shows the difference in the systems and the different approaches as well as the different functionalities used in the system. Also focusing on the different features that each system has, whether that feature is present in the other systems or not.

**CHAPTER 3:**

**PROBLEM DEFINITION**

# PROBLEM DEFINITION

People finder System deals with users that want to find details about the employees working at TIFR (Tata Institute of Fundamental Research). People finder is the optimization and up-gradation of the existing system.

The problem with the existing system was that it didn’t work in major android devices/versions and was unable to show the data of the employees on various filters and list them out.

A system was to be developed which was snappy and also can store employee’s data offline so that it can be accessed if the internet is not available. Most importantly handy and easy to use with good user experience and interface.

**CHAPTER 4:**

**REQUIREMENT ANALYSIS**

# REQUIREMENT ANALYSIS

## FUNCTIONAL REQUIREMENTS

* To search and display employee details that are currently working or worked at TIFR.
* Details of the individual employee can be stored offline.
* The employee can be searched using listed filters
* Name.
* In-Service.
* Ex-Members.
* Department.
* Mumbai Campus.
* TIFR Centres & Field Station.
* Room Number.
* Email ID.
* Extension.
* A favorite tab that shows all offline employee’s data.
* Email, websites, phone numbers are clickable.
* A button to add and remove employees from favorite (offline).

## NON-FUNCTIONAL REQUIREMENTS

### AVAILABILITY

As the user of the application can need it anytime so the system is made available 24\*7. System availability is the amount of time it is operational and available to use. Therefore, the system should be available 24\*7.

### MAINTAINABILITY

The application is developed in Java in the android framework and the business logic is in PHP and data in MySQL and easy to maintain.

### PORTABILITY

The application can be installed on any android device and no setup is required it is easy to use. And hence for it can we worked anywhere anytime.

### PERFORMANCE

* The total load time of the data will be less than 2 seconds but can increase or decrease upon internet connectivity.
* No setup required, easy to go.
* Offline/Favourite employees shall remain until the application data is cleared manually.

## HARDWARE AND SOFTWARE REQUIREMENTS

### HARDWARE REQUIREMENTS

* Device storage of 3-8 MB for installation and offline information storage.
* Android Device with internet connectivity (3G/4G LTE/Wi-Fi).

### SOFTWARE REQUIREMENTS

* Web servers supporting PHP.
* Playstore to install the application.
* MySQL Database.

## TECHNOLOGY TO BE USED

* For Application:
* Language: Java.
* Framework: Android.
* Database: SQLite.
* For data:
* Database: MySQL.
* Language: PHP.

**CHAPTER 5:**

**PROPOSED SYSTEM**

# PROPOSED SYSTEM

Our proposed system aims to bring TIFR employee details to the users who want to view it right in their hands. These details can be employee's email ids, contact numbers or even office address.

All the employee data is stored in the main servers which are accessed by our system and the required employee can be filtered out. This filtering makes it easy for the user to look for the employee without wasting a lot of time. The system also provides a modern UI that is pleasant to look at with additional features like direct email access or contact from within the app.

The app is built with a minimum SDK version set as 19. This allows for android devices running android version 4.4 i.e. Android KitKat to be the minimum supported version. The application is also future proof as there are no particular SDK specific functions used. The app also has an offline saving feature which is supported by the GLIDE library used to fetch images from the database.

**Features of the System**

1. **Filter Search:**

The application allows users to search employees based on filters like their name, department, email id or room no. This helps to find the specific employee easy and quick. It also eliminates the process of going through several unwanted employee profiles.

1. **Modern UI/UX:**

The whole application is developed keeping Google’s latest material design guidelines in mind. Button placements are done for easy access. Also, for enhanced user experience email ids, contact numbers and websites of individual employees are clickable allowing for a seamless transition between different apps for quick functioning. Clicking on email id opens the stock phone email app with the selected email id already entered in the composition activity making it quick to mail the employee without jumping again and again between the apps. Similarly, clicking the contact number opens the phone app and the website opens the browser app.

1. **Offline Saving:**

The employee details viewed on the app can be saved offline for later viewing when not in the network connection. This also allows for saving frequently viewed employee profiles for easy access. The textual data is saved in the phone’s internal database storage. The profile images are saved in the form of caches. This allows for better storage management. When running on low storage, images can be easily removed by simply cleaning the app cache with all the textual data still being present on the app's offline page.

1. **Easy Security Implementation:**

The entire application's data fetching is done on the server-side. This means, to improve overall security, the security mechanism of the main server only needs to be updated with few updates in the data fetching file in the server.

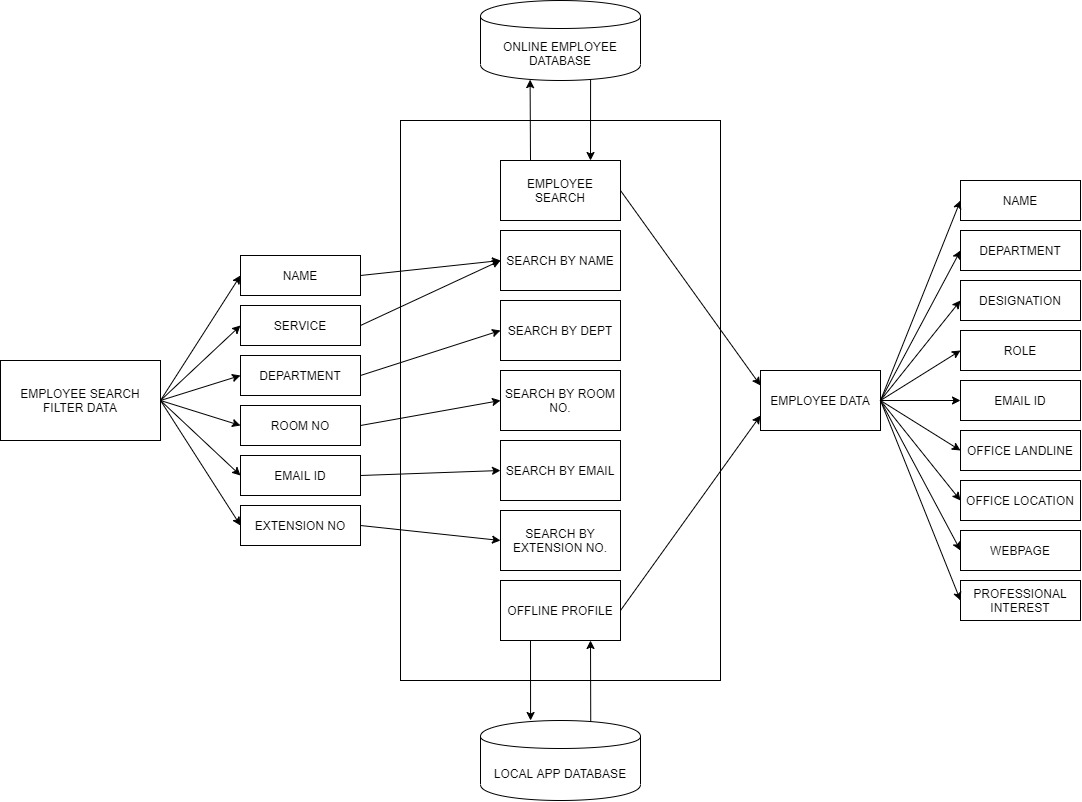
1. **Quick All-Access Shutdown:**

In case of application being accessed by unauthorized users, on detection, the server-side file can be blocked to instantly shut down the app's capability to retrieve data from the server. This allows for a quick and easy countermeasure against data leakage.

**CHAPTER 6:**

**SYSTEM BLOCK DIAGRAM**

# BLOCK DIAGRAM



*Figure 6.1 – Block Diagram for People Finder Application*

The block diagrams represent the functionality where the name of the employee is taken as the input and in return the details of the employees are fetched from the server. In addition to this search it can be more flexible by filtering out the names concerning designation, branch, department.

**CHAPTER 7:**

**METHODOLOGY**

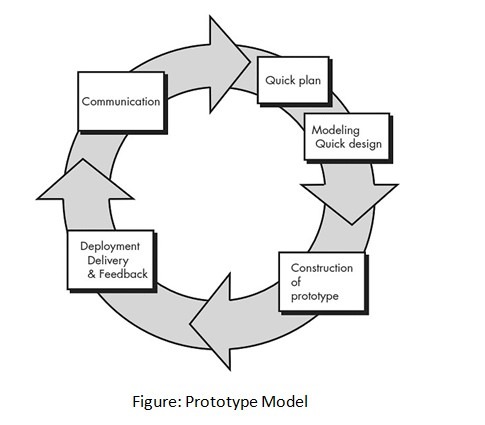
# METHODOLOGY (PROTOTYPING MODEL)

A prototype is a working model of software with some limited functionality. The prototype does not always hold the exact logic used in the actual software application but can be used to demonstrate concepts, design options and find out more about the problem and its possible solutions.

Prototyping is used to allow the users to evaluate developer proposals and try them out before implementation. It also helps understand the requirements which are user-specific and may not have been considered by the developer during product design.

The prototyping paradigm begins with communication where you meet with other stakeholders to define the overall objectives for the software. A prototyping iteration is planned quickly, and modelling (in the form of a “quick design”) occurs. A quick design focuses on a representation of those aspects of the software that will be visible to end-users (e.g. Human interface layout of output display formats). The quick design leads to the construction of a prototype.

The prototype is deployed and evaluated by stakeholders, who provide feedback that is used to further refine requirements.



*Figure 7.1 - Prototyping Model*

Iteration occurs as the prototype is tuned to satisfy the needs of various stakeholders, while at the same time enabling you to better understand what needs to be done.

The prototype can serve as “the first system.” Although some prototypes are built as “throwaways,” others are evolutionary in the sense that the prototype slowly evolves into the actual system.

Both stakeholders and software engineers like the prototyping paradigm. Users get a feel for the actual system, and developers get to build something immediately.

**When to use the Prototyping Model?**

1. When the requirements are not completely defined initially.
2. When the actual project is very large.
3. When the risk involved is medium to high.
4. When a working model is demanded by the client before the actual, complete version of the system.
5. When significant changes are expected in the course of the project.
6. For long-term projects.
7. The prototyping model should be used when the desired system needs to have a lot of interaction with the end-users.
8. Online systems, web interfaces have a very high amount of interaction with end-users, are best suited for the Prototype model. It might take a while for a system to be built that allows ease of use and needs minimal training for the end-user.
9. Prototyping ensures that the end-users constantly work with the system and provide feedback which is incorporated in the prototype to result in a useable system. They are excellent for designing good human-computer interface systems.

**Advantages of the Prototyping Model**:

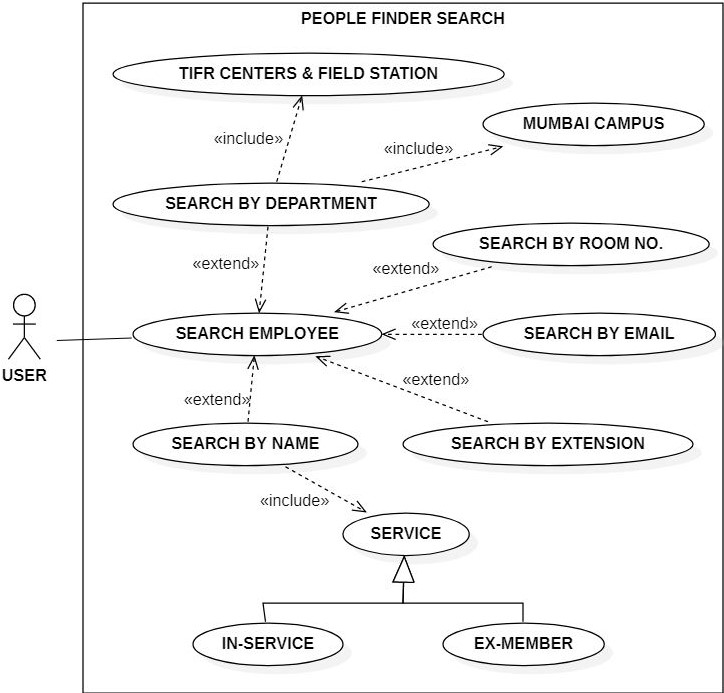
1. Stakeholders are actively involved in development.
2. Since in this methodology a working model of the system is provided, the users get a better understanding of the system being developed.
3. Missing functionalities can be identified easily.
4. Confusing or difficult functions can be identified.
5. Improved system usability.
6. A closer match to users’ real needs.
7. Improved design quality.
8. Improved maintainability.
9. Reduced development effort.

**CHAPTER 8:**

**SYSTEM DIAGRAM**

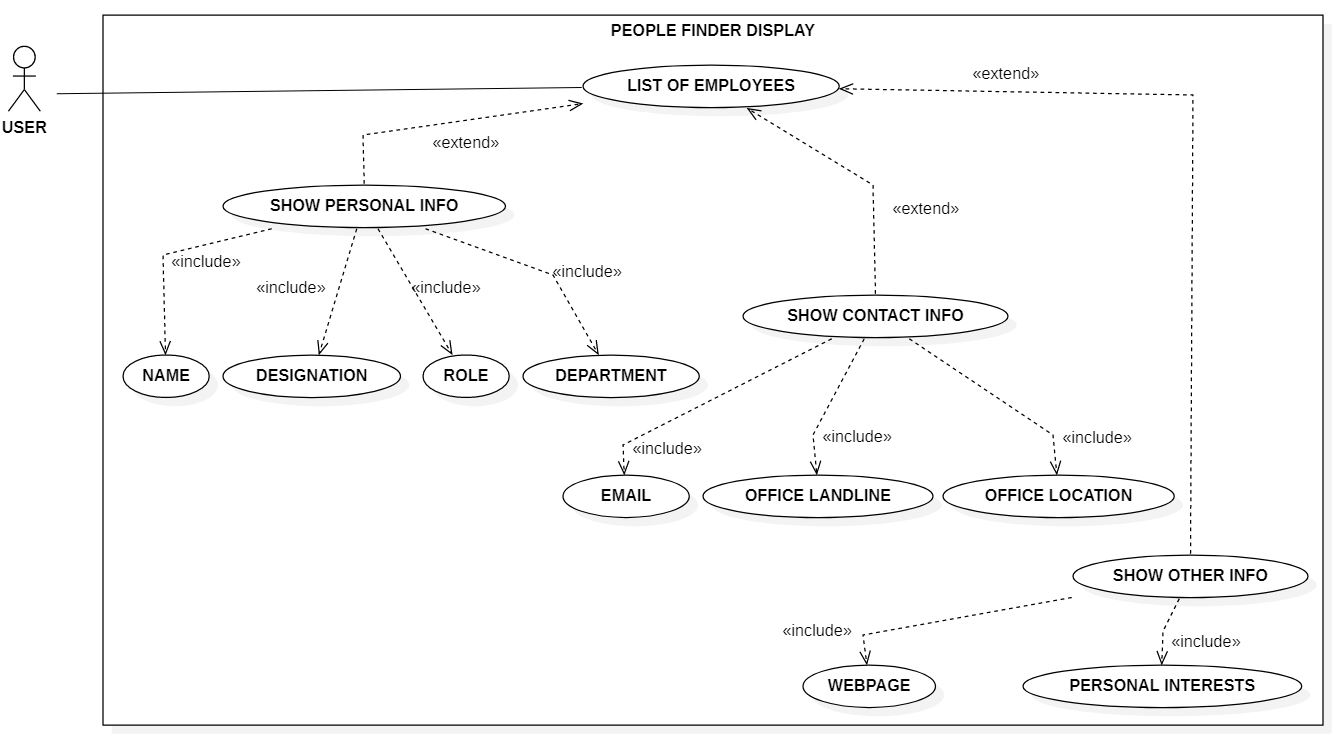
# SYSTEM DESIGN

## Use Case Diagram



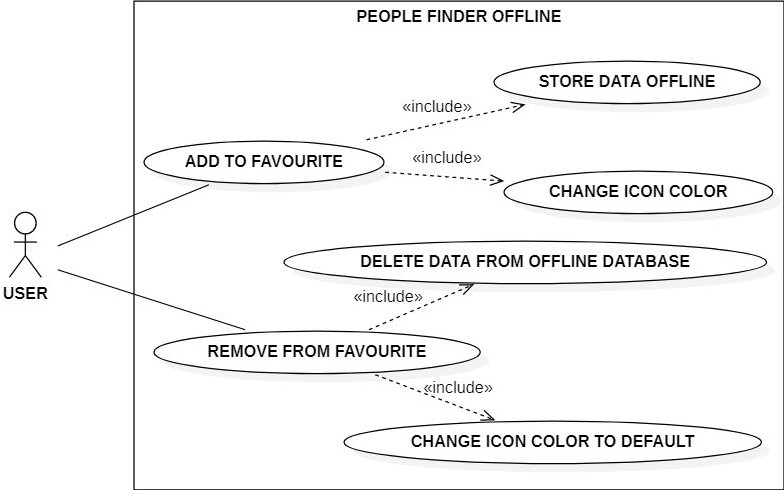
*Figure 8.1.1 – People Finder Search Use case*

The above diagram shows the use case of the system related to the search functionality where the user searches the employee details using the filter such as department, branch and many other attributes. It also provides the feature to search concerning the In-Service and Ex-Service of the employee.

**

*Figure 8.1.2 – People Finder Display Use case*

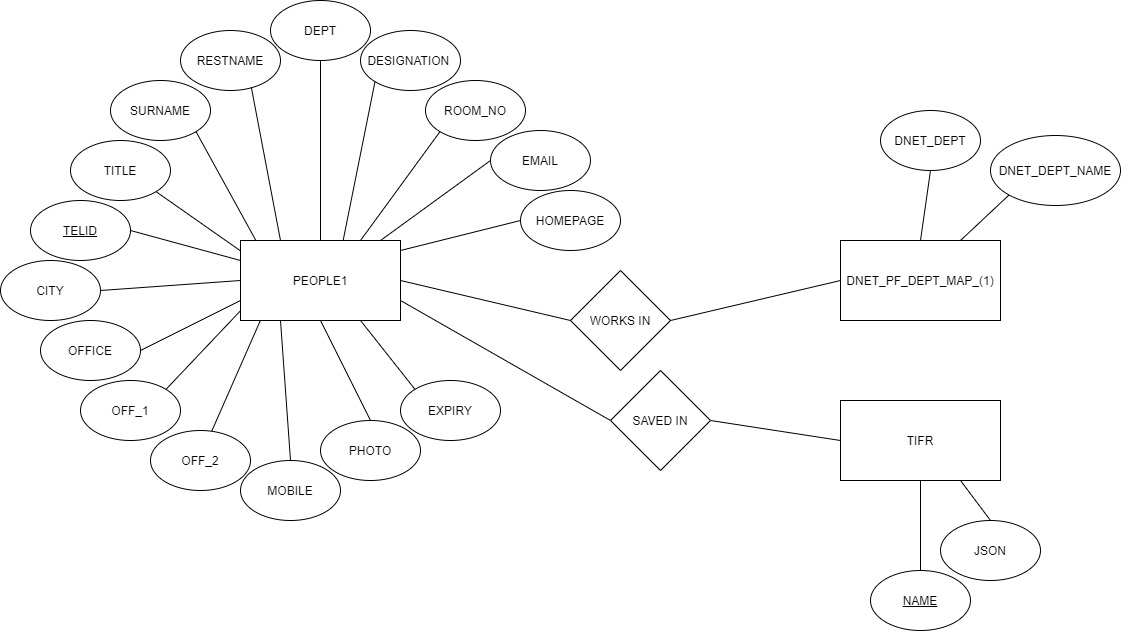
The above use case represents the display function where the searched employee details are fetched as per the query in the database. Thus, these details will be fetched and displayed. The details that will be displayed will contain the Personal Information as well as the Contact Info.

**

*Figure 8.1.3 – People Finder Offline Use case*

The given figure displays the offline feature of the application where the employee details can be stored offline in the mobile database. The employee details that are being searched can be stored in the favourite bar which is accessible even without the internet.

## ER Diagram



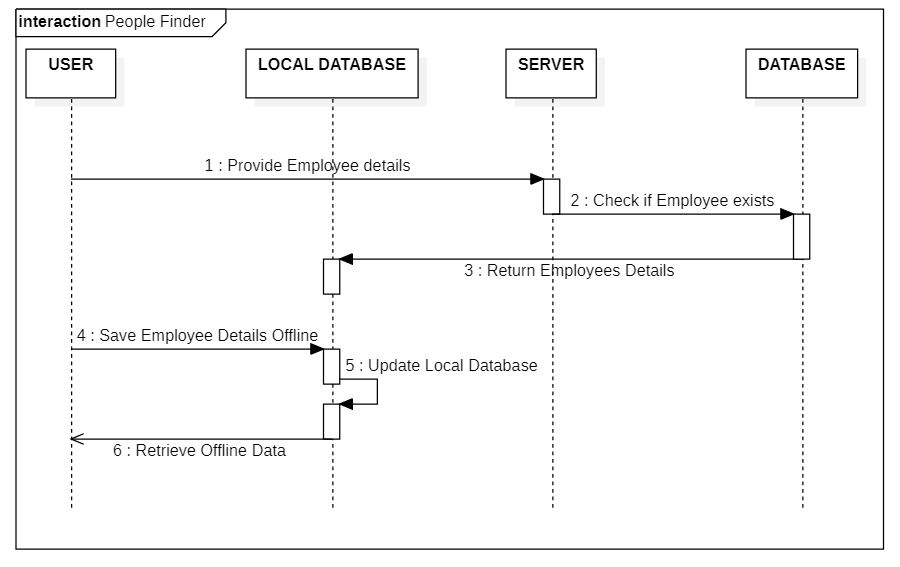
*Figure 8.2.1 – ER Diagram for People Finder Application*

The above Entity-Relationship (ER) diagram shows the relationship of entity sets stored in the database. An entity can be an object, component of data, etc. Here, the objects are the different kinds of visitors. These entities can have attributes (ex - name, phone number, photo, gender, etc) that define their properties. By defining the entities, their attributes, and showing the relationships between them, an ER diagram illustrates the logical structure of databases.

The main attributes People1 displays all the entities related to the search elements of the employees which will be used as a query in the database.

The other two display the attributes concerning the android server connectivity which will fetch the data from the server in the android application.

## Sequence Diagram

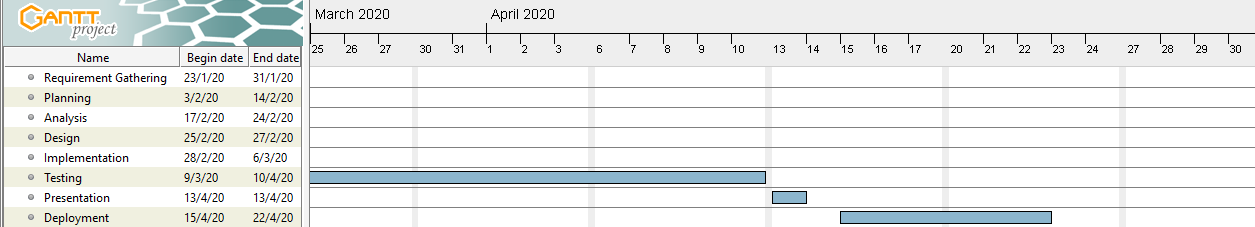


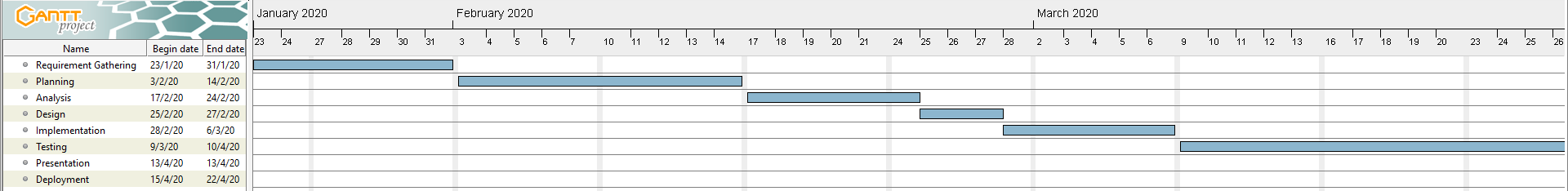
*Figure 8.3.1 – Sequence Diagram for People Finder Application*

The above diagram shows the sequence diagram. A sequence diagram simply depicts the interaction between objects in sequential order i.e. the order in which these interactions take place. We can also use the terms event diagrams or event scenarios to refer to a sequence diagram.

As shown in the figure the diagrams depict the functionality of the system sequentially which means step by step operations of the system which starts with the user entering the name to be searched to the end step by displaying the details of the employees.

## Timeline Chart





*Figure 8.4.1 - Timeline Chart for People Finder Application*

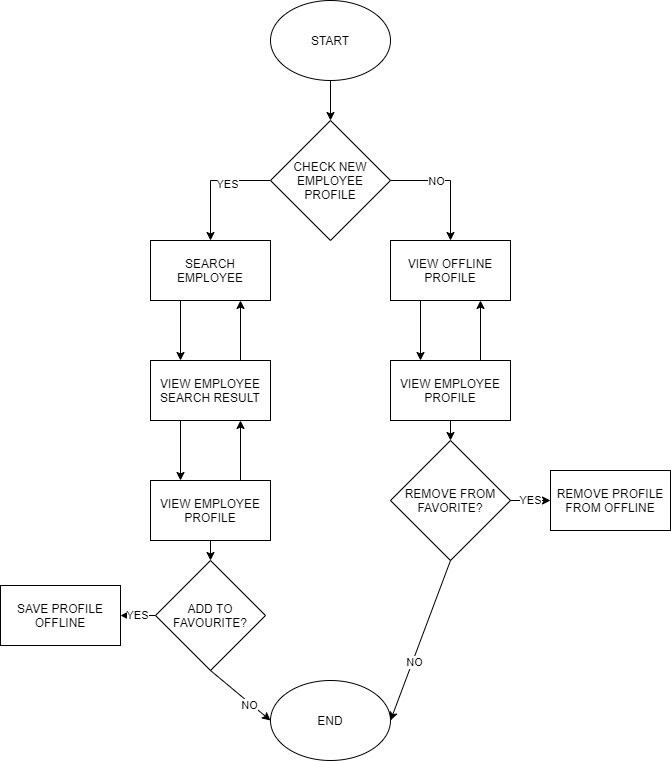
The above figure depicts the Timeline Chart of the entire project. It involves phases like Requirement Gathering, Planning, Analysis, Design, Implementation, Testing, Presentation and Deployment. These phases is schedules from January 2020 to April 2020 respectively.

**CHAPTER 9:**

**IMPLEMENTATION**

# IMPLEMENTATION

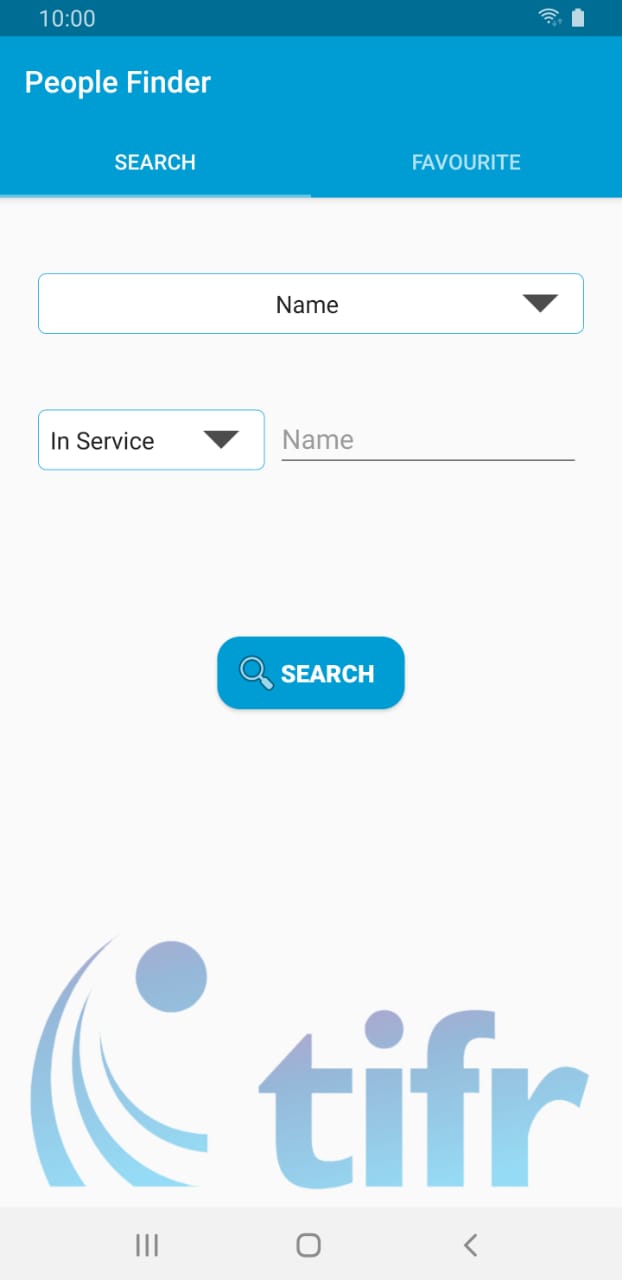
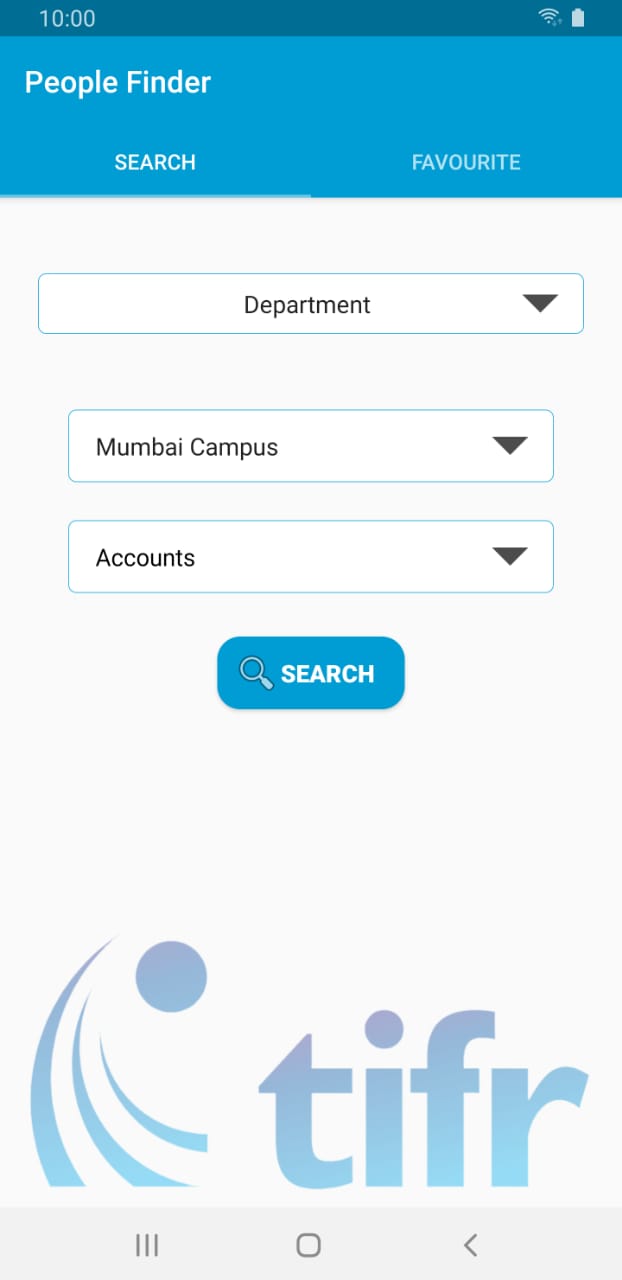
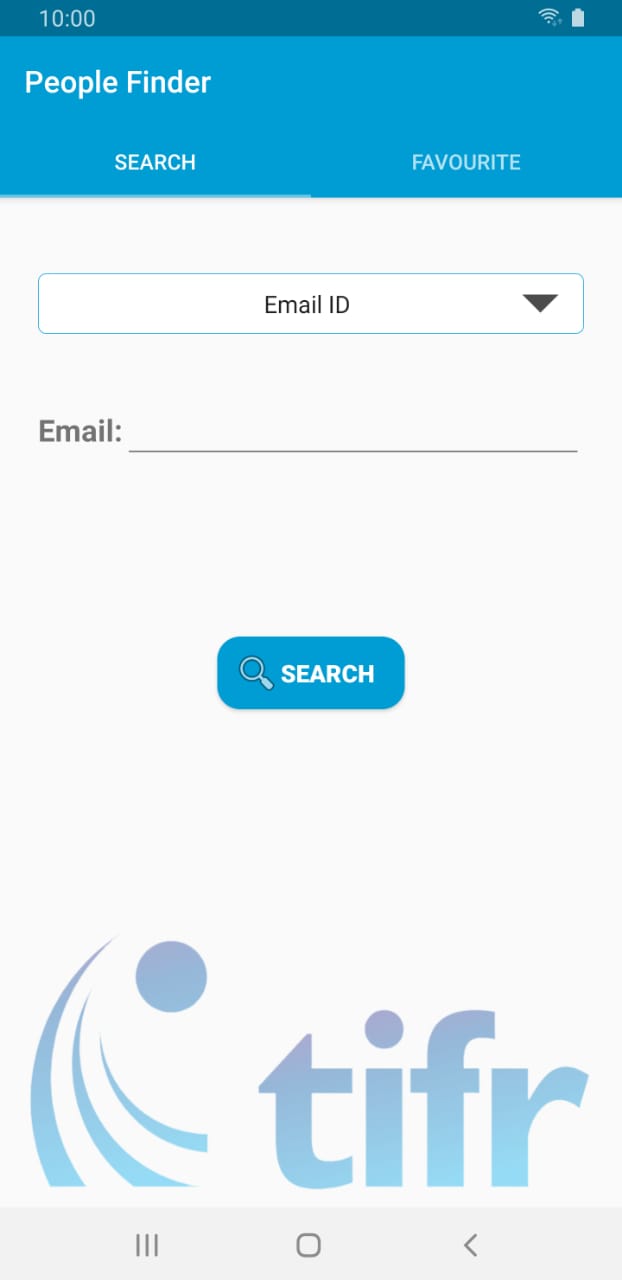
## Flowchart



*Figure 9.1.1 – Flowchart for People Finder Application*

The above diagram signifies the whole functionality of the system. Considering all the cases that will take place in the system will be shown and then showing the results of the system.

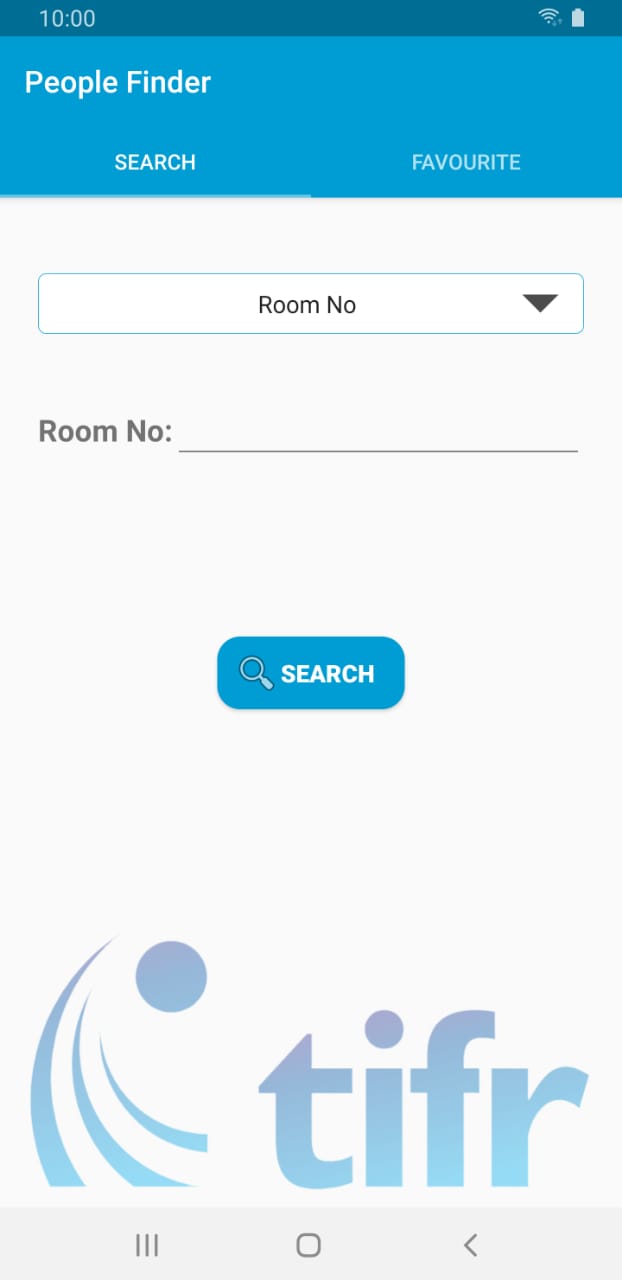
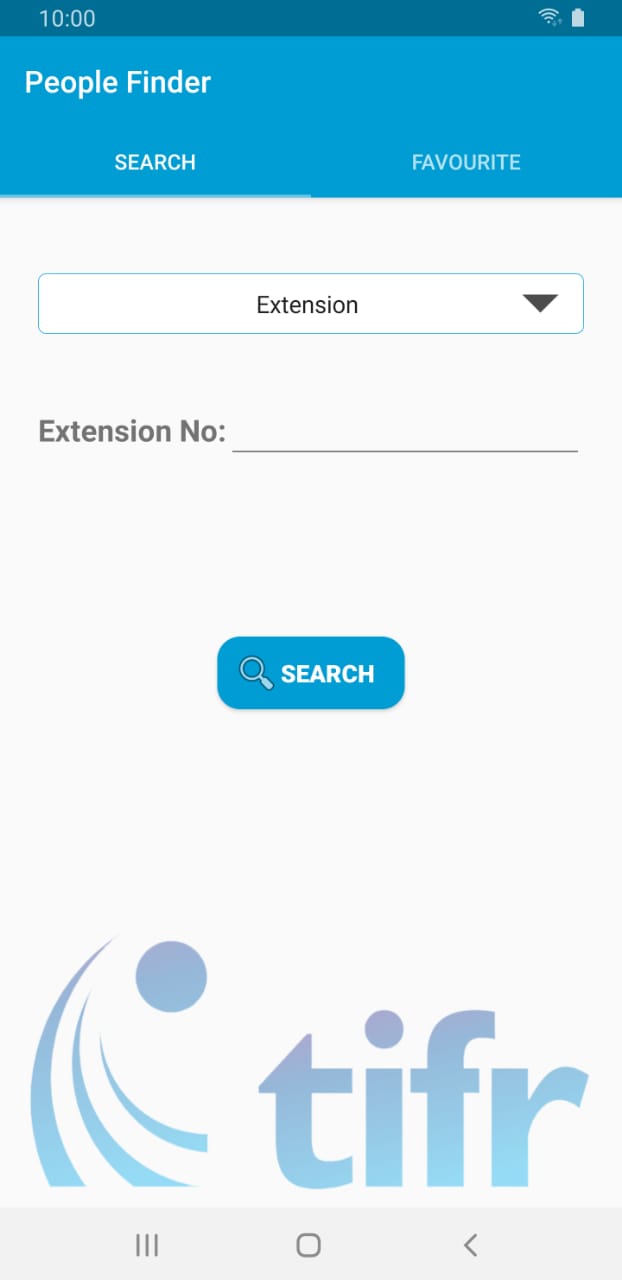
## Screens

*Fig 9.2.2- Search by Department*

*Fig 9.2.3- Search by Email ID*

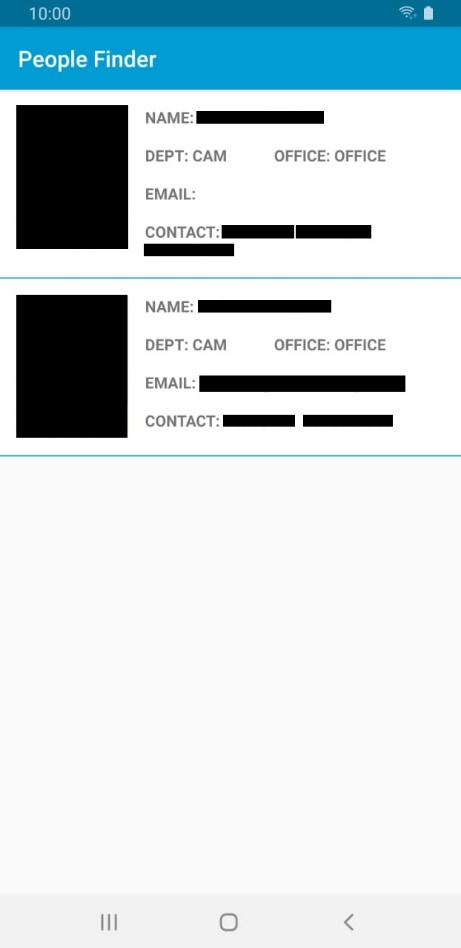
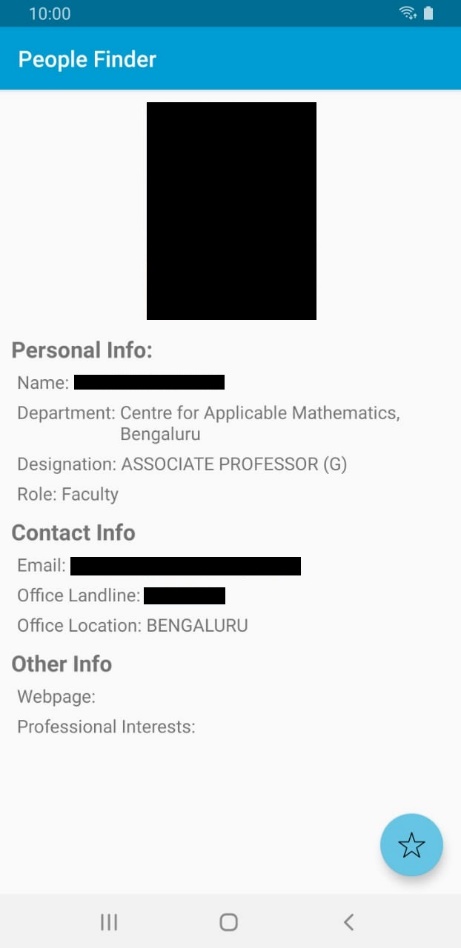
*Fig 9.2.1- Search by Name*

*Fig 9.2.5- Search by Extension*

*Fig 9.2.4- Search by Room No*

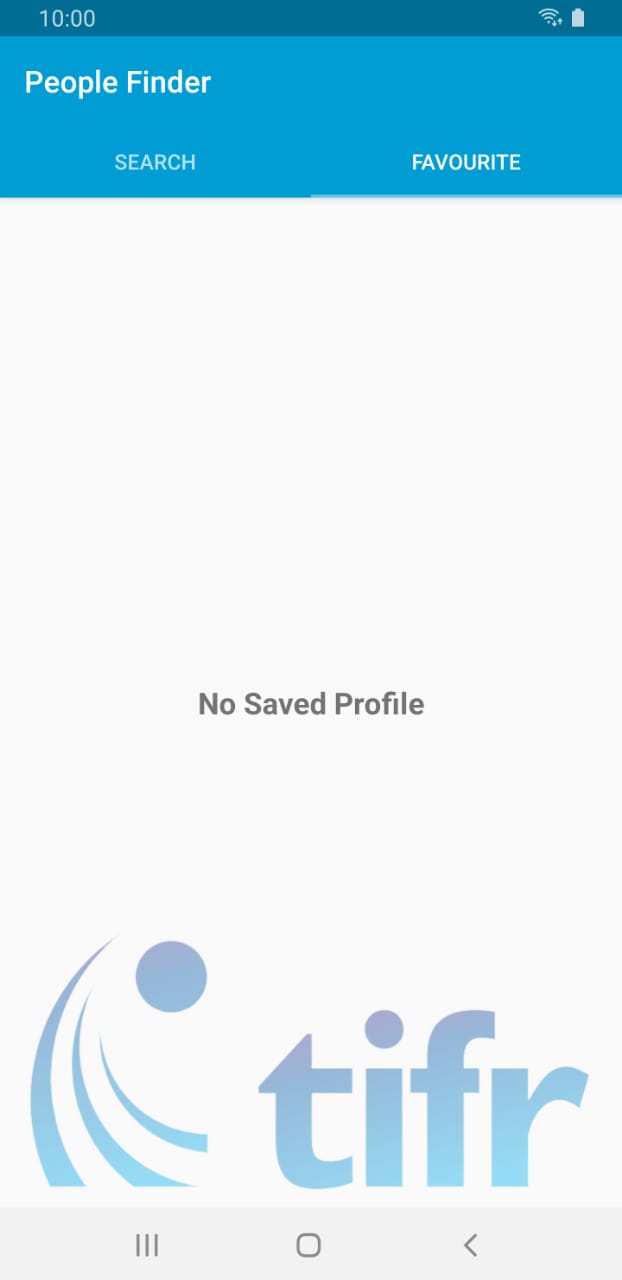
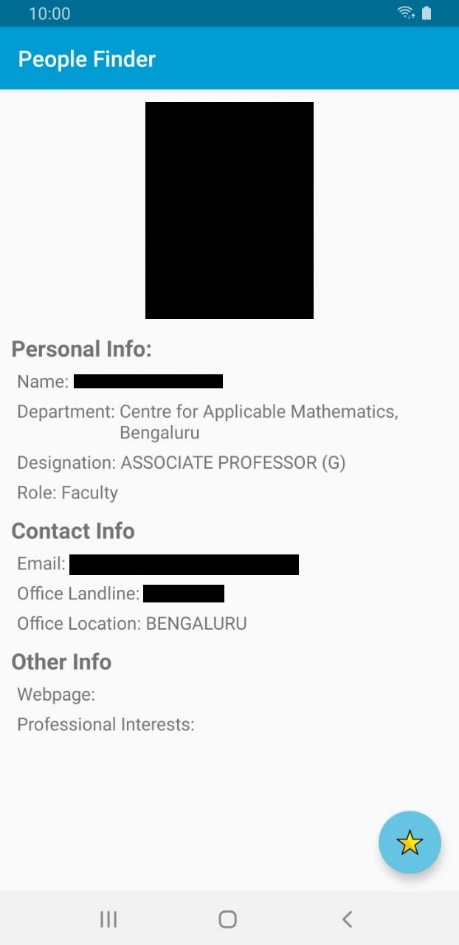
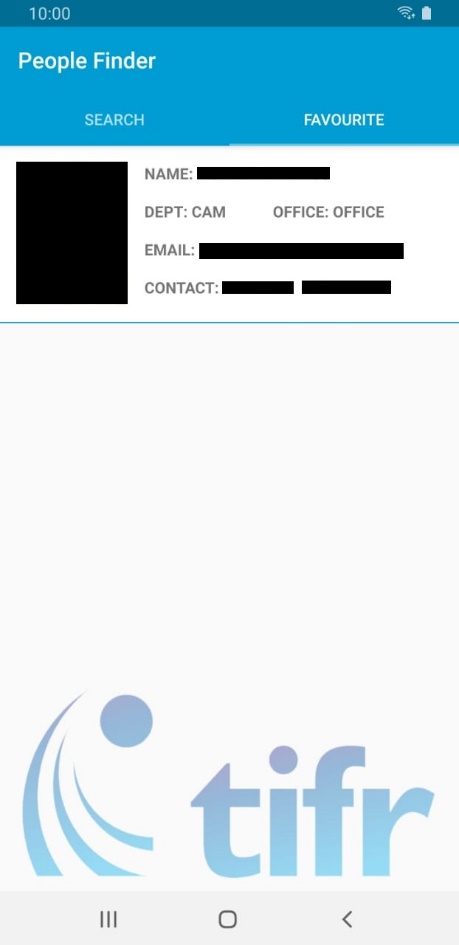
The above set of images are the function where user can search the employee concerning various filters like name, department, email, room no and extension no. Thus by searching with the help of such a field we can fetch the details of the desired employee.

*Fig 9.2.7- Employee Details*

*Fig 9.2.6- Search Result*

The above image is an example of a search result. When a user searches the employee, the results are shown in such a format containing all the details in it as well as the image of the employee.

*Fig 9.2.8- Favourite Bar without any data*

*Fig 9.2.10- Favourite Bar containing data*

*Fig 9.2.9- The details displayed in offline feature*

The above set of images are representing the favourite bar in our application. The offline feature is one of our important feature where the user can store the data of the desired employee into the mobile database for offline usage.

**CHAPTER 10:**

**LIMITATIONS**

# LIMITATIONS

The application is a simple interface between the online employee database and the user. It is recreating of TIFR's people finder web application into an android mobile application. There are a few things that aren't implemented in the application.

1. The system won’t have a switchable dark theme mode. This is due to the applications minimum SDK version 19 that does not support it.
2. User authentication is not implemented in the system. No biometric authentication method can be used on SDK version 19.
3. There is no mechanism to identify a user using the system.
4. System security is completely dependent on the main database servers. There is no dedicated security mechanism in the application.

**CHAPTER 11:**

**CONCLUSION**

# CONCLUSION

In our system we have used the approach of the android server connectivity. The system fetches the data from the server and displays it on any android device. Making it look interactive and user friendly. The application consists of the offline feature to store the data and use it without the internet.

The system integrates the huge data which is stored on the server and fetch as per the requirement. The major flexibility shown by the application is the sorting of the data as per the department, current employee, ex-employee, branch. This gives the user to search the employee with ease and consumes less time. Thus, we can conclude by saying that this application is available and works on the majority of the android devices, built without any dependencies and also is user friendly and interactive consuming very less time of the user.

**CHAPTER 12:**

**REFERENCES**

# REFERENCES

1. <https://appsource.microsoft.com/en-us/product/office/WA104380550?src=Office&tab=Overview>