

A Project/Dissertation ETE Report

on

A DATA-DRIVEN APP FOR HELPMEEET ASSISTANCE

*Submitted in partial fulfillment of the
requirement for the award of the degree of*

B.tech in Computer Science and Engineering



Under The Supervision of

Dr. Anuj Kumar Singh

Submitted By

Kunal Kaushik 20SCSE1010865

Nilesh Shekher 19SCSE1010277

**SCHOOL OF COMPUTING SCIENCE AND ENGINEERING
DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
GALGOTIAS UNIVERSITY, GREATER NOIDA
INDIA, May 2023**

CANDIDATE'S DECLARATION

I/We hereby certify that the work which is being presented in the thesis/project/dissertation, entitled **“A Data-Driven App for Helpmeet Assistance”** in partial fulfillment of the requirements for the award of the B.Tech submitted in the School of Computing Science and Engineering of Galgotias University, Greater Noida, is an original work carried out during the period of February 2023 to July 2023, under the supervision of Dr. Anuj Kumar Singh, Department of Computer Science and Engineering/Computer Application and Information and Science, Galgotias University, Greater Noida

The matter presented in the thesis/project/dissertation has not been submitted by me/us for the award of any other degree of this or any other places.

Kunal Kaushik 20SCSE1010865

Nilesh Shekher 19SCSE1010277

This is to certify that the above statement made by the candidates is correct to the best of my knowledge.

Dr Anuj Kumar Singh

Asst.Professor

CERTIFICATE

The Final Thesis/Project/ Dissertation Viva-Voce examination of Nilesh Shekher and Kunal Kaushik has been held on_____and his/her work is recommended for the award of B.Tech

Signature of Examiner(s)

Signature of Supervisor(s)

Signature of Program Chair

Signature of Dean

Date:

Place: Greater Noida

Abstract

In this competitive era, the education among the people is so increasing that the jobs for them are now decreasing. The companies even want the people who are the best in their fields. At that time, it becomes difficult to find people who are intelligent enough to be hired. A data-driven app using AI bot is a software program that integrates artificial intelligence (AI) and data analytics to provide personalized recommendations, assistance, and insights to users. In this paper, we explore the design and implementation of a data-driven app using AI bot, including the benefits and challenges of such an app. We also examine the potential impact of a data-driven app using AI bot on various industries and society as a whole. The person will have the account after registration and be called the applied user. If he were qualified, he would be interacting with the system for updates. The project is created to fulfill the requests of the company managers so that the recruitment module can be placed on the company's website and the users who visit the website can view the vacancies in the company and will be able to apply directly from a remote place even. The vacancies will be posted by the administrator on the basis of the needs of the manpower in the company. Dealing with household services like plumbing, pest control, carpentry, electricity, etc. is a major problem in urban areas where people are busy with their daily activities. It is also difficult because of the non-availability of service providers around a certain region/ area or locality. So, in such a situation developing an Android App is very useful and can provide all the basic household services at the fingertip. As we all know, how apps have changed our lives in all aspects. So, taking this into consideration, an app is developed which will provide all the basic household services like plumbing, carpentry, electricity, etc., which will help the users in all aspects. This app is developed using different programming languages like Java, C, etc., and using different approaches. In this application, any user after creating his profile can look up any type of household services. The user gets a list view of all the service providers of that particular service. Also, the user can apply the filter to the list so that he/she gets a list of service providers around that particular region/ area. The user can make online payments through this app only. It is fully secure. After completion of the job, the user can also give feedback for the service. The service providers get jobs through this app, so they get employed also. This app is very useful as it provides all the services within this app only. The app is user-friendly and provides a safer environment for both customers and service providers.

Table Of Contents

S.No.	Title	Page No.
1.	Abstract	II
2.	List of Table	III
3.	List of Diagrams	IV
4.	Chapter 1	V
	1.1 Introduction	
	1.2 Formulation of Problem	
	1.3 Tool and Technology Used	VI

List of Figures

S.No.	Title	Page No.
1	Flowchart Diagram	
2	Data Flow Diagram Level 0	
3	Data Flow Diagram Level 1	
4	Data Flow Diagram Level 2	

Acronyms

B.Tech.	Bachelor of Technology
M.Tech.	Master of Technology
BCA	Bachelor of Computer Applications
MCA	Master of Computer Applications
B.Sc. (CS)	Bachelor of Science in Computer Science
M.Sc. (CS)	Master of Science in Computer Science
SCSE	School of Computing Science and Engineering

CHAPTER-1

Introduction

Due to fast developments in technology, smartphones are proven to be a vital source of communication and now have grown to be an imperative part of our everyday lives. Nowadays in India, human beings are facing problems in finding and hiring nearby professional experts to avail repairing job and maintenance services for their properties and offices. Because of the unavailability of such kind of system, new people coming to the city are also dealing with the predicament of discovering terrific work in their vicinity. To overcome this problem, it's immensely necessary to provide a platform in order to make bridge the communication gap between the technical employees and end-users which can provide a convenient and understandable interface for both the usage of the cutting-edge developments of technology. To get the great of smartphones, this work proposes an android utility or application for android smartphones with the name Helpmeet Assistance. Helpmeet Assistance can be a terrific begin in this context in India. It is an android app that can communicate with the consumer and experts possessing different technical skills. The app presently has only four classes of professionals i.e. painters, Electricians, Mechanics, and Plumbers but it is expandable.

Today we are living in a technologically superior era where the whole lot from education, business, and finance to amusement is accomplished through a cell machine and the use of quite a number of OS structures. More than 2 billion users are reachable on Facebook and other social media users in the same echelon. Its capacity by some means each eight out of 10 people comprehend the utilization of clever phones. According to, smart smartphone enterprise is growing day by way of day in India. Almost 72% of mobile users are using smart telephones and 52% access net on edge, 3G, and GPRS. Due to the widespread purpose of clever phones, all kinds of groups are gradually shifting on modern-day gadgets. Nowadays orders should be made via simply some clicks on smart telephones and wanted matters are acquired at the doorsteps.

People of India are additionally using many smartphone apps which are very popular in versatile contexts such as Foodpanda, zomato, Uber, Ola, swiggy, and many others. In today's gadget-friendly surroundings people are seeking for job and buyers are looking out for services without delay by clever phones. Mobile enterprise is now a predominant supply of employment opportunities. To serve in this recent developing vogue we have proposed and developed an android software which shows how an awful lot of consumers and the provider issuers are a ways away from every other. It will be used to set up connections between users and local professional employees to supply repairing and protection offerings and will decorate employment opportunities. This device affords the answer to all troubles associated with the home, workplace, or any different place. Helpmeet Assistance is a contribution in the direction of answers for neighborhood employees who are unaware of the electricity of smartphones. This App is designed and developed on the android platform, It is basically developed in kotlin, and has two main pages- One is for users who want any person for a job and one is for the job seeker.

The predominant goal of this work is to boost a gadget comprising of an internet site and an android app which consists of the following features:

- Provide alternatives for most wished workers i.e. Plumber, Electrician, Mechanic, and Decorator.
- Inform by way of a push notification with a work description to the worker
- Send acceptance push notifications to customers.
- Customer and worker can name or send SMS to each other

In recent years, data analytics and AI have become increasingly popular in various industries, from e-commerce to healthcare. These technologies have the potential to provide valuable insights and improve decision-making, but they also require significant resources and expertise to implement. A data-driven app using AI bot can help to overcome some of these challenges by providing a user-friendly interface and personalized recommendations based on data analysis. In this paper, we explore the design and implementation of a data-driven app using AI bot and its potential impact. India has greater unemployment and a large range of nearby workers. It is evident that many apps are working in India and have received recognition in a relatively brief time span like HouseJoy, UrbanClap, Mr. Right, CHEEP, and many others, with hundreds of thousands of users. However, there is a handful of android applications especially available for India, so the Helpmeet Assistance will be helpful as soon as people get started with it.

The app is developed for the android platform with the usage of Android Software Development KIT (SDK) and Android Studio. SDK is a collection of equipment that make easiness and assist in app development. It is an integral part of the android software improvement Android Studio is an official IDE (Integrated Development Environment) for the Android Operating System of Google. It is in particular built for Android Development and is based totally on IntelliJ IDEA software. Mostly Java is used however apps can also be constructed using C++. It aims at all dimensions of display gadgets like android clever phones, Tablets, Smart TVs, and Wearable devices. For coding, there is a featured editor and a layout designer. For the output, an Emulator is given which is additionally recognized as Android Virtual Device (AVD) which looks like an actual device. The android app first traces your location and then asks for the needed worker. It shows the people contacts near you. All the people need to click their picture and also their aadhar card which would be verified by the police also. This application will show users the data, picture, and aadhar card images with the contact details of people. Yet there are various other apps already available in the market. But this android app is different from all of them as this app is elated with various features.

Some of them are-

- This app provides details of all the service providers which are required in daily life like carpenters, plumbers, barbers, electricians, pest control, etc.
- This app is beneficial for both the persons requiring service as well as the service providers.
- For persons requiring service in a way that they can get details of the service providers around their locality as well as around any region / area. Secondly, it saves their time as they get services sitting at home itself.
- For the service providers in a sense that they get jobs/ orders sitting at homes.
- This app filters out the results of these service providers according to the region / area.
- It gives all the details of the persons providing services like name, age, address, mobile no., job experience, previous records, no. of complaints, etc.

This app overcomes various drawbacks available in previous apps and also it provides many services in a single app.

- This app provides details of various types of service providers like carpenters, plumbers, electricians, etc. which other apps do not provide.
- This app allows users to log in/ sign up as a person requiring service or a person providing service.
- This app allows payment via Online Payment Mode or Cash-on-Delivery Mode.

This app provides an option for filling out the feedback form after they get the required services to tell others whether the person is good or not in his/ her job.

It will be used to set up connections between users and local professional employees to supply repair and protection offerings and will decorate employment opportunities. This device affords the answer to all troubles associated with the home, workplace, or any different place. HELP MEET ASSISTANCE is a contribution in the direction of answers for neighborhood employees who are unaware of the electricity of smartphones. This App is designed and developed on the android platform, It is basically developed in kotlin, and has two main pages- One is for users who want any person for a job and one is for the job seeker.

The predominant goal of this work is to boost a gadget comprising of an internet site and an android app which consists of the following features:

- Provide alternatives for most wished workers i.e. Plumber, Electrician, Mechanic, and Decorator.
- Inform by way of a push notification with a work description to the worker. Send acceptance push notifications to customers.
- Customer and worker can name or send SMS to every other

Survey Of Technologies-

In an Android application, there is a scope for a large number of platforms, languages and frameworks to choose from. Before selecting from this large array of technologies, the following aspects, which are characteristic to android based application like this one, have been kept in mind:

- Data validation
- Performance
- Reliability
- Scalability
- Security
- Portability
- Performance
- Time constraint
- Cost constraint

WHAT IS ANDROID-

It is an open source software platform and operating system for mobile devices based on the Linux kernel and Developed by Google and later the Open Handset Alliance (OHA) allows writing managed code in the Java language android has its own virtual machine i.e. Android Studio, which is used for executing the android application. Android is a free downloadable open source software stack for mobile devices that include an Operating system.

Android OS is developed under a code name based on dessert items. The software stack is split into Four Layers:

- The application layer
- The application framework
- The libraries and runtime
- The kernel

Application Framework: - This is all written in a Java programming language and the application framework is the toolkit that all applications use. These applications include the ones that come with a phone like the home applications, or the phone application. It includes applications written by Google, and it includes apps that will be written by you. So, all apps use the same framework and the same APIs. These are as follows: -

Activity manager:-It manages the lifecycle of applications. It enables proper management of all the activities. All the activities are controlled by activity manager.

Resource manager:-It provides access to non-code resources such as graphics etc.

Notification manager:-It enables all applications to display custom alerts in status bar.

Location manager: - It fires alerts when user enters or leaves a specified geographical location.

Package manager:- It is use to retrieve the data about installed packages on device.

Window manager:- It is use to create views and layouts.

Telephony manager:- It is use to handle settings of network connection and all information about services on device.

Application Layer:- The final layer on top is Applications. It includes the home application, the contacts application, the browser, and apps. It is the most upper layer in android architecture. All the applications like camera, Google maps, browser, sms, calendars, contacts are native applications. These applications work with end user with the help of application framework to operate.

Android is a multi-process system, in which each application (and parts of the system) runs in its own process. Most security between applications and the system is enforced at the process level through standard Linux facilities, such as user and group IDs that are assigned to applications. Android is designed having multi-layer security which provides flexibility for this platform. When attackers attempt attack on device, android platform helps to reduce the portability of the attack.

System Analysis

Existing System:

The existing system requires applicants to search through print and visual media for job opportunities. Applicants need to apply for jobs using conventional methods and wait for further details for a long period of time. Employers need to advertise the vacancies and sort all applicant details, conduct selection procedures and complete the formalities. This approach is tedious and requires much effort and resources.

Proposed System

- Simple and professional GUI for all qualification groups.
- All vacancies are available on a single interface.
- Job Seekers can save jobs according to their requirement.
- Reduce paper work and extra cost.
- Filer and search facility for job seekers according to their requirement.
- Applicant gets instant response for any query through notification and other media.
- Easy posting of job vacancies by the employers
- Applicants can update their personal and educational details which helps in better filtering of matching jobs.

Constraints:

The interface is only provided in English hence the user must know English in order to use the application.

Login and password is used for identification of the user and there is no facility for guest users.

Only registered users have the rights to access the facilities provided by the system.

Assumptions:

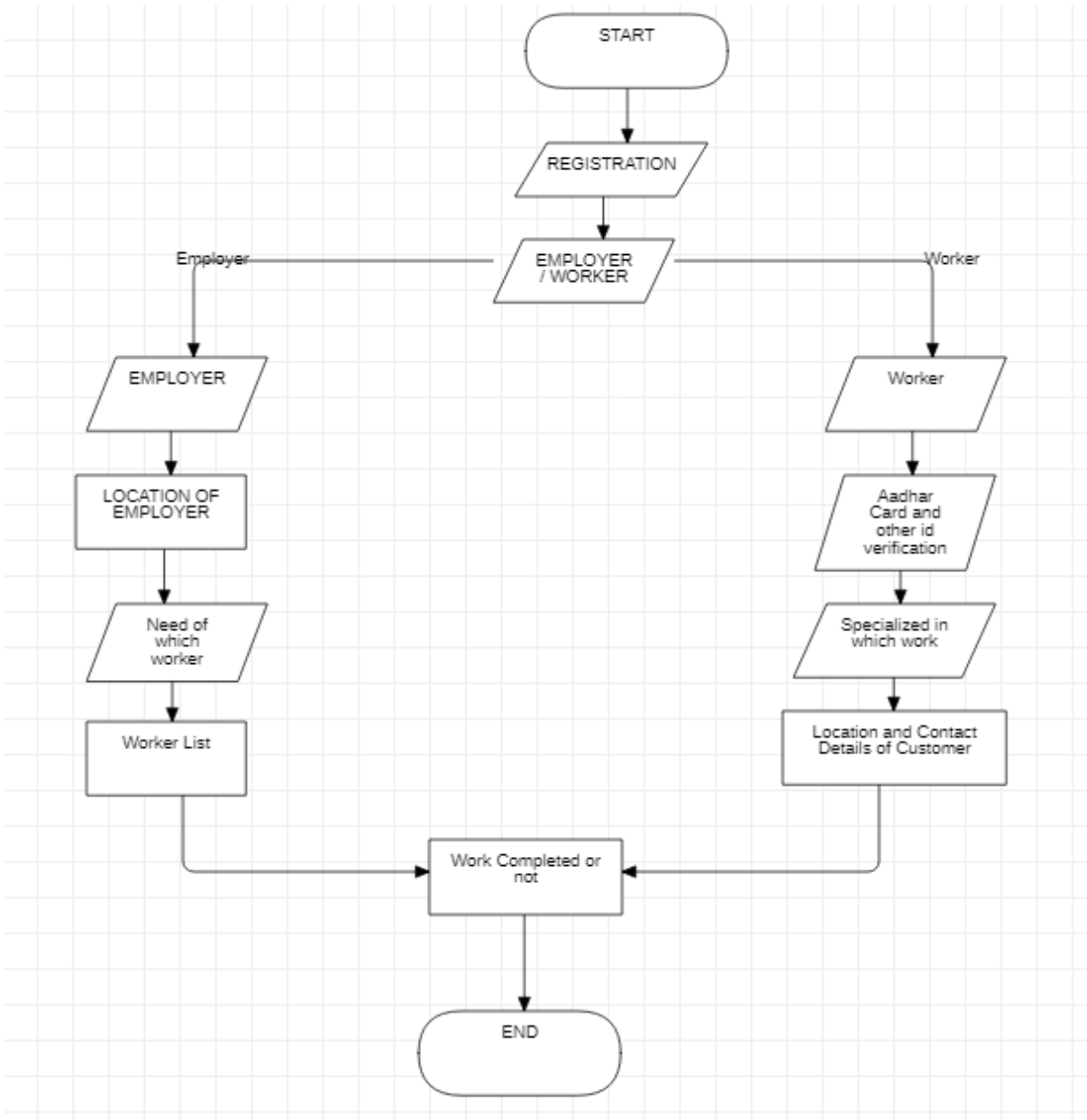
The user has intermittent knowledge of android devices and its UI.

As the system is an online system, the user must have an active internet connection on the device.

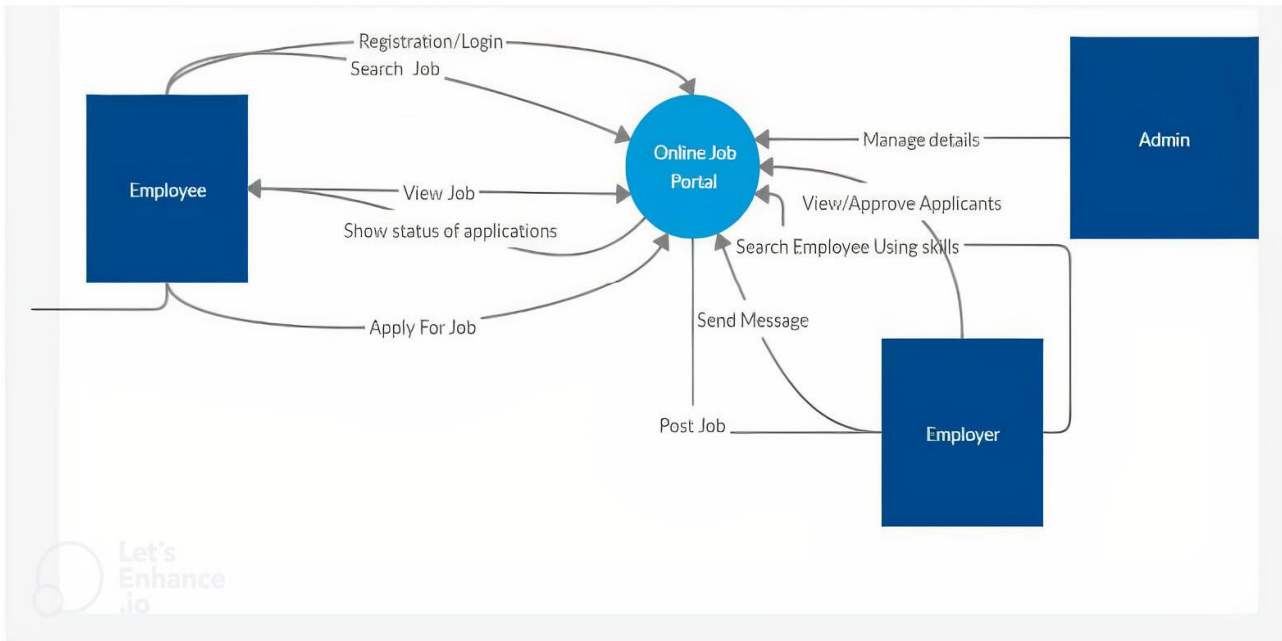
The user knows English as the GUI is provided in English.

The Job seeker has a degree/diploma in some recognized field of study.

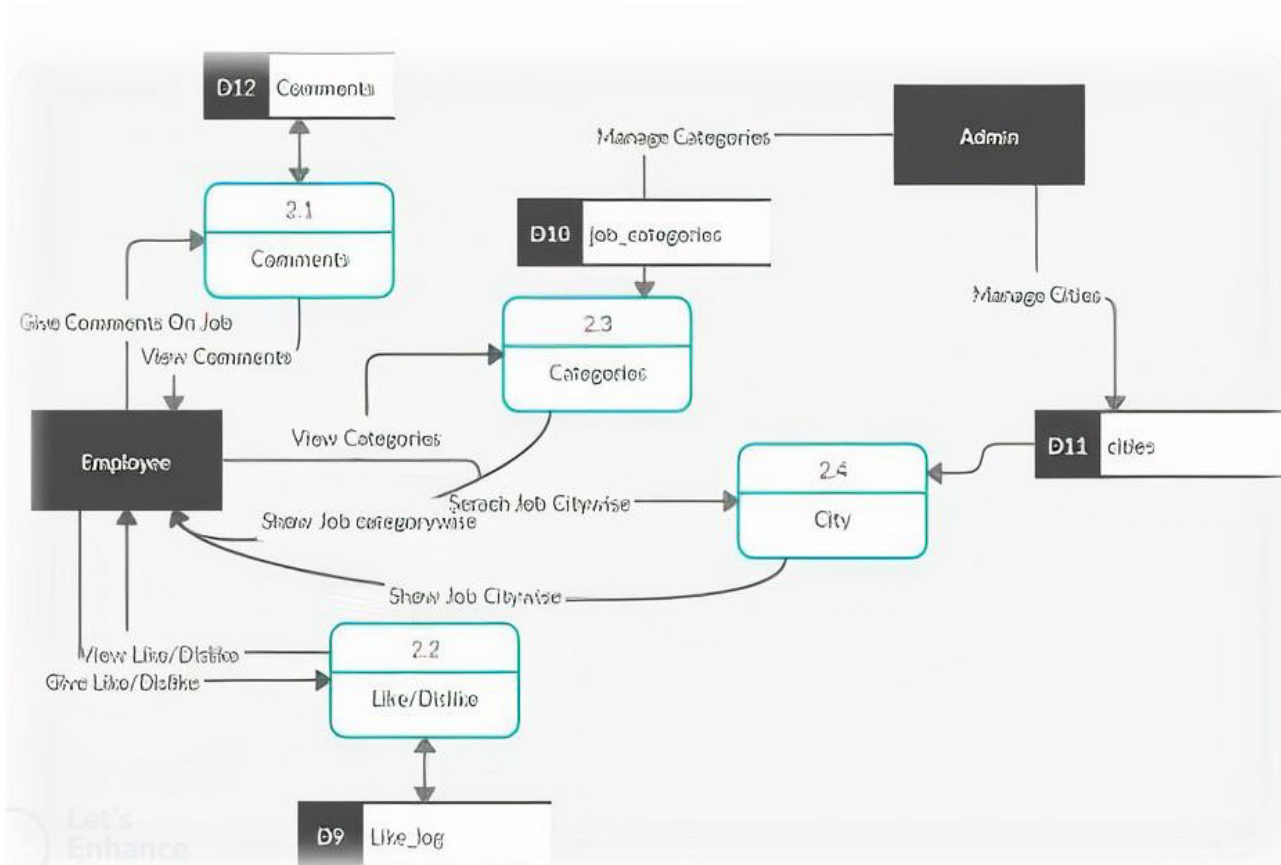
Flowchart-



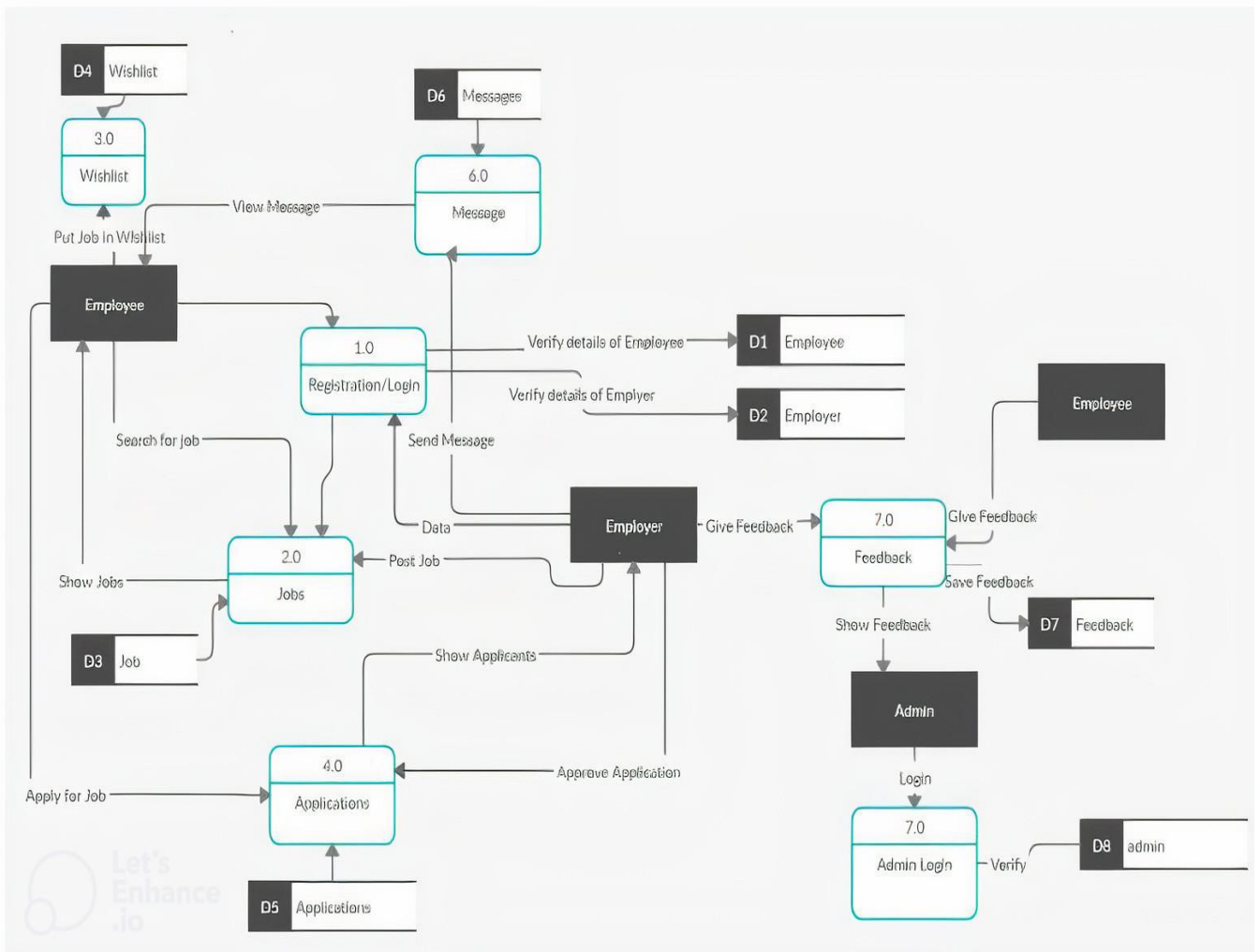
DFD Diagrams



Level 0



Level 1



Level 2

CHAPTER-2

Literature Survey

An Android Application is developed which provides almost all the basic household services. This android application provides a user-friendly and safe environment and interfaces for booking basic services. It is secure as it generates the QR code which is unique for customers and service providers. It will provide regular and timely notifications so that the user gets updated always. So, this app is very useful to all the persons and in all aspects. It provides all the basic household services at a reasonable and affordable cost and with much ease. Future Works-- The following features can make this app very effective -- In the future, the main focus is to provide a better interface for the app. It will also provide a satellite view of the service providers and the customers, as it is provided in Google Maps. The online database storage will be made large in size so that it can be used in a better way. It will also provide monthly statistical reports of the users and customers as well in the form of graphs and charts, which shows how much they used the app, i.e., active time and their activities in the past months. Map navigation can make the app better. It will help the service providers to find the location of the user and vice-versa. It will spot the location of both the users and service providers similar to Google maps. The problem is about developing an app that provides all the basic household services to all the people with ease.

The user has to create his ID in the app and after that, he can look for any type of service like plumbing, pest control, electricity, carpentry, etc. The user is provided with a list of service providers of that service. The user can check the profile of the service provider and then can book an appointment according to his choice. The user can make online payments through this app and after successful completion of the job the user can give feedback also. Before developing this app, firstly we have to consider the problem for which we need to develop this app. Household Services are the basic needs of life. If these services are not provided on the time, then it affects our lifestyle. So, a proper solution to these problems is necessary. In this busy life, it is not possible for everyone to go out and look for service providers in order to solve the problem. This process is very time taking also as it involves physical work also. Sometimes, it is also very difficult to get service providers in a locality as there may be chances that he is very far away, maybe we will not able to contact him, his price is very high, etc. In order to overcome these problems, this app is very useful as one can look for service providers of any service and book appointments. This is beneficial both for service providers and users as well. For users in a way, they get service by sitting at home only and with just one click, and for service providers in a way that they get employed and earn money. The app is very useful to all persons and individuals. The app can be easily used by people of every age group. As all the features are present in a single app so, one finds it very useful in the present scenario. The online database storage will be made large in size so that it can be used in a better way. It will also provide monthly statistical reports of the users and customers as well in the form of graphs and charts, which shows how much they used the app, i.e., active time and their activities in the past months. Map navigation can make the app better. It will help the service providers to find the location of the user and vice-versa. It will spot the location of both the users and service providers similar to Google maps. The problem is about developing an app that provides all the basic household services to all the people with ease. The user has to create his ID in the app and after that, he can look for any type of service like plumbing, pest control, electricity, carpentry, etc.

MODULE 3 Project Design

The user is provided with a list of service providers of that service.
This app provides details of all the service providers which are required in daily life like carpenters, plumbers, barbers, electricians, pest control, etc.



Sign up and Login.



Select Work For House



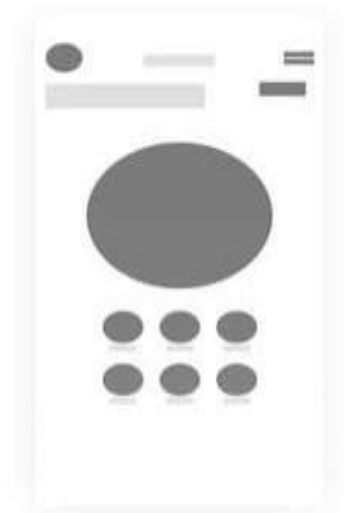
Your Profile



List Of Workers



Profile Of Workers



Settings

This app is beneficial for both the persons requiring service as well as the service providers. For persons requiring service in a way that they can get details of the service providers around their locality as well as around any region/area. Secondly, it saves their time as they get services sitting at home itself. The service providers in the sense that they get jobs/ orders sitting at home. This app filters out the results of these service providers according to the region/area. It will help the service providers find the user's location and vice-versa. It will spot the location of both the users and service providers similar to Google maps. The problem is about developing an app that provides all the basic household services to all the people with ease. The user has to create his ID in the app and after that, he can look for any type of service like plumbing, pest control, electricity, carpentry, etc. This article describes an AI-powered job assignment data program that helps anyone, everyone, and even small businesses in any profession help others. In this situation, it can be very convenient to develop an Android application and have all the basic tools at hand. As we all know, some apps have changed our lives in every situation. Therefore, keeping this in mind, an application has been developed that provides all the basic household services such as plumbing, carpentry, electricity, etc. to help users in every way. This program is developed using many programming languages like Java, C, etc.

A comparison of other articles shows that most of them only look at company employees, while others do not check profiles. Also, it is often difficult for small workers to find work. This program helps them to get daily wage jobs. Two types of profiles are created here. A profile that needs a job and another profile that needs a person after the job is done. You can also turn off the requirement and request another request.

Tools And Technology Used

Android Studio- This app is developed for Android platform using Android SDK and Android Studio. An SDK is a set of tools that make it easier to develop applications. It is an integral part of Android.

Java Software Enhancement - primarily uses Java, but can also write programs in Java. It targets display devices of all sizes, such as Android smartphones, tablets, smart TVs, and wearable devices. For coding, there is a featured editor and a layout designer.

For output, an emulator is provided, which is additionally recognized as an Android Virtual Device (AVD) and looks like a real device.

XML-Extensible Markup Language (XML) is a markup language and file format for storing, transferring, and reconstructing arbitrary data. It defines a set of rules for encoding documents in human-readable and machine-readable formats. The World Wide Web Consortium's 1998 XML 1.0 specification and many more related specifications are free and open standards.

API - An application programming interface (API) is a means by which two or more computer codes can contact among each other. It is a software interface that provides services to other software. [1] A document or standard describing how to build or use such a connection or interface is called an API specification. Computer systems conforming to this standard are expected to implement or expose the API. The term API can refer to a specification or an implementation. System requirements are expressed in a software requirement document. The Software requirement specification (SRS) is the official statement of what is required of the system developers. This requirement document includes the requirements definition and the requirement specification. The software requirement document is not a design document. It should set out what the system should do without specifying how it should be done. The requirement set out in this document is complete and consistent.

The software specification document satisfies the following: -

- It specifies the external system behaviors.
- It specifies constraints on the implementation.
- It is easy to change.
- It serves as a reference tool for system maintainers.
- It records forethought about the life cycle of the system.
- It characterizes acceptable responses to undesired events.

Following Non-functional requirements will be there in the online shopping portal. Secure access of confidential data (candidate's details). 24 X 7 availability. Better component design to get better performance at peak time. Flexible service based architecture will be highly desirable for future extension. Nonfunctional requirements define system properties and constraints. It arises through user needs, because of budget constraints or organizational policies, or due to external factors such as safety regulations, privacy registration and so on. Various other Non-functional requirements are:

- Security
- Reliability
- Maintainability
- Portability
- Extensibility
- Reusability
- Application Affinity/Compatibility
- Resource Utilization

Scope and Analysis of Project

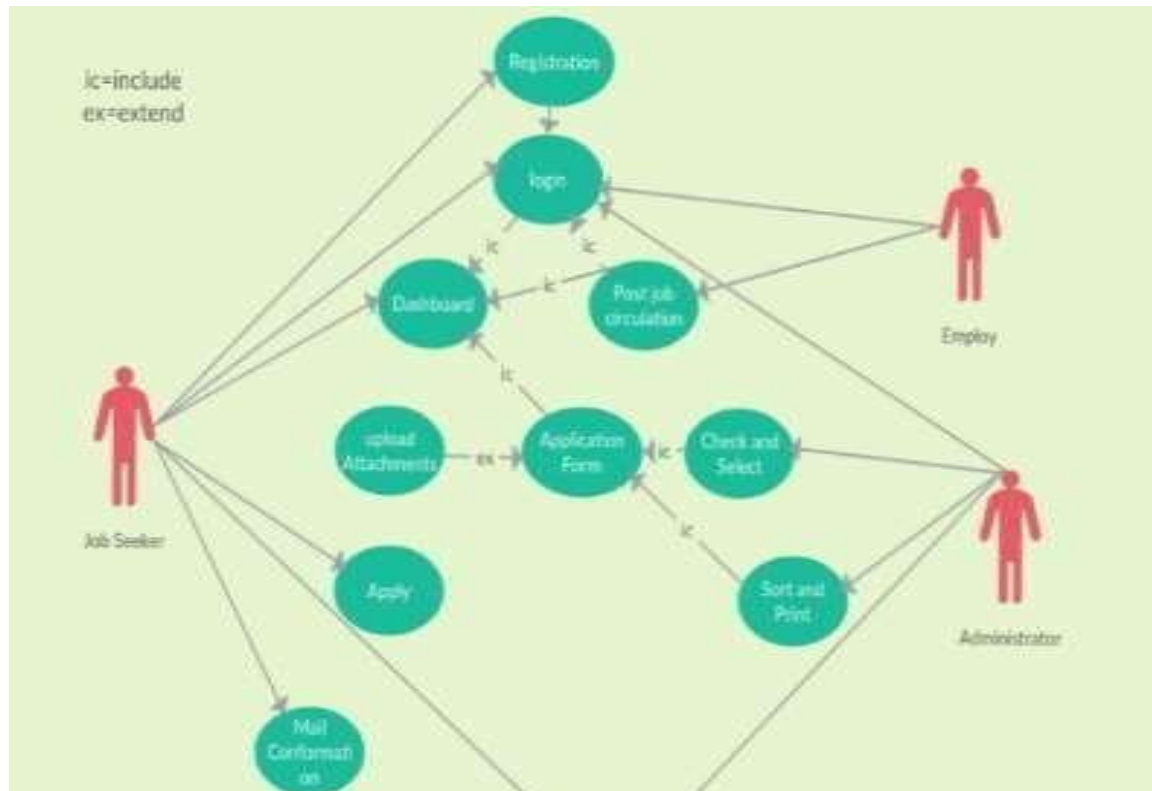
- Provide alternatives to workers who need them most, ie. Plumbers, electricians, mechanics, and decorators.
- Notify workers via push notifications with job descriptions. Send acceptance push notifications to customers. Customers and staff can give each other names or text each other.
- This app provides details of all the service providers you need in your daily life like carpenter, plumber, hairdresser, electrician, pest control etc.
- This application benefits both the person needing the service and the service provider.
- Provide full details such as name, age, address, mobile number, etc. of the person providing the service, work experience, past records, no. Complaints, etc.

Helpmeet Assistance acts as a portal between Job Seekers and Recruiters. It is helpful for fresher as well as young graduates, to get started with their career. Using Helpmeet Assistance, one can search job matching his or her qualification and appear for the selection process once the application is being registered and accepted. It can be very helpful as it allows users of different profile to upload their educational details and search matching job according to their qualification. Also, every user can apply for multiple job at a time. Helpmeet Assistance connects jobseekers and recruiters by accurately matching candidate profile to the relevant job openings through an advanced 2-way matching technology. While most online job portals focus on getting candidates the next job, Helpmeet Assistance focuses on the entire career growth of the candidate. The primary objective of Helpmeet Assistance is to provide a platform for young graduates and fresher to find the right and satisfactory job according to their qualification, to provide job opportunities to the job seekers, to meet manpower requirements of Industries and Projects in corporate sectors and to provide live vacancies. Helpmeet Assistance is designed with a view to bring the right job for the talented and bright candidates. It is developed with an eye to bridge the gap between talent and opportunities and offers end-to-end recruitment solutions. For individuals thinking about new job, new career, or new direction, Helpmeet Assistance helps you explore the possibilities and find the opportunities that are right for you.

CHAPTER-4

Module Description

Before developing this app, firstly we have to consider the problem for which we need to develop this app. Household Services are the basic needs of life. If these services are not provided on the time, then it affects our lifestyle. So, a proper solution to these problems is necessary. In this busy life, it is not possible for everyone to go out and look for service providers in order to solve the problem.



This process is very time taking also as it involves physical work also. Sometimes, it is also very difficult to get service providers in a locality as there may be chances that he is very far away, maybe we will not able to contact him, his price is very high, etc. In order to overcome these problems, this app is very useful as one can look for service providers of any service and book appointments.

Sign Up: The 'Sign up' activity allows the user to register himself in Helpmeet Assistance. It is to be noted that there is no option for 'Guest Login' in the application. Hence, any user willing to access the facilities provided by the application must register using proper information. During the signup process, the user must first enter a valid email address followed by a password of his/her choice (minimum 8 characters). After this process is completed, the user will be asked to fill some additional information regarding his/her personal and education details. Once this process is done, the user is directed to the 'Home' activity.

Login: The 'Login' facility is added for user authentication purpose. It helps to verify the credentials of an authorized user. In this activity the user needs to enter the proper credentials, matching the credentials entered during 'Signup' to access the rest of the functionalities. Home This is first layout the user interacts with after his/her registration is completed. Here he/she will find two tabs headed as 'Matched' and 'Recommended'. In 'Matched' the user will see jobs according to his qualification whereas in 'Recommended' he/she will see all the available jobs. The user can also filter the jobs acc.

to his/her Skill, Location and preferred Company. He/she can also search job according to desired field of work, say for example Mobile App Development, Web Developer, etc.

Activity: The 'Activity' layout keeps track of user's activity within the app. It is divided into three tabs headed as 'Viewed', 'Saved' and 'Applied'. In 'Viewed', the user can find all the jobs he/she has been viewing recently. In 'Saved', the user can find the job he/she has saved for later. In 'Applied', the user can find the jobs he/she has applied for.

Notification: The 'Notification' layout notifies the user once a job is successfully applied.

Feasibility Study:

It is necessary and prudent to evaluate the feasibility of a project at the earliest possible time. There may be different ways of checking whether a system is feasible or not. The following feasibility studies were performed to gauge the feasibility of the system.

Operational Feasibility:

In this test, the operational scope of the system is checked. The system under consideration should have enough operational reach. It is observed that the proposed system is very user friendly and since the system is built with enough help, even persons with little knowledge of android can find the system very easy.

Technical Feasibility:

This test includes a study of function, performance and constraints that may affect the ability to achieve an acceptable system. This test begins with an assessment of the technical viability of the proposed system. One of the main factors to be assessed is the need of various kinds of resources for the successful implementation for the proposed system.

Economic Feasibility:

An evaluation of development cost weighed against the ultimate income or benefit derived from the development of the proposed system is made. Care must be taken that incurred in the development of the proposed of the system should not exceed from the system. The income can be in terms of money or goodwill, since the software brings in both, the system is highly viable.

This is beneficial both for service providers and users as well. For users in a way, they get service by sitting at home only and with just one click, and for service providers in a way that they get employed and earn money. The app is very useful to all persons and individuals. The app can be easily used by people of every age group. It is very convenient for users to contact us, service providers can also contact us, the price is very high, etc. To overcome these problems, this app is very useful as it allows you to search and book any provider of any service. This is beneficial for both service providers and users. For users, this service is with one click from the comfort of their own home, and for service providers, through smartphones and access to the edge of 52%, 3G and GPRS. With the rise of smartphones, groups of all kinds are gradually moving towards the latest gadgets.

A list of service providers that provide services to the user. The user can view the provider's profile and book as desired. Users can pay online through the app and after success users can also provide their feedback. Before developing this application, we must first think about the issues related to the development of this application. The program currently only has four specialist categories: Painter, Electrician, Mechanic and Plumber, but it is scalable. Today, we live in an age of high technology, from education, business, finance to entertainment, all through the use of mobile phones and multiple computer system architectures. Use. More than 2 billion users reach the size of Facebook and other social media users. Its capacity is such that 8 out of 10 people know how to use a smartphone.

Project Code:

Activity Main:

```
<?xml version="1.0" encoding="utf-8"?>
```

```
<android.support.constraint.ConstraintLayout
```

```
    xmlns:android="http://schemas.android.com/apk/res/android"
```

```
    xmlns:tools="http://schemas.android.com/tools"
```

```
    xmlns:app="http://schemas.android.com/apk/res-auto"
```

```
    android:layout_width="match_parent"
```

```
    android:layout_height="match_parent"
```

```
    tools:context=".activity.MainActivity">
```

```
<android.support.design.widget.TextInputLayout
```

```
    android:layout_width="match_parent"
```

```
    android:layout_height="wrap_content"
```

```
    app:layout_constraintTop_toTopOf="parent"
```

```
    android:layout_marginTop="100dp" app:layout_constraintStart_toStartOf="parent"
```

```
    app:layout_constraintEnd_toEndOf="parent" app:layout_constraintHorizontal_bias="0.0"
```

```
    app:layout_constraintBottom_toBottomOf="parent"
```

```
    android:layout_marginBottom="480dp"
```

```
    android:layout_marginStart="8dp" android:layout_marginEnd="8dp"
```

```
    app:layout_constraintVertical_bias="0.0"
```

```
    android:id="@+id/textInputLayout4" android:hint="@string/taskhint">
```

```
<android.support.design.widget.TextInputEditText
```

```
    android:layout_width="match_parent"
```

```
    android:layout_height="wrap_content"
```

```
    android:id="@+id/jobId" android:textColor="@color/textcolor"/>
```


</android.support.design.widget.TextInputLayout>

<android.support.design.widget.TextInputLayout

android:layout_width="match_parent"

android:layout_height="55dp"

android:layout_marginTop="43dp"

app:layout_constraintTop_toBottomOf="@+id/textInputLayout4"

app:layout_constraintStart_toStartOf="parent"

app:layout_constraintEnd_toEndOf="parent" android:layout_marginStart="8dp"

app:layout_constraintBottom_toBottomOf="parent"

android:layout_marginBottom="424dp"

android:layout_marginEnd="8dp" app:layout_constraintHorizontal_bias="0.0"

app:layout_constraintVertical_bias="0.0" android:id="@+id/textInputLayout5">

<android.support.design.widget.TextInputEditText

android:layout_width="match_parent"

android:layout_height="48dp"

android:hint="@string/assignedto" android:id="@+id/acceptorId"

android:textColor="@color/textcolor"/>

</android.support.design.widget.TextInputLayout>

<android.support.design.widget.TextInputLayout

android:layout_width="match_parent"

android:layout_height="66dp"

android:layout_marginTop="43dp"

app:layout_constraintStart_toStartOf="parent"

app:layout_constraintEnd_toEndOf="parent" android:layout_marginStart="8dp"

android:layout_marginEnd="8dp"

app:layout_constraintHorizontal_bias="0.0"

```
        android:layout_marginBottom="300dp"
app:layout_constraintBottom_toBottomOf="parent"

        app:layout_constraintTop_toBottomOf="@+id/textInputLayout5"
app:layout_constraintVertical_bias="0.005"

        android:id="@+id/textInputLayout6">
```

```
<android.support.design.widget.TextInputEditText
```

```
        android:layout_width="match_parent"

        android:layout_height="45dp"

        android:hint="@string/assignedby" android:id="@+id/allocatorId"
android:textColor="@color/textcolor"/>
```

```
</android.support.design.widget.TextInputLayout>
```

```
<Button
```

```
        android:text="Record Data"

        android:layout_width="266dp"

        android:layout_height="53dp"

        android:id="@+id/saveButton" android:layout_marginTop="72dp"

        app:layout_constraintTop_toBottomOf="@+id/textInputLayout6"
android:layout_marginBottom="8dp"

        app:layout_constraintBottom_toBottomOf="parent"
app:layout_constraintVertical_bias="0.0"

        app:layout_constraintEnd_toEndOf="parent" android:layout_marginEnd="162dp"

        app:layout_constraintStart_toStartOf="parent" android:layout_marginStart="162dp"

        app:layout_constraintHorizontal_bias="0.502" android:textAllCaps="true"
android:textStyle="bold"

        android:textAlignment="center" android:background="@color/colorPrimary"
android:textColor="#FBF8F8"/>
```

```
</android.support.constraint.ConstraintLayout>
```

Row List:

```
<?xml version="1.0" encoding="utf-8"?>
```

```
<android.support.constraint.ConstraintLayout
```

```
xmlns:android="http://schemas.android.com/apk/res/android"
```

```
xmlns:app="http://schemas.android.com/apk/res-auto"
```

```
xmlns:tools="http://schemas.android.com/tools"
```

```
android:layout_width="match_parent"
```

```
android:layout_height="wrap_content">
```

```
<android.support.v7.widget.CardView
```

```
android:padding="10dp"
```

```
app:layout_constraintVertical_bias="0.21"
```

```
android:layout_width="match_parent"
```

```
android:layout_height="wrap_content"
```

```
app:layout_constraintStart_toStartOf="parent"
```

```
android:layout_marginTop="8dp"
```

```
android:layout_marginBottom="8dp"
```

```
app:layout_constraintLeft_toLeftOf="parent"
```

```
app:layout_constraintRight_toRightOf="parent"
```

```
app:cardBackgroundColor="@color/newcolor"
```

```
app:layout_constraintBottom_toBottomOf="parent"
```

```
app:layout_constraintTop_toTopOf="parent"
```

```
app:layout_constraintEnd_toEndOf="parent" android:layout_marginStart="10dp"
```

```
android:layout_marginEnd="10dp">
```

```
<RelativeLayout
```

```
android:padding="5dp"
```

```
android:layout_width="match_parent"
```

```
android:layout_height="match_parent"
```

```
>
```

```
<ImageView
```

```
android:contentDescription="@string/app_name"
```

```
android:layout_width="82dp"
```

```
android:layout_height="86dp" app:srcCompat="@mipmap/button"
```

```
android:id="@+id/ImageViewId"
```

```
android:layout_marginLeft="10dp"
```

```
/>
```

```
<TextView
```

```
android:text="Job Name"
```

```
android:textStyle="bold"
```

```
android:textSize="18dp"
```

```
android:layout_marginLeft="10dp"
```

```
android:textColor="@color/design_default_color_primary"
```

```
android:textAlignment="center"
```

```
android:layout_toRightOf="@+id/ImageViewId"
```

```
android:layout_width="wrap_content"
```

```
android:layout_height="wrap_content"
```

```
android:id="@+id/ListJobName"
```

```
android:layout_alignParentStart="true" android:layout_marginStart="97dp"/>
```

```
<TextView
```

```
android:text="AssignedBy"
```

```
android:layout_width="wrap_content"
```

```
android:layout_height="wrap_content" android:id="@+id/ListAssignedBy"
```

```
android:layout_below="@id/ListJobName"
```

android:layout_marginLeft="10dp"

android:layout_toRightOf="@id/ImageViewId"/>

<TextView

android:layout_width="wrap_content" android:layout_height="wrap_content"

android:id="@+id/ListDate"

android:layout_alignParentRight="true"

android:layout_alignBottom="@id/ImageViewId"

android:text="Date: 13 July 2019"

android:layout_marginRight="14dp"

>

<Button android:id="@+id/EditButton"

android:layout_width="25dp"

android:layout_height="25dp"

android:background="@android:drawable/ic_menu_edit"

android:elevation="11dp"

android:layout_alignParentTop="true"

android:layout_alignParentEnd="true"

android:layout_marginEnd="35dp" />

android:layout_marginBottom="-1dp"

android:layout_marginLeft="40dp"

>

<Button

android:id="@+id/DeleteButton"

android:layout_width="25dp"

```
android:layout_height="25dp"
android:layout_marginLeft="9dp"

android:background="@android:drawable/ic_delete"
android:layout_alignParentTop="true"
android:layout_alignParentEnd="true" />
```

```
<TextView
```

```
    android:layout_width="wrap_content" android:layout_height="wrap_content"
    android:id="@+id/ListAssignedTo"
    android:text="AssignedTo"
    android:layout_below="@id/ListAssignedBy"
    android:layout_toRightOf="@id/ImageViewId"
    android:layout_marginLeft="10dp"
    android:textStyle="italic"
```

```
/>
```

```
</RelativeLayout>
```

```
</android.support.v7.widget.CardView>
```

```
</android.support.constraint.ConstraintLayout>
```

Job List:

```
package com.example.android_kotlin_job_allocation_app.activity
```

```
import android.content.Intent
```

```
import android.support.v7.app.AppCompatActivity
```

```
import android.os.Bundle
```

```
import android.support.v7.app.AlertDialog
```

```
import android.support.v7.widget.LinearLayoutManager
import android.support.v7.widget.RecyclerView
import android.text.TextUtils
import android.util.Log
import android.view.LayoutInflater
import android.view.Menu
import android.view.MenuItem
import com.example.android_kotlin_job_allocation_app.R
import com.example.android_kotlin_job_allocation_app.data.JobDatabaseHandler
import com.example.android_kotlin_job_allocation_app.data.JobListAdapter
import com.example.android_kotlin_job_allocation_app.model.job
import kotlinx.android.synthetic.main.activity_job_list.*
import kotlinx.android.synthetic.main.activity_main.*
import kotlinx.android.synthetic.main.popup.view.*

class JobList : AppCompatActivity() {

    private var adapter: JobListAdapter? = null
    private var joblist: ArrayList<job>? = null
    private var jobListItems: ArrayList<job>? = null
    private var dialogBuilder: AlertDialog.Builder? = null
    private var dialog: AlertDialog? = null

    private var layoutManager: RecyclerView.LayoutManager? = null
    var dbhandler: JobDatabaseHandler? = null
```

```
override fun onCreate(savedInstanceState: Bundle?) {  
    super.onCreate(savedInstanceState)  
    setContentView(R.layout.activity_job_list)  
  
    dbhandler = JobDatabaseHandler(this)  
  
    layoutManager = LinearLayoutManager(this)  
    joblist = ArrayList<job>()  
    jobListItems = ArrayList()  
    adapter = JobListAdapter(jobListItems!!, this)  
  
    //setup list = recycler view  
    recyclerViewId.layoutManager = layoutManager  
    recyclerViewId.adapter = adapter  
  
    // load our jobs  
    joblist = dbhandler!!.readJobs()  
    joblist!!.reverse()  
  
    for (c in joblist!!.iterator()) {  
  
        val job = job()  
        job.jobName = c.jobName  
        job.assignedBy = "Assigned By: ${c.assignedBy}"  
        job.assignedTo = "Assigned to: ${c.assignedTo}"  
    }  
}
```



```
job.id = c.id
```

```
job.showHumanDate(c.timeAssigned!!)
```

```
jobListItems!!.add(job)
```

```
}
```

```
adapter!!.notifyDataSetChanged()
```

```
}
```

```
override fun onCreateOptionsMenu(menu: Menu?): Boolean {
```

```
    menuInflater.inflate(R.menu.top_menu, menu) //menu object is passed here declared in  
    overriding function
```

```
    return true
```

```
}
```

```
override fun onOptionsItemSelected(item: MenuItem?): Boolean { //this function is used to  
provide and onClickListener to the add menu button
```

```
if (item!!.itemId == R.id.add_menu_button){
```

```
    Log.d("chutiya", "bht bda")
```

```
    createPopupDialog()
```

```
}
```

```
return super.onOptionsItemSelected(item)
```

```
}
```

```
fun createPopupDialog(){ // instantiate dialog builder and dialog
```

```
    var view = inflater.inflate(R.layout.popup, null) // this view will have our popup
```

```
    var jobName = view.popenterjobId
```

```
    var assignedBy = view.popenterallocatorId
```

```
    var assignedTo = view.popenteracceptorId
```

```
    var savejob = view.popsaveButtonId
```

```
    // instantiating dialog builder
```

```
    dialogBuilder = AlertDialog.Builder(this).setView(view)
```

```
    dialog = dialogBuilder!!.create()
```

```
    dialog?.show()
```

```
    savejob.setOnClickListener {
```

```
        if (!TextUtils.isEmpty(jobName.text.toString().trim()))
```

```
            && !TextUtils.isEmpty(assignedBy.text.toString().trim()))
```

```
            && !TextUtils.isEmpty(assignedTo.text.toString().trim())){
```

```
                var job = job()
```

```
                job.jobName = jobName.text.toString().trim()
```

```
                job.assignedBy = assignedBy.text.toString().trim()
```

```
                job.assignedTo = assignedTo.text.toString().trim()
```

```
dbhandler!!.createJob(job)

dialog!!.dismiss()

startActivity(Intent(this, JobList :: class.java))

finish()
```

```
}
```

```
}
```

```
}
```

```
}
```

Main Activity:

```
package com.example.android_kotlin_job_allocation_app.activity
```

```
import android.app.ProgressDialog
```

```
import android.content.Intent
```

```
import android.support.v7.app.AppCompatActivity
```

```
import android.os.Bundle
```

```
import android.support.v7.widget.LinearLayoutManager
```

```
import android.support.v7.widget.RecyclerView
```

```
import android.text.TextUtils
```

```
import android.util.Log
```

```
import android.widget.ProgressBar
```

```
import android.widget.Toast
```

```
import com.example.android_kotlin_job_allocation_app.R
```

```
import com.example.android_kotlin_job_allocation_app.data.JobDatabaseHandler
```

```
import com.example.android_kotlin_job_allocation_app.data.JobListAdapter
```

```
import com.example.android_kotlin_job_allocation_app.model.job
```

```
import kotlinx.android.synthetic.main.activity_main.*
```

```
class MainActivity : AppCompatActivity() {
```

```
    var dbhandler: JobDatabaseHandler? = null
```

```
    var progressBar: ProgressDialog? = null
```

```
override fun onCreate(savedInstanceState: Bundle?) {
```

```
    super.onCreate(savedInstanceState)
```

```
    setContentView(R.layout.activity_main)
```

```
    progressBar = ProgressDialog(this)
```

```
    dbhandler = JobDatabaseHandler(this)
```

```
    checkDB()
```

```
    saveButton.setOnClickListener {
```

```
        progressBar!!.setMessage("saving...")
```

```
        progressBar!!.show()
```

```
if (!TextUtils.isEmpty(jobId.text.toString()) && !TextUtils.isEmpty(allocatorId.text.toString())
&&
    !TextUtils.isEmpty(acceptorId.text.toString())){

    // save to database

    var job: job = job()

    job.jobName = jobId.text.toString()

    job.assignedTo = acceptorId.text.toString()

    job.assignedBy = allocatorId.text.toString()

    saveTODB(job)

    progressBar!!.cancel()

    startActivity(Intent(this, JobList::class.java))

}

else {

    Toast.makeText(this, "Please enter the information", Toast.LENGTH_LONG).show()

}

}

}

fun saveTODB(job: job){
```

```

        dbhandler!!.createJob(job)
    }

    fun checkDB(){//checking if database has some data then it will redirect user to second job list
page
        if (dbhandler!!.getJobsCount() > 0)
            startActivity(Intent(this, JobList :: class.java))

    }
}

```

Job DataBase Handler:

```

package com.example.android_kotlin_job_allocation_app.data

import android.content.ContentValues
import android.content.Context
import android.database.Cursor
import android.database.sqlite.SQLiteDatabase
import android.database.sqlite.SQLiteOpenHelper
import android.util.Log
import com.example.android_kotlin_job_allocation_app.model.*
import java.text.DateFormat
import java.util.*

class JobDatabaseHandler(context: Context):
    SQLiteOpenHelper(context, DATABASE_NAME, null, DATABASE_VERSION) {

```

```

override fun onCreate(db: SQLiteDatabase?) {

    // use SQL to create table

    var CREATE_JOB_TABLE = "CREATE TABLE " + TABLE_NAME + "(" + KEY_ID + "
    INTEGER PRIMARY KEY,"+

        KEY_JOB_NAME + " TEXT," +

        KEY_JOB_ASSIGNED_BY + " TEXT," +

        KEY_JOB_ASSIGNED_TO + " TEXT," +

        KEY_JOB_ASSIGNED_TIME + " LONG" +");"

    db?.execSQL(CREATE_JOB_TABLE)

}

```

```

override fun onUpgrade(db: SQLiteDatabase?, oldVersion: Int, newVersion: Int) {

    db?.execSQL("DROP TABLE IF EXISTS " + TABLE_NAME)

    // create a new table after deleting the previous using above function

    onCreate(db)

}

```

/ CRUD = create read update delete */*

```

fun createJob(job: job) {

    var db: SQLiteDatabase = writableDatabase

    var values: ContentValues = ContentValues()

```

```

values.put(KEY_JOB_NAME, job.jobName)

values.put(KEY_JOB_ASSIGNED_BY, job.assignedBy)

values.put(KEY_JOB_ASSIGNED_TO, job.assignedTo)

values.put(KEY_JOB_ASSIGNED_TIME, System.currentTimeMillis())

db.insert(TABLE_NAME, null, values)

Log.d("DEBUG", "SUCCESS")

db.close()
}

```

```

fun readAJob(id: Int): job {

    var db: SQLiteDatabase = writableDatabase

    var cursor: Cursor = db.query(
        TABLE_NAME, arrayOf(
            KEY_ID, KEY_JOB_NAME, KEY_JOB_ASSIGNED_BY,
            KEY_JOB_ASSIGNED_TO, KEY_JOB_ASSIGNED_TIME
        ), KEY_ID + "=?", arrayOf(id.toString()),
        null, null, null, null
    )

    if (cursor != null)
        cursor.moveToFirst()

    var job = job()

```



```
job.id = cursor.getInt(cursor.getColumnIndex(KEY_ID))
job.jobName = cursor.getString(cursor.getColumnIndex(KEY_JOB_NAME))
job.assignedBy = cursor.getString(cursor.getColumnIndex(KEY_JOB_ASSIGNED_BY))
job.assignedTo = cursor.getString(cursor.getColumnIndex(KEY_JOB_ASSIGNED_TO))
job.timeAssigned = cursor.getLong(cursor.getColumnIndex(KEY_JOB_ASSIGNED_TIME))
```

```
var dateFormat: java.text.DateFormat = DateFormat.getDateInstance()
```

```
var formatteddate = dateFormat.format(
    Date(
        cursor.getLong(
            (cursor.getColumnIndex(KEY_JOB_ASSIGNED_TIME))
        ).time
    )
)
```

```
return job
```

```
}
```

```
fun readJobs(): ArrayList<job> {
```

```
var db: SQLiteDatabase = readableDatabase
```

```
var list: ArrayList<job> = ArrayList()
```

```
//Select all chores from table
```

```
var selectAll = "SELECT * FROM " + TABLE_NAME
```

```
var cursor: Cursor = db.rawQuery(selectAll, null)
```

```
//loop through our chores
```

```
if (cursor.moveToFirst()) {
```

```
  do {
```

```
    var job = job()
```

```
    job.id = cursor.getInt(cursor.getColumnIndex(KEY_ID))
```

```
    job.jobName = cursor.getString(cursor.getColumnIndex(KEY_JOB_NAME))
```

```
    job.assignedTo = cursor.getString(cursor.getColumnIndex(KEY_JOB_ASSIGNED_TO))
```

```
    job.timeAssigned = cursor.getLong(cursor.getColumnIndex(KEY_JOB_ASSIGNED_TIME))
```

```
    job.assignedBy = cursor.getString(cursor.getColumnIndex(KEY_JOB_ASSIGNED_BY))
```

```
    list.add(job)
```

```
  }while (cursor.moveToNext())
```

```
}
```

```
return list
```

```
}
```

```
fun updatejob(job: job): Int {
```

```
var db: SQLiteDatabase = writableDatabase
```

```
var values: ContentValues = ContentValues()
```

```
values.put(KEY_JOB_NAME, job.jobName)
```

```
values.put(KEY_JOB_ASSIGNED_BY, job.assignedBy)
```

```
values.put(KEY_JOB_ASSIGNED_TO, job.assignedTo)
```

```
values.put(KEY_JOB_ASSIGNED_TIME, System.currentTimeMillis())
```

```
//update a row
```

```
return db.update(TABLE_NAME, values, KEY_ID + "=?", arrayOf(job.id.toString()))
```

```
}
```

```
fun deletejob(id: Int) {
```

```
var db: SQLiteDatabase = writableDatabase
```

```
db.delete(TABLE_NAME, KEY_ID + "=?", arrayOf(id.toString()))
```

```
db.close()
```

```
}
```

```
fun getJobsCount(): Int {
```

```
var db: SQLiteDatabase = readableDatabase
```

```
var countQuery = "SELECT * FROM " + TABLE_NAME
```

```
var cursor: Cursor = db.rawQuery(countQuery, null)
```

```
return cursor.count
```

```
}
```

```
}
```

References

- [1] JOHNSON S M. Optimal two and three-stage production schedules with setup times included[J]. Naval Research Logistics Quarterly,1954,1(1):61 - 68.
- [2]. ADAMS J,BALAS E,ZAWACK D. The shifting Management Science,1988,34(3):391 - 401.
- [3]. DAVIS L. Job shop scheduling with genetic algorithms[C]/ /Proceedings of an International Conference on Genetic Algorithms and Their Applications,1985:136 - 140.
- [4]. GONCALVES J F,MAGALHAES MENDES J J,RESENDE M G C.A hybrid genetic algorithm for the job shop scheduling problem[J].Computers & Operations scheduling problem [J].European Journal of Research,2007,34(11):3229-3242
- [5]. ZHANG C Y,RAO Y Q,LI P G. An effective genetic algorithm for the job shop scheduling problem[J].International Journal of Advanced 282 Manufacturing Technology,2008,39:965 - 974.
- [6] <https://www.pewresearch.org/internet/2017/05/03/the-future-of-jobs-and-jobs-training/>
- [7]. Yildirim M B., Mouzon G. Single-Machine Sustainable Production Planning to Minimize Total Energy Consumption and Total Completion Time Using a Multiple Objective Genetic Algorithm[J]. 2012,59(4):585-597.
- [8]. GONCALVES J F,RESENDE M G C. An Research,2015,53:154 - 164. extended akers graphical method with a biased random-key genetic algorithm for job shop <http://docs.spring.io/spring/docs/current/spring-frame,scheduling>[J].International Transactions in Operational Research,2014,21(2):215 - 246.
- [9]. Taillard E D. Parallel taboo search techniques for the job shop scheduling problem[J]. ORSA Journal on Computing. 1994,2(6):108-117.
- [10].NOWICKI E,SMUTNICKI C.A fast taboo search algorithm for the job shop problem[J]. Management Science,1996,42 (6):797-813
- [11]. <https://en.wikipedia.org/wiki/API>
- [12]. NOWICKI E,SMUTNICKI C. Some new tools to solve the job shop problem[R]. Institute of Engineering Cybernetics , Wroclaw University of Technology,2002.
- [13]. ZHANG C Y,LI P G,GUAN Z L,et al.A tabu search algorithm with a new neighborhood structure for the job Management Science,1988,34(3):391 – 401
- [14]. <https://en.wikipedia.org/wiki/XML>