o

SYNOPSIS

This project acts as an important role in saving life of human beings and which is also its main aim. The project Android Blood Bank system is developed so that users can view the information about registered blood donors such as name, address, and other such personal information along with their details of blood group and other medical information of donor. The project also has a login page where in the user is required to register and only then can view the availability of blood and may also register to donate blood if he/she wishes to. Thus, this application helps to select the right donor instantly using medical details along with the blood group. The main aim of developing this application is to reduce the time to a great extent that is spent in searching for the right donor and the availability of blood required. Thus, this application provides the required information in no time and also helps in quicker decision making.

CHAPTER 1

1. INTRODUCTION

1.1 ORGANIZATION PROFILE

Tata Consultancy Services Ltd (TCS), a subsidiary of Tata Sons Pvt Ltd, is a provider of information technology (IT) services. It offers IT infrastructure services, engineering, and industrial services, business intelligence, business process outsourcing, and consulting services, cloud services, quality engineering, block chain, enterprise solutions, and IoT. The company offers business solutions to various industries, including banking, financial services, communication, media, technology, insurance, life sciences and healthcare, retail, consumer goods and distribution, manufacturing, information services, Hi-Tech, education, energy and utilities, life sciences, healthcare, travel, transportation, and hospitality. It's software products comprise TCS BaNCS, Ignio, TAP, TCS iON, TCS TwinX, TCS Optumera, TCS OmniStore, TCS ADD, TCS HOBS, Quartz, Jile, and TCS MasterCraft. The company has operations across Latin America, North America, Europe, Asia-Pacific, the Middle East, and Africa. TCS is headquartered in Mumbai, Maharashtra, India.

TCS offers a consulting-led, cognitive-powered, integrated portfolio of business,

technology, and engineering services and solutions. This is delivered through its unique Location Independent Agile™ delivery model, recognized as a benchmark of excellence in software development.

A part of the Tata group, India's largest multinational business group, TCS has over 592,000 of the world's best-trained consultants in 55 countries. The company generated consolidated revenues of US $25.7 billion in the fiscal year ended March 31, 2022, and is listed on the BSE (formerly Bombay Stock Exchange) and the NSE (National Stock Exchange) in India.

1.2 SYSTEM SPECIFICATION

System Requirements Specification also known as Software Requirements Specification, is a document or set of documentation that describes the features and behavior of a software application

WINDOWS OS

Windows is a graphical operating system developed by Microsoft. It allows users to view and store files, run the software, play games, watch videos, and provides a way to connect to the internet. It was released for both home computing and professional works.

Microsoft introduced the first version as 1.0

It was released for both home computing and professional functions of Windows on 10 November 1983. Later, it was released on many versions of Windows as well as the current version, Windows 10.

In 1993, the first business-oriented version of Windows was released, which is known as Windows NT 3.1. Then it introduced the next versions, Windows 3.5, 4/0, and Windows 2000. When the XP Windows was released by Microsoft in 2001, the company designed its various versions for a personal and business environment. It was designed based on standard x86 hardware, like Intel and AMD processor. Accordingly, it can run on different brands of hardware, such as HP, Dell, and Sony computers, including home-built PCs.

Play Video

Editions of Windows

Microsoft has produced several editions of Windows, starting with Windows XP. These versions have the same core operating system, but some versions included advance features with an additional cost. There are two most common editions of Windows:

- Windows Home
- Windows Professional

Windows Home is basic edition of Windows. It offers all the fundamental functions of Windows, such as browsing the web, connecting to the Internet, playing video games, using office software, watching videos. Furthermore, it is less expensive and comes pre-installed with many new computers.

JAVA

Java is a high-level programming language developed by Sun Microsystems. It was originally designed for developing programs for set-top boxes and handheld devices, but later became a popular choice for creating web applications.

The Java syntax is similar to C++, but is strictly an object-oriented programming language. For example, most Java programs contain classes, which are used to define objects, and methods, which are assigned to individual classes. Java is also known for being stricter than C++, meaning variables and functions must be explicitly defined. This means Java source code may produce errors or "exceptions" more easily than other languages, but it also limits other types of errors that may be caused by undefined variables or unassigned types.

Unlike Windows executable (.EXE files) or Macintosh applications (.APP files), Java programs are not run directly by the operating system. Instead, Java programs are interpreted by the Java Virtual Machine, or JVM, which runs on multiple platforms. This means all Java programs are multiplatform and can run on different platforms, including Macintosh, Windows, and Unix computers. However, the JVM must be installed for Java applications or applets to run at all. Fortunately, the JVM is included as part of the Java Runtime Environment (JRE),

ORACLE

An Oracle database is a collection of data treated as a unit. The purpose of a database is to store and retrieve related information. A database server is the key to solving the problems of information management. In general, a server reliably manages a large amount of data in a multiuser environment so that many users can concurrently access the same data. All this is accomplished while delivering high performance. A database server also prevents unauthorized access and provides efficient solutions for failure recovery.

Oracle Database is the first database designed for enterprise grid computing, the most flexible and cost effective way to manage information and applications. Enterprise grid computing creates large pools of industry-standard, modular storage and servers. With this architecture, each new system can be rapidly provisioned from the pool of components. There is no need for peak workloads, because capacity can be easily added or reallocated from the resource pools as needed.

The database has logical structures and physical structures. Because the physical and logical structures are separate, the physical storage of data can be managed without affecting the access to logical storage structures.

The Oracle grid architecture pools large numbers of servers, storage, and networks into a flexible, on-demand computing resource for enterprise computing needs. The grid computing infrastructure continually analyzes demand for resources and adjusts supply accordingly.

For example, you could run different applications on a grid of several linked database servers. When reports are due at the end of the month, the database administrator could automatically provision more servers to that application to handle the increased demand.

1.2.1 HARDWARE SPECIFICATION

     ▫   Processor   : P 4 700 GHz.

     ▫   RAM   : 4 GB RAM

     ▫   Hard Disk Drive   : 180 GB

1.2.2 SOFTWARE SPECIFICATION

Operating System   : Windows 7/8/10

 Front End   : JAVA

 Back End   : ORACLE

CHAPTER 2

SYSTEM STUDY

2.1 EXISTING SYSTEM

Blood bank system normally used in when do we need a blood just contacts with friends, friends of friends or hospital. This takes more time at an emergency time. Normally the donor has to go hospital and give their personal details then the hospital nurse and doctors has contact them. This may not use at the emergency situation.

2.1.1 DRAWBACKS

- Takes too much time for contact the donor
- Someone not attending the call
- Can't get updated details

2.2 PROPOSED SYSTEM

In this application is a mobile application, every user can have mobile so we just getting GPS location on every 5 minutes. So, we can track the latest donor location. Have an option for finding the nearby donors' details so we can easily contact the nearby donor. It's very helps to find the donor who are all willing to donate the blood.

2.2.1 FEATURES

- Contact donor immediately
- Avoid fake donors

- Very fast and secure mobile app

CHAPTER 3

SYSTEM DESIGN AND DEVELOPMENT

3.1 FILE DESIGN

The selection of the file system design approach is done according to the needs of the developers what are the needed requirements and specifications for the new design. It allowed us to identify where our proposal fitted in with relation to current and past file system development. Our experience with file system development is limited so the research served to identify the different techniques that can be used. The variety of file systems encountered show what an active area of research file system development is. The file systems may be from one of the two fundamental categories. In one category, the file system is developed in user space and runs as a user process. Another file system may be developed in the kernel space and runs as a privileged process. Another one is the mixed approach in which we can take the advantages of both aforesaid approaches. Each development option has its own pros and cons. In this article, these design approaches are discussed.

A file system is the data structure designed to support the abstraction of the data blocks as an archive and collection of files. This data structure is unique because it is stored on secondary storage (usually the disk), which is a very slow device.

The file system structure is the most basic level of organization in an operating system. Almost all of the ways an operating system interacts with its users, applications, and security model are dependent upon the way it organizes files on storage devices.

File Design Information systems in business are file and database oriented. Data are accumulated into files that are processed or maintained by the system. The systems analyst is responsible for designing files, determining their contents and selecting a method for organizing the data.

The most important purpose of a file system is to manage user data. This includes storing, retrieving and updating data. Some file systems accept data for storage as a stream of bytes which are collected and stored in a manner efficient for the media.

3.2 INPUT DESIGN

The input design is the link between the information system and the user. It comprises the developing specification and procedures for data preparation and those steps are necessary to put transaction data in to a usable form for processing can be achieved by inspecting the computer to read data from a written or printed document or it can occur by having people keying the data directly into the system. The design of input focuses on controlling the amount of input required, controlling the errors, avoiding delay, avoiding extra steps and keeping the process simple. The input is designed in such a way so that it provides security and ease of use with retaining the privacy. Input Design considered the following things:'

- What data should be given as input?
- How the data should be arranged or coded?

☒ The dialog to guide the operating personnel in providing input.

☒ Methods for preparing input validations and steps to follow when error occur.

OBJECTIVES

Input Design is the process of converting a user-oriented description of the input into a computer-based system. This design is important to avoid errors in the data input process and show the correct direction to the management for getting correct information from the computerized system.

It is achieved by creating user-friendly screens for the data entry to handle large volume of data. The goal of designing input is to make data entry easier and to be free from errors. The data entry screen is designed in such a way that all the data manipulates can be performed. It also provides record viewing facilities.

When the data is entered it will check for its validity. Data can be entered with the help of screens. Appropriate messages are provided as when needed so that the user

will not be in maize of instant. Thus the objective of input design is to create an input layout that is easy to follow

3.3 OUTPUT DESIGN

The design of output is the most important task of any system. During output design, developers identify the type of outputs needed, and consider the necessary output controls and prototype report layouts.

External Outputs

Manufacturers create and design external outputs for printers. External outputs enable the system to leave the trigger actions on the part of their recipients or confirm

actions to their recipients.

Some of the external outputs are designed as turnaround outputs, which are implemented as a form and re-enter the system as an input.

Internal outputs

Internal outputs are present inside the system, and used by end-users and managers. They support the management in decision making and reporting.

Output Integrity Controls

Output integrity controls include routing codes to identify the receiving system, and verification messages to confirm successful receipt of messages that are handled by network protocol.

Printed or screen-format reports should include a date/time for report printing and the data. Multipage reports contain report title or description, and pagination. Pre-printed forms usually include a version number and effective date.

3.4 DATABASE DESIGN

Today's businesses depend on their databases to provide information essential for day-to-day operations, especially in case of electronic commerce businesses who has a definite advantage with up-to-date database access. Good design forms the foundation of any database, and experienced hands are required in the automation process to design for optimum and stable performance.

Software Solutions have been constantly working on these platforms and have attained a level of expertise. We apply proven methodologies to design, develop, integrate and implement database systems to attain its optimum level of performance and maximize security to meet the client's business model.

Business needs addressed:

- Determine the basic objects about which the information is stored

- Determine the relationships between these groups of information and the objects

- Effectively manage data and create intelligent information

- Remote database administration or on site administrative support

- Database creation, management, and maintenance

- Information retrieval efficiency, remove data redundancy and ensure data security

The most important consideration in designing the database is how the information will be used. The main objective of designing a database is Data Integration, Data Integrity and Data Independence.

Data Integration

In a database, information from several files is coordinated, accessed and operated upon as through it is in a single file. Logically, the information is centralized, physically; the data may be located on different devices, connected through data communication facilities.

Data Integrity

Data integrity means storing all data in one place only and how each application accesses it. This approach results in more consistent information, one update being sufficient to achieve a new record status for all applications. This leads to less data redundancy that is data items need not be duplicated.

Data Independence

Data in dependence is the insulation of application programs from changing aspects of physical data organization. This objective seeks to allow changes in the content and organization of physical data without reprogramming of application and allow modifications to application programs without reorganizing the physical data.

3.5 SYSTEM DEVELOPMENT

Systems development is the process of defining, designing, testing, and implementing a new software application or program. It could include the internal development of customized systems, the creation of database systems, or the acquisition of third party developed software.

Systems development life cycle phases include planning, system analysis, system design, development, implementation, integration and testing, and operations and   maintenance.

3.5.1 DESCRIPTION OF MODULES

Donor Registration

This module will be helps to handle for collecting the donor's details, which is manually, collect the donor's details from the donors. The donors give their details like mobile number, email id, address details and current address etc. which is may be used to contact the donors at the emergency time.

Find donors

This module is a main module of this project, it could be finding the donors by as a blood group wise. When the user needs a blood, they just give a blood group name it will automatically find and give people list who are all having a same blood.

Find nearby donors

Basically, this module helps to collect the real time user location from the donors. When the emergency people need a blood from the nearby donors, it may give an accurate nearby donor.

Blood Request

This module handles the request and response for the all the blood request, this can manage the relationship between blood donors.

Manage Donor's Request

This module having lot of concept, it will be managing a sender and receiver donors communicate request. A sender sends a request to the donors a receiver can accept or reject the request. This is a basic concept of this module.

CHAPTER 4

TESTING AND IMPLEMENTATION

TESTING METHODOLOGIES

System testing is state of implementation, which is aimed at ensuring that the system works accurately and efficiently as expect before live operation commences. It certifies that the whole set of programs hang together.

System testing requires a test plan that consists of several key activities and step for run program, string, system and user acceptance testing. The implementation of newly designed package is important in adopting a successful new system

Testing is the important stage in software development. the system test in implementation stage in software development process. The system testing implementation should be confirmation that all is correct and an opportunity to show the users that the system works as expected. It accounts the largest percentage of technical effort in the software development process.

Testing phase in the development cycle validates the code against the functional specification testing is vital to achievement of the system goals. The objective of the testing is to discover errors to fulfill this objective a series of test step unit, integration. Validation and system tests were planned and executed the

test steps are:

SYSTEM TESTING

Testing is an integral part of any system development life cycle. Insufficient and untested applications may tend to crash and the result is loss of economic and manpower investment besides user's dissatisfaction and downfall of reputation. Software testing can be looked upon as one among many processes, an organization performs, and that provides the lost opportunity to correct any flaws in the developed system. Software testing includes selecting test data that have more probability of giving errors.

The first step in system testing is to develop a plan that tests all aspects of the system. Completeness, correctness, reliability and maintainability of the software are to be tested for the best quality assurance that the system meets the specification and requirements for its intended use and performance. System testing is the most useful practical process of executing a program with the implicit intention of finding errors that make the program fails. System testing is done in three phases.

- Unit Testing
- Integration Testing
- Validation Testing

UNIT TESTING

Unit testing focuses verification effort on the smallest unit of software the module. Using the detailed design and the process specification testing is done to registration by the user with in the boundary of the Login module. The login form

receives the username and password details and validates the value with the database. If valid, the home page is displayed.

INTEGRATION TESTIN

Integration Testing is the process of this activity can be considered as testing the design and hence module interaction. The primary objective of integration testing is to discover errors in the interfaces between the components. Login form and registration form are integrated and tested together. If the user is newly registered, the received details will be stored in the registration table. While logging in, the application will check for valid user name and password in the registration table and if valid the user is prompted for submitting complaints.

Data can be lost across an interface, one module can have adverse effect on another sub function when combined it may not produce the desired major functions. Integration testing is a systematic testing for constructing test to uncover errors associated within an interface.

The objectives taken from unit tested modules and a program structure is built for integrated testing. All the modules are combined and the test is made.

A correction made in this testing is difficult because the vast expenses of the entire program complicated the isolation of causes. In this integration testing step, all the errors are corrected for next testing process.

VALIDATION TESTIN

Validation are independent procedures that are used together for checking that a product, service, or system meets requirements and specifications and that it fulfills its in purpose the actual result from the expected result for the complaint process. Select the complaint category of the complaint by user. The input given to various forms fields are validated effectively. Each module is tested independently. It is tested that

the complaint module fields receive the correct input for the necessary details such as complaint category, complaint id, reference name, complaint description, and email for further process.s

After the completion of the integrated testing, software is completely assembled as a package; interfacing error has been uncovered and corrected and a final series of software test validation begins.

Validation testing can be defined in many ways but a simple definition is that validation succeeds when the software function in a manner that can be reasonably expected by the customer. After validation test has been conducted, one of two possible conditions exists.

OUTPUT TESTIN

The next process of validation testing, is output testing of the proposed system, since no system could be successful if it does not produce the required output in the specified format. Asking the user about the format required, list the output to be generated or displayed by the system under considerations.

Output testing is a different test whose primary purpose is to fully exercise the computer based system although each test has a different purpose all the work should verify that all system elements have been properly integrated and perform allocated functions.

The output format on the screen is found to be corrected as the format was designed in the system design phase according to the user needs for the hard copy also; the output testing has not resulted in any correction in the system.

SYSTEM IMPLEMENTATION

When the initial design was done for the system, the client was consulted for the acceptance of the design so that further proceedings of the system development can be carried on. After the development of the system a demonstration was given to them about the working of the system. The aim of the system illustration was to identify any malfunction of the system.

After the management of the system was approved the system implemented in the concern, initially the system was run parallel with existing manual system. The system has been tested with live data and has proved to be error free and user friendly.

Implementation is the process of converting a new or revised system design into an operational one when the initial design was done by the system; a demonstration was given to the end user about the working system.

This process is uses to verify and identify any logical mess working of the system by feeding various combinations of test data. After the approval of the system by both end user and management the system was implemented.

System implementation is made up of many activities. The six major activities are as follows.

CODING

Coding is the process of whereby the physical design specifications created by the analysis team turned into working computer code by the programming team. A design code may be a tool which helps ensure that the aspiration for quality and quantity for customers and their requirements, particularly for large scale projects, sought by the water agency Design pattern are documented tried and tested solutions for recurring problems in a given context. So basically you have a problem context and the proposed solution for the same.

INSTALLATION

Installation is the process during which the current system is replaced by the new system. This includes conversion of existing data, software, and documentation and work procedures to those consistent with the new system.

DOCUMENTATION

Documentation is descriptive information that describes the use and operation of the system. The user guide is provided to the end user as the student and administrator. The documentation part contains the details as follows,

User requirement and water agency details administration has been made online. Any customer can request their water requirement details through online and also use of documentation, they can view the purpose of each purpose, The admin could verify the authentication of the users, users requirements and need to take delivery process, thus the documentation is made of full view of project thus it gives the guideline to study the project and how to execute also.

USER TRAINING AND SUPPORT

The software is installed at the deployment environment, the developer will give training to the end user of the regional transport officer and police admin officer in that software. The goal of an end user training program is to produce a motivated user who has the skills needed to apply what has been to apply what has been learned to perform the job related task. The following are the instruction which is specified the handling and un-handling events in the application,

The authenticated user of admin and office workers only login in the application with authorized username and password.

Don't make user waste their time to come straight to the water agency or make a phone call.

It can easily track through online by the user.

Very user friendliness software

IMPLEMENTATION PROCEDURES

Implementation includes all the activities that take place to convert the old system to the new one. Proper implementation is essential to provide a reliable system to meet the organization requirements. Implementation is the stage in the project where the theoretical design is turned into a working system. The most crucial stage is achieving a successful new system & giving the user confidence in that the new system will work efficiently & effectively in the implementation state.

IMPLEMENTATION PROCEDURES

PILOT RUNNING

Processing the current data by only one user at a time called the pilot running process. When one user is accessing the data at one system, the system is sets to be engaged and connected in network. This process is useful only in system where more than one user is restricted.

PARALLEL RUNNING:

Processing the current data by more than one user at a time simultaneously is said to be parallel running process. This same system can be viewed and accessed by more than one user at the time. Hence the implementation method used in the system is a pilot type of implementation.

Implementation is the stage in the project where the theoretical design is turned into a working system. The most crucial stage is achieving a successful new system & giving the user confidence in that the new system will work efficiently & effectively in the implementation state.

The stage consists of,

- Testing the developed program with sample data.

- Detection's and correction of error.

- Creating whether the system meets user requirements.

- Making necessary changes as desired by the user.

- Training user personnel.

USER TRAINING

User Training is designed to prepare the user for testing &consenting the system. .

- User Manual.
- Help Screens.
- Training Demonstration.

USER MANUAL

The summary of important functions about the system and software can be provided as a document to the user.

HELP SCREENS

This features now available in every software package, especially when it is used with a menu. The user selects the " Help" option from the menu. The system accesses the necessary description or information for user reference.

TRAINING DEMONSTRATION:

Another User Training element is a Training Demonstration. Live demonstrations with personal contact are extremely effective for Training Users.

SYSTEM MAINTENANCE

Maintenance is actually the implementation of the review plan. As important as it is, many programmers and analysts are to perform or identify themselves with the maintenance effort. There are psychological, personality and professional reasons for this. Analysts and programmers spend far more time maintaining programs than they do writing them. Maintenance accounts for 50-80 percent of total system development

Maintenance is expensive. One way to reduce the maintenance costs are through maintenance management and software modification audits.

Maintenance is not as rewarding as exciting as developing systems. It is perceived as requiring neither skill not experience.

Users are not fully cognizant of the maintenance problem or its high cost.

Few tools and techniques are available for maintenance.

A good test plan is lacking.

Standards, procedures, and guidelines are poorly defined and enforced.

Programs are often maintained without care for structure and documentation.

There are minimal standards for maintenance.

Programmers expect that they will not be in their current commitment by time their programs go into the maintenance cycle.

Corrective Maintenance

It means repairing, processing or performance failure or making changes because of previously uncovered problems or false assumptions. Task performed to identify, isolate, and rectify a fault so that the failed equipment, machine, or system can be restored to an operational condition within the tolerances or limits established for in-service operations.

Corrective maintenance can be subdivided into "immediate corrective maintenance" (in which work starts immediately after a failure) and "deferred corrective maintenance" (in which work is delayed in conformance to a given set of maintenance rules).

Perfective Maintenance

It means changes made to a system to add new features or to improve performance. Preventive maintenance is predetermined work performed to a schedule with the aim of preventing the wear and tear or sudden failure of equipment components. process or control equipment failure can have adverse results in both human and economic terms. In addition to down time and the costs involved to repair and/or replace equipment parts or components, there is the risk of injury to operators, and of acute exposures to chemical and/or physical agents.

Time-based or run-based Periodically inspecting, servicing, cleaning, or replacing parts to prevent sudden failure .On-line monitoring of equipment in order to use important/expensive parts to the limit of their serviceable life. Preventive maintenance involves changes made to a system to reduce the chance of future system failure.

An example of preventive maintenance might be to increase the number of records that a system can process far beyond what is currently needed or to generalize how a system sends report information to a printer so that so that the system can adapt to changes in printer technology.

Preventive Maintenance

Changes made to a system to avoid possible future problems Perfective maintenance involves making enhancements to improve processing performance, interface usability, or to add desired, but not necessarily required, system features. The objective of perfective maintenance is to improve response time, system efficiency, reliability, or maintainability.

During system operation, changes in user activity or data pattern can cause a decline in efficiency, and perfective maintenance might be needed to restore performance. Usually, the perfective maintenance work is initiated by the IT department, while the corrective and adaptive maintenance work is normally requested   by   users.

CHAPTER 5

CONCLUSION

Blood banks play a critical role in ensuring that patients in need of blood transfusions have access to safe and reliable blood. The importance of blood banks cannot be overstated as they provide a lifeline for patients who require blood transfusions due to various medical conditions, surgical procedures, and accidents.

The success of a blood bank relies on several factors, including the availability of a sufficient and diverse blood supply, effective screening and testing of blood donors, appropriate storage and distribution of blood products, and adherence to strict safety protocols.

Blood banks are not only responsible for collecting and distributing blood products but also for educating the public about the importance of blood donation and increasing awareness about the need for regular blood donations.

In conclusion, blood banks are vital institutions that contribute significantly to the healthcare system by providing a safe and reliable supply of blood products. They play a critical role in saving lives and improving the quality of life for patients who require blood transfusions. It is essential to support blood banks and promote regular blood donation to ensure that they can continue to provide these life-saving services.

BIBLIOGRAPHY

Book Reference:

- Bloch, Joshua. Effective Java: Programming Language Guide. Addison-Wesley, 2017.
- Eckel, Bruce. Thinking in Java. Prentice Hall, 2006.
- Freeman, Eric, and Elisabeth Robson. Head First Java. O'Reilly Media, 2005.

- Horstmann, Cay S. Core Java Volume I--Fundamentals. Prentice Hall, 2018.
- Lea, Doug. Concurrent Programming in Java: Design Principles and Patterns. Addison-Wesley, 2000.

Website:

- Baeldung. "Java Tutorials and Articles." Baeldung, 2023, https://www.baeldung.com/java-tutorials.
- GeeksforGeeks. "Java Programming Language." GeeksforGeeks, 2023, https://www.geeksforgeeks.org/java/.
- Stack Overflow. "Questions tagged [java]." Stack Overflow, https://stackoverflow.com/questions/tagged/java.
- Tutorials Point. "Java Tutorial." Tutorials Point, 2023, https://www.tutorialspoint.com/java/index.htm.
- Vogella. "Java Tutorials." Vogella, 2023, https://www.vogella.com/tutorials/java.html.

APPENDICES

A. DATA FLOW DIAGRAM

A data-flow diagram (DFD)is a way of representing a flow of a data of a process or system. The DFD also provides information about the outputs and inputs of each entity and process itself. A data-flow diagram is a part of structured-analysis modeling tools.

LEVEL 0:

LEVEL 1:

```
┌─────────────┐              ╭──────────╮              ┌─┬─────────┬─┐
│             │              │  Donor   │              │ │         │ │
│    Donor    │─────────────▶│Registration│───────────▶│ │  Donor  │ │
│             │              │          │              │ │         │ │
└─────────────┘              ╰──────────╯              └─┴─────────┴─┘
                                  │                          │
                                  ▼                          │
                             ╭──────────╮                    │
                             │          │◀───────────────────┘
                             │Find Donors│
                             │          │
                             ╰──────────╯
                                  │
                                  ▼
                             ╭──────────╮              ┌─┬─────────┬─┐
                             │  Nearby  │◀─────────────│ │Location │ │
                             │  Donors  │              │ │         │ │
                             ╰──────────╯              └─┴─────────┴─┘
                                  │
                                  ▼
                             ╭──────────╮              ┌─┬─────────┬─┐
                             │  Blood   │─────────────▶│ │ Request │ │
                             │ Request  │              │ │         │ │
                             ╰──────────╯              └─┴─────────┴─┘
                                  │                          │
                                  ▼                          │
                             ╭──────────╮                    │
                             │  Manage  │◀───────────────────┘
                             │  Donors  │
                             ╰──────────╯
```
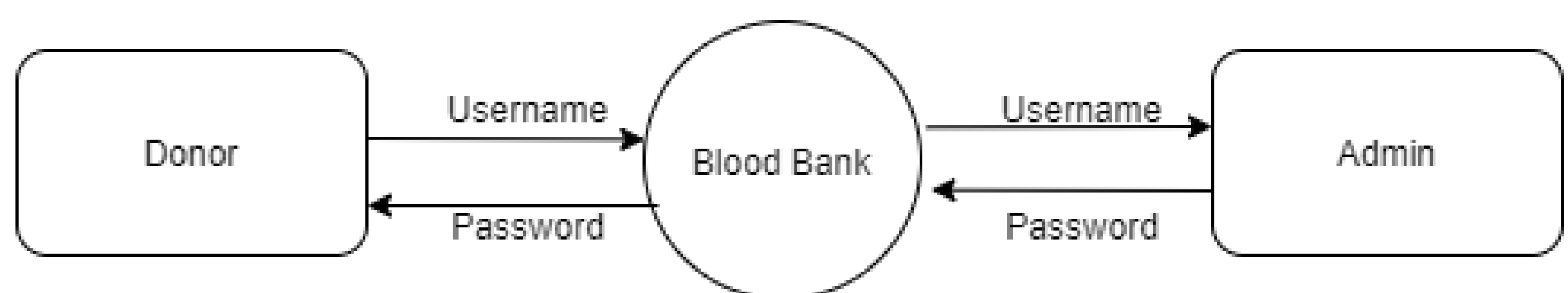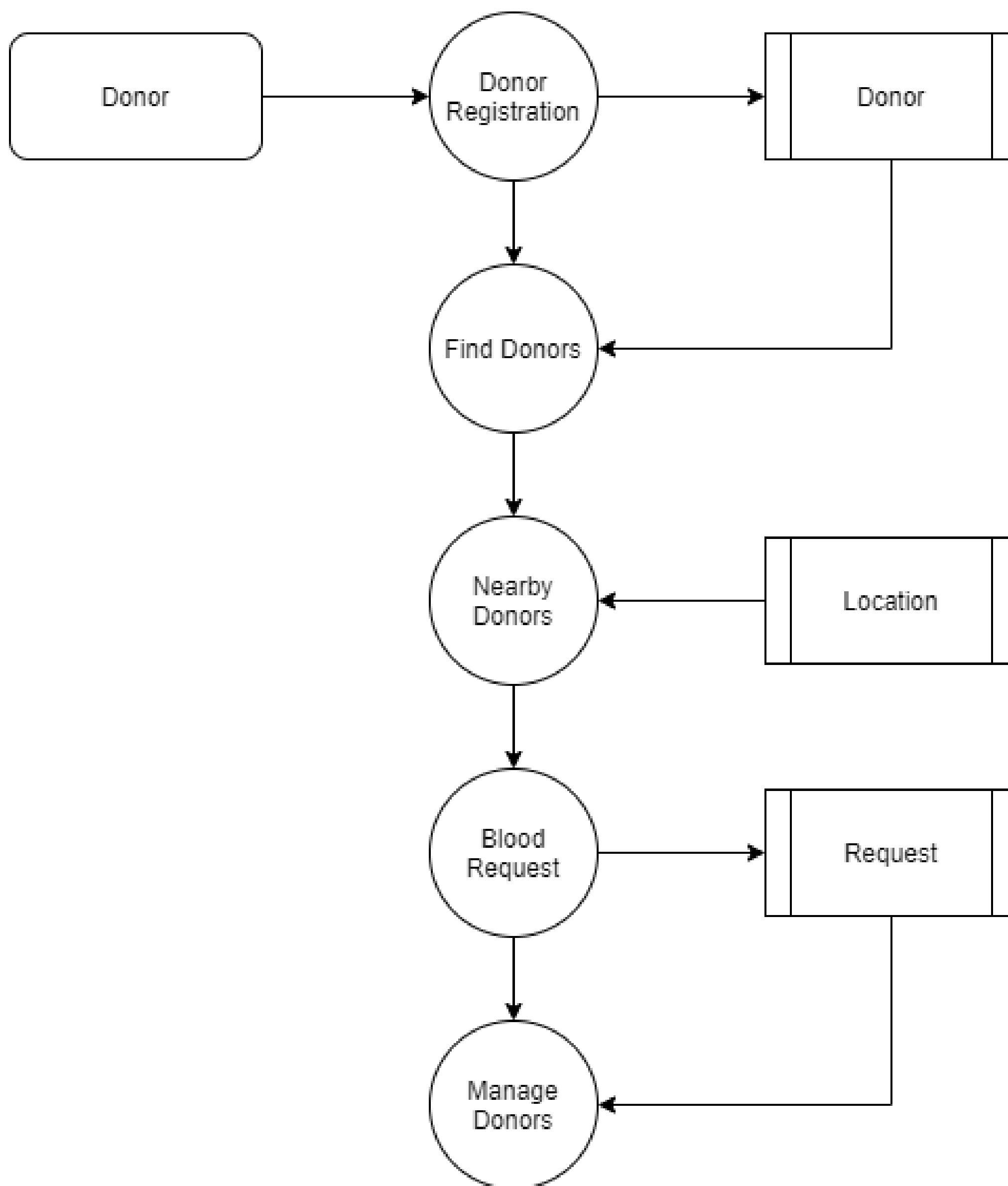
B. **TABLE STRUCTURE**

The table needed for each module was designed and the specification of each and every column was given based on the records and details collected during record specification of the system study.

TABLE NAME: ADMIN

FIELD	DATA TYPE	SIZE	CONSTRAINT
Admin id	Int	10	Primary key
Username	Varchar	20	Not null
password	Varchar	20	Not null

TABLE NAME: DONOR

FIELD	DATA TYPE	SIZE	CONSTRAINT
Donor id	Int	10	Primary key
User type	Varchar	20	Not null
Name	Varchar	20	Not null
Age	Int	10	Not null
Blood group	Varchar	2	Not null
Mobile	Int	10	Not null
Username	Varchar	20	Not null
Password	Varchar	20	Not null
City	Varchar	20	Not null

District	Varchar	20	Not null
Pin code	Int	6	Not null
Address	Varchar	20	Not null

TABLE NAME: REQUEST

FIELD	DATA TYPE	SIZE	CONSTRAINT
Request id	Int	10	Primary key
Sender id	Int	10	Foreign key
Receiver id	Int	10	Foreign key
Sending status	Varchar	5	Not null
Receiving status	Varchar	5	Not null
date	Date	10	Not null

TABLE NAME: LOCATION

FIELD	DATA TYPE	SIZE	CONSTRAINT
Location id	Int	10	Primary key
Donor id	Int	10	Foreign key
Latitude	Varchar	10	Not null
Longitude	Varchar	10	Not null
Last location	Varchar	10	Not null
Current location	Varchar	10	Not null

C. SAMPLE CODEING

```java
package bloodbank.com.bb;


import android.support.design.widget.TextInputEditText;

import android.support.v7.app.AppCompatActivity;

import android.os.Bundle;

import android.view.View;

import android.widget.AdapterView;

import android.widget.ArrayAdapter;

import android.widget.Button;

import android.widget.Spinner;

import android.widget.Toast;


import com.android.volley.Request;

import com.android.volley.RequestQueue;

import com.android.volley.Response;

import com.android.volley.VolleyError;

import com.android.volley.toolbox.StringRequest;
```

```java
import com.android.volley.toolbox.Volley;

import org.json.JSONException;
import org.json.JSONObject;

import java.util.ArrayList;
import java.util.HashMap;
import java.util.List;
import java.util.Map;

public class RegistrationActivity extends AppCompatActivity implements
AdapterView.OnItemSelectedListener {
    Spinner spinner;
    TextInputEditText name, age, mobile, username, password, confirmpassword, city, district,
pincode, fulladdress;
    Button register;
    List<String> categories = new ArrayList<String>();
    URL url;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_registration);
        name = (TextInputEditText) findViewById(R.id.name);
        age = (TextInputEditText) findViewById(R.id.age);
        mobile = (TextInputEditText) findViewById(R.id.mobile);
        username = (TextInputEditText) findViewById(R.id.username);
        password = (TextInputEditText) findViewById(R.id.password);
        confirmpassword = (TextInputEditText) findViewById(R.id.confirmpassword);
        city = (TextInputEditText) findViewById(R.id.city);
        district = (TextInputEditText) findViewById(R.id.district);
        pincode = (TextInputEditText) findViewById(R.id.pincode);
        fulladdress = (TextInputEditText) findViewById(R.id.address);
```

```java
        spinner = (Spinner) findViewById(R.id.bloodgroup);
        register = (Button) findViewById(R.id.register);


        spinner.setOnItemSelectedListener(this);


        categories.add("A-");
        categories.add("A+");
        categories.add("B-");
        categories.add("B+");
        categories.add("O-");
        categories.add("O+");
        categories.add("AB+");
        categories.add("AB-");
                        ArrayAdapter<String>  dataAdapter  =  new  ArrayAdapter<String>(this,
android.R.layout.simple_spinner_item, categories);


dataAdapter.setDropDownViewResource(android.R.layout.simple_spinner_dropdown_item);
        spinner.setAdapter(dataAdapter);


        register.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                checkValidation();
            }
        });
    }


    private void checkValidation() {
        final String _name = name.getText().toString().trim();
        final String _age = age.getText().toString().trim();
        final String _mobile = mobile.getText().toString().trim();
        final String _username = username.getText().toString().trim();
        final String _password = password.getText().toString().trim();
```

```java
String _confirmpassword = confirmpassword.getText().toString().trim();

final String _city = city.getText().toString().trim();

final String _district = district.getText().toString().trim();

final String _pincode = pincode.getText().toString().trim();

final String _fulladdress = fulladdress.getText().toString().trim();

int _bg = spinner.getSelectedItemPosition();

final String _bloodgroup = categories.get(_bg).trim();


    if (_name.length() == 0 || _age.length() == 0 || _mobile.length() == 0 || _username.length()
== 0 || _password.length() == 0 || _confirmpassword.length() == 0 || _city.length() == 0 ||
_district.length() == 0 || _pincode.length() == 0 || _fulladdress.length() == 0) {
                          Toast.makeText(getApplicationContext(), "Enter All Fields values",
Toast.LENGTH_SHORT).show();
    } else {
       if (_mobile.length() < 10) {
                             Toast.makeText(getApplicationContext(), "Invalid Mobile number",
Toast.LENGTH_SHORT).show();
       } else if (_pincode.length() < 6) {
                             Toast.makeText(getApplicationContext(), "Invalid Pincode Number",
Toast.LENGTH_SHORT).show();
       } else if (Integer.parseInt(_age) <= 0) {
                                Toast.makeText(getApplicationContext(), "Invalid Age",
Toast.LENGTH_SHORT).show();
       } else if (!_password.equals(_confirmpassword)) {
                           Toast.makeText(getApplicationContext(), "Password does not match",
Toast.LENGTH_SHORT).show();
       } else {
          RequestQueue requestQueue = Volley.newRequestQueue(this);
          StringRequest stringRequest = new StringRequest(Request.Method.POST, url.register,
new Response.Listener<String>() {
             @Override
             public void onResponse(String response) {
                try {
```

```java
                    JSONObject jsonObject1 = new JSONObject(response);

                    JSONObject jsonObject11=jsonObject1.getJSONObject("data");

                    if(jsonObject11.getInt("status")==1) {

                                            Toast.makeText(getApplicationContext(),
String.valueOf(jsonObject11.getString("msg")),Toast.LENGTH_LONG).show();

                        name.setText("");

                        age.setText("");

                        mobile.setText("");

                        username.setText("");

                        password.setText("");

                        confirmpassword.setText("");

                        city.setText("");

                        district.setText("");

                        pincode.setText("");

                        fulladdress.setText("");


                    }else if(jsonObject11.getInt("status")==0){

                                            Toast.makeText(getApplicationContext(),
String.valueOf(jsonObject11.getString("msg")),Toast.LENGTH_LONG).show();


                    }
                } catch (JSONException e) {
                    e.printStackTrace();
                }



            }
        }, new Response.ErrorListener() {
            @Override
            public void onErrorResponse(VolleyError error) {
                            Toast.makeText(getApplicationContext(),  String.valueOf(error),
Toast.LENGTH_SHORT).show();
            }
```

```java
        }) {
            @Override
            protected Map<String, String> getParams() {
                Map<String, String> params = new HashMap<String, String>();

                params.put("name", _name);
                params.put("age", _age);
                params.put("bloodgroup", _bloodgroup);
                params.put("mobile", _mobile);
                params.put("username", _username);
                params.put("password", _password);
                params.put("city", _city);
                params.put("district", _district);
                params.put("pincode", _pincode);
                params.put("fulladdress", _fulladdress);


                return params;
            }
        };
        requestQueue.add(stringRequest);
    }


}

@Override
public void onItemSelected(AdapterView<?> adapterView, View view, int i, long l) {

}
```

```java
    @Override
    public void onNothingSelected(AdapterView<?> adapterView) {


    }



}
package bloodbank.com.bb;


/**
 * Created by gokul on 25-01-2018.
 */


import java.io.IOException;

import java.util.List;

import java.util.Locale;


import android.Manifest;

import android.app.AlertDialog;

import android.app.Service;

import android.content.Context;

import android.content.DialogInterface;

import android.content.Intent;

import android.content.pm.PackageManager;

import android.location.Address;

import android.location.Geocoder;

import android.location.Location;

import android.location.LocationListener;

import android.location.LocationManager;

import android.os.Bundle;

import android.os.IBinder;

import android.provider.Settings;

import android.support.v4.app.ActivityCompat;
```

```java
import android.util.Log;


class GPSTracker extends Service implements LocationListener {

    // Get Class Name
    private static String TAG = GPSTracker.class.getName();


    private final Context mContext;


    // flag for GPS Status
    boolean isGPSEnabled = false;


    // flag for network status
    boolean isNetworkEnabled = false;


    // flag for GPS Tracking is enabled
    boolean isGPSTrackingEnabled = false;


    Location location;
    double latitude;
    double longitude;


    // How many Geocoder should return our GPSTracker
    int geocoderMaxResults = 1;


    // The minimum distance to change updates in meters
    private static final long MIN_DISTANCE_CHANGE_FOR_UPDATES = 10; // 10 meters


    // The minimum time between updates in milliseconds
    private static final long MIN_TIME_BW_UPDATES = 1000 * 60 * 1; // 1 minute


    // Declaring a Location Manager
```

```java
protected LocationManager locationManager;

    // Store  LocationManager.GPS_PROVIDER  or  LocationManager.NETWORK_PROVIDER
information
    private String provider_info;

    public GPSTracker(Context context) {
        this.mContext = context;
        getLocation();
    }

    /**
     * Try to get my current location by GPS or Network Provider
     */
    public void getLocation() {

        try {

            locationManager     =      (LocationManager)
mContext.getSystemService(LOCATION_SERVICE);

            //getting GPS status
            isGPSEnabled        =
locationManager.isProviderEnabled(LocationManager.GPS_PROVIDER);

            //getting network status
            isNetworkEnabled        =
locationManager.isProviderEnabled(LocationManager.NETWORK_PROVIDER);

            // Try to get location if you GPS Service is enabled
            if (isGPSEnabled) {
                this.isGPSTrackingEnabled = true;

                Log.d(TAG, "Application use GPS Service");
```

```java
        /*
         * This provider determines location using
         * satellites. Depending on conditions, this provider may take a while to return
         * a location fix.
         */

        provider_info = LocationManager.GPS_PROVIDER;

    } else if (isNetworkEnabled) { // Try to get location if you Network Service is enabled
        this.isGPSTrackingEnabled = true;

        Log.d(TAG, "Application use Network State to get GPS coordinates");

        /*
         * This provider determines location based on
         * availability of cell tower and WiFi access points. Results are retrieved
         * by means of a network lookup.
         */
        provider_info = LocationManager.NETWORK_PROVIDER;

    }

    // Application can use GPS or Network Provider
    if (!provider_info.isEmpty()) {
                                        if    (ActivityCompat.checkSelfPermission(this,
Manifest.permission.ACCESS_FINE_LOCATION) != PackageManager.PERMISSION_GRANTED
&&                                            ActivityCompat.checkSelfPermission(this,
Manifest.permission.ACCESS_COARSE_LOCATION)                                      !=
PackageManager.PERMISSION_GRANTED) {
            // TODO: Consider calling
            //    ActivityCompat#requestPermissions
            // here to request the missing permissions, and then overriding
```

```
//   public void onRequestPermissionsResult(int requestCode, String[] permissions,
//                              int[] grantResults)
// to handle the case where the user grants the permission. See the documentation
// for ActivityCompat#requestPermissions for more details.
        return;
    }
    locationManager.requestLocationUpdates(
        provider_info,
        MIN_TIME_BW_UPDATES,
        MIN_DISTANCE_CHANGE_FOR_UPDATES,
        this
    );


    if (locationManager != null) {
        location = locationManager.getLastKnownLocation(provider_info);
        updateGPSCoordinates();
    }
  }
}
catch (Exception e)
{
    //e.printStackTrace();
    Log.e(TAG, "Impossible to connect to LocationManager", e);
}
}


/**
 * Update GPSTracker latitude and longitude
 */
public void updateGPSCoordinates() {
    if (location != null) {
        latitude = location.getLatitude();
        longitude = location.getLongitude();
```

```java
    }
}


/**
 * GPSTracker latitude getter and setter
 * @return latitude
 */
public double getLatitude() {
    if (location != null) {
        latitude = location.getLatitude();
    }

    return latitude;
}


/**
 * GPSTracker longitude getter and setter
 * @return
 */
public double getLongitude() {
    if (location != null) {
        longitude = location.getLongitude();
    }

    return longitude;
}


/**
 * GPSTracker isGPSTrackingEnabled getter.
 * Check GPS/wifi is enabled
 */
public boolean getIsGPSTrackingEnabled() {
```

```java
        return this.isGPSTrackingEnabled;
    }


    /**
     * Stop using GPS listener
     * Calling this method will stop using GPS in your app
     */
    public void stopUsingGPS() {
        if (locationManager != null) {
            locationManager.removeUpdates(GPSTracker.this);

        }
    }


    /**
     * Function to show settings alert dialog
     */
    public void showSettingsAlert() {
        AlertDialog.Builder alertDialog = new AlertDialog.Builder(mContext);


        //Setting Dialog Title
        alertDialog.setTitle("GPS TITLE");


        //Setting Dialog Message
        alertDialog.setMessage("Alert");


        //On Pressing Setting button
        alertDialog.setPositiveButton("OK", new DialogInterface.OnClickListener() {


            @Override
            public void onClick(DialogInterface dialog, int which)
            {
                Intent intent = new Intent(Settings.ACTION_LOCATION_SOURCE_SETTINGS);
                mContext.startActivity(intent);
```

```java
            }
        });


        //On pressing cancel button
        alertDialog.setNegativeButton("Cancel", new DialogInterface.OnClickListener() {


            @Override
            public void onClick(DialogInterface dialog, int which)
            {
                dialog.cancel();
            }
        });


        alertDialog.show();
    }


    /**
     * Get list of address by latitude and longitude
     * @return null or List<Address>
     */
    public List<Address> getGeocoderAddress(Context context) {
        if (location != null) {


            Geocoder geocoder = new Geocoder(context, Locale.ENGLISH);


            try {
                /**
                 * Geocoder.getFromLocation - Returns an array of Addresses
                 * that are known to describe the area immediately surrounding the given latitude and
longitude.
                 */
                List<Address> addresses = geocoder.getFromLocation(latitude, longitude,
this.geocoderMaxResults);
```

```java
            return addresses;
        } catch (IOException e) {
            //e.printStackTrace();
            Log.e(TAG, "Impossible to connect to Geocoder", e);
        }
    }


    return null;
}


/**
 * Try to get AddressLine
 * @return null or addressLine
 */
public String getAddressLine(Context context) {
    List<Address> addresses = getGeocoderAddress(context);


    if (addresses != null && addresses.size() > 0) {
        Address address = addresses.get(0);
        String addressLine = address.getAddressLine(0);


        return addressLine;
    } else {
        return null;
    }
}


/**
 * Try to get Locality
 * @return null or locality
 */
public String getLocality(Context context) {
```

```java
        List<Address> addresses = getGeocoderAddress(context);

        if (addresses != null && addresses.size() > 0) {
            Address address = addresses.get(0);
            String locality = address.getLocality();

            return locality;
        }
        else {
            return null;
        }
    }

    /**
     * Try to get Postal Code
     * @return null or postalCode
     */
    public String getPostalCode(Context context) {
        List<Address> addresses = getGeocoderAddress(context);

        if (addresses != null && addresses.size() > 0) {
            Address address = addresses.get(0);
            String postalCode = address.getPostalCode();

            return postalCode;
        } else {
            return null;
        }
    }

    /**
     * Try to get CountryName
     * @return null or postalCode
```

```java
 */
public String getCountryName(Context context) {
    List<Address> addresses = getGeocoderAddress(context);
    if (addresses != null && addresses.size() > 0) {
        Address address = addresses.get(0);
        String countryName = address.getCountryName();


        return countryName;
    } else {
        return null;
    }
}


@Override
public void onLocationChanged(Location location) {
}


@Override
public void onStatusChanged(String provider, int status, Bundle extras) {
}


@Override
public void onProviderEnabled(String provider) {
}


@Override
public void onProviderDisabled(String provider) {
}


@Override
public IBinder onBind(Intent intent) {
    return null;
}
```

}

D. SAMPLE INPUT & OUTPUT DESIGN