

TREES
A Project Report Submitted
In Partial Fulfilment of the Requirements
For the Degree of
MASTER OF COMPUTER APPLICATION

By
DIVYANSHI GOYAL
University Roll No. 1900290149043

Under the Supervision of
Mr. Ankit Verma
Assistant Professor
KIET Group of Institutions



Submitted to
DEPARTMENT OF COMPUTER APPLICATION
Affiliated to
DR. A. P. J ABDUL KALAM TECHNICAL UNIVERSITY,
LUCKNOW
JULY,2021

DECLARATION

I hereby declare that the work presented in this report entitled “**TREES**”, was carried out by US. I have not submitted the matter embodied in this report for the award of any other degree or diploma of any other University or Institute.

I have given due credit to the original authors/sources for all the words, ideas, diagrams, graphics, computer programs, experiments, results, that are not my original contribution.

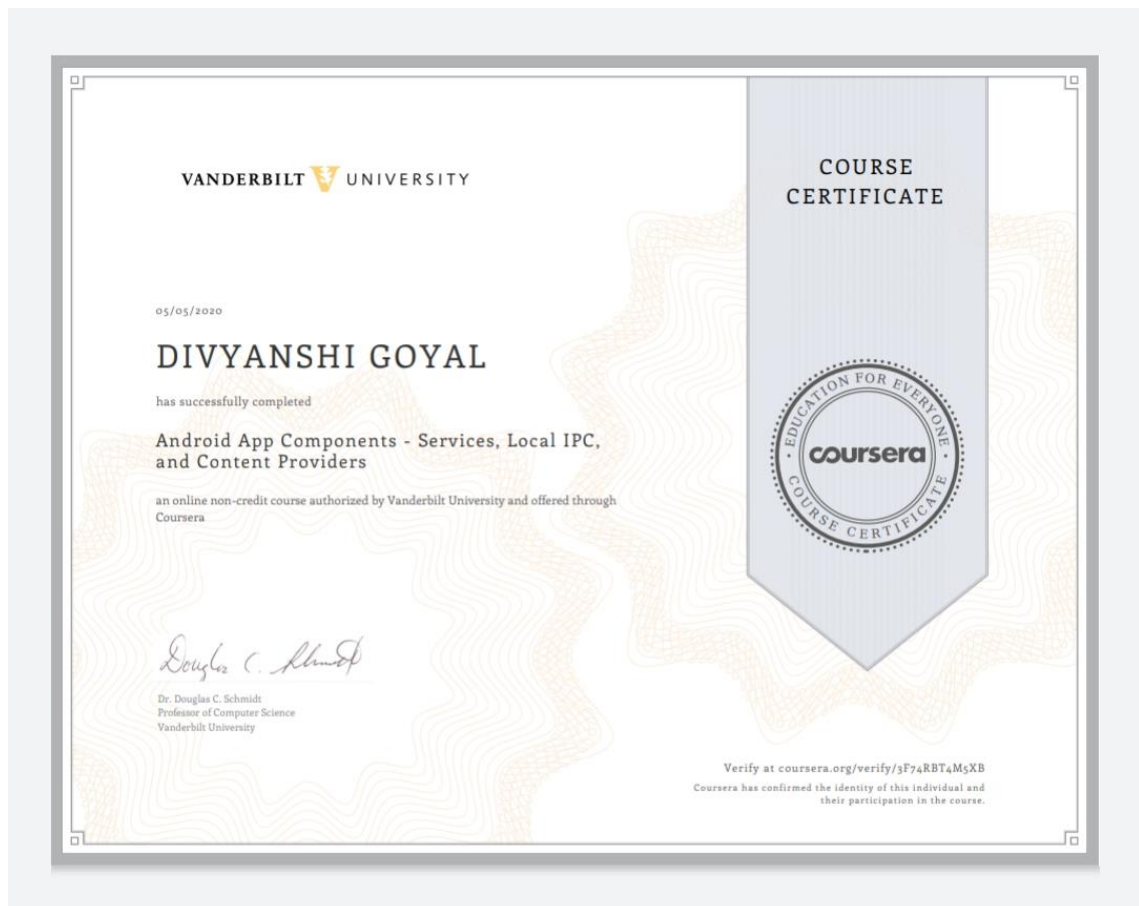
I have used quotation marks to identify verbatim sentences and given credit to the original authors/sources.

I affirm that no portion of my work is plagiarized, and the experiments and results reported in the report are not manipulated. In the event of a complaint of plagiarism and the manipulation of the experiments and results, We shall be fully responsible and answerable.

DIVYANSHI GOYAL

University Roll No.-1900290149043

TRAINING CERTIFICATE



CERTIFICATE

Certified that **DIVYANSHI GOYAL (Univ. Roll No.-1900290149043)** have carried out the project work having “**Trees**” for Master of Computer Application from Dr.A.P.J.AbdulKalam Technical University (AKTU), Technical University, Lucknow under my supervision. The project report embodies original work, and studies are carried out by the student himself/herself and the contents of the project report do not form the basis for the award of any other degree to the candidate or to anybody else from this or any other University/Institution.

Date:

DivyanshiGoyal (Univ. Roll No -1900290149043)

This is to certify that the above statement made by the candidate is correct to the best of our knowledge.

Date:

Mr. Ankit Verma
Assistant Professor
Department of Computer Application
KIET Group of Institutions, Ghaziabad

Signature of External Examiner

Signature of Internal Examiner

Dr. AjayKumarShrivastava
Head, Department of Computer Application
KIET Group of Institutions, Ghaziabad

ABSTRACT

“Trees” is an android application to connect to people who want to plant trees easily. Users who have an android phone & active internet connection can contact and start with their trees planted easily through this application. The purpose of this application is to help people who are unable to care for their plants and do not have space for planting trees.

- Easy accessibility.
- It makes taking care of environment easier.
- User can plant a tree according to their liking for a small fee.
- The user need is not required to have an in-depth knowledge of plants.
- On the whole it liberates the user from taking regular care of plants.
- So this project is about providing convenience.

ACKNOWLEDGEMENTS

Success in life is never attained single handedly. My deepest gratitude goes to my thesis supervisor **Mr. Ankit Verma**, for his guidance, help and encouragement throughout our project work. Their enlightening ideas, comments, and suggestions.

Words are not enough to express my gratitude to **Dr. Ajay Kumar Shrivastava**, Professor and Head, Department of Computer Application, for his insightful comments and administrative help at various occasions.

Fortunately, I have many understanding friends, who have helped me a lot on many critical conditions.

Finally, my sincere thanks go to my family members and all those who have directly and indirectly provided me moral support and other kind of help. Without their support, completion of this work would not have been possible in time. They keep my life filled with enjoyment and happiness.

Divyanshi Goyal

Univ. Roll No. 1900290149043

TABLE OF CONTENTS

Declaration	ii
Certificate	iii
Abstract	iv
Acknowledge	v
Table of content	vi
List of tables	viii
List of figures	viii
Chapter 1: Introduction	9-20
1.1 Project description	9
1.2 Project Scope	11
1.3 Identification of need	12
1.4 Problem Statement	13
1.5 Hardware/software used in project	14
1.6 Project schedule	18
1.6.1 Pert Chart	
1.6.2 Gantt Chart	
Chapter 2: Literature Review	21-26
Chapter 3: Feasibility Study	27-32
3.1 Introduction	27
3.2 Main Aspects	29
3.2.1 Technical feasibility	
3.2.2 Economical Feasibility	
3.2.3 Operational Feasibility	
3.3 Benefits	31
3.4 SRS	32

Chapter 4:Design	34-44
4.1 Introduction	34
4.2 Analysis	35
4.3 SDLC	37
4.4 Soft. Engg. Paradigm	39
4.4.1 Prototype model	
4.5 DFD	42
4.7ERDiagram	44
 Chapter 5: Report	 45-51
5.1 Gist	45
5.2Some Screenshots	46
 Chapter 6: Coding	 52-59
 Chapter 7: Testing	 60-62
7.1 Introduction	60
7.2 Level of testing	61
7.3 Some Important observations	62
7.4 test case result summary	62
 Chapter 8: Conclusion & Future scope	 64
8.1Conclusion	64
8.2 Future Scope	64
 Bibliography	 65

LIST OF FIGURES

Figure 1.1 Pert chart	17
Figure 1.2:Gannt chart for project.....	18
Figure 2.1:Design	23
Figure 2.2:Flow of website.....	23
Figure 2.3:Some screens	24
Figure 4.1:Above image depicting the planning step.....	377
Figure 4.2:Prototype model.....	40
Figure 4.3:Apex Work.....	42
Figure 4.4 Context level	42
Figure 4.5Dfd 1 level	43
Figure 4.6:Uml use case diagram.....	43
Figure 4.7:ER diagram of system.....	444
Figure 5.1 System.....	455
Figure 7.1:Testing pyramid.....	61

CHAPTER 1

INTRODUCTION

1.1 PROJECT DESCRIPTION

The proposed Trees application helps the people wanting to plant trees but incapable of doing so because of a number of reasons by giving them the details of the availability of group of people that are willing to do so on their behalf. The people who want to plant the tree themselves or can simply choose the group to look after their plants and visit the plants whenever they wish to do so. It saves time for the users but also help them give their own contribution in the healing of environment. The people can select from a wide variety of plants that are available for the plantation depending upon the area they live in and the type of soil available. Our application can work 24x7 so user can register themselves at any point of time. People willing to grow plants can join us in our attempt to heal environment . The main benefit of this system is the betterment of environment and improving the daily depreciating Air Quality Index. In an ever depreciating environment where the quality of air we breathe in and the pollution of fresh water reservoir our application is effort to bring some improvements in the health of our habitat because we are healthy only if our surroundings are clean and healthy. Availability of Android Geo-location enables easier way of obtaining data for soil quality. Implementation of Real time notification using One Signal Server.

1.2 PROJECT SCOPE

The following documentation is a project the “Trees”. It describes the shortcomings in our duty towards the environment due to our unhealthy lifestyle and busy schedules. It highlights our responsibility towards the environment and how to fulfil them without changing our schedules at all. The new proposed system is created understanding the current needs and responsibilities and to fulfil them without changing our schedules.

The new system takes into account the various factors while designing a new system. It keeps into the account the what trees can be planted in which terrain. The foremost thing that is taken care of is the need and requirements of the user.

It is basically for those people who either do not have time to take care of plants or do not have enough space for them.

- It is time saving as it doesn't involve manual process for facing difficulties due to difficult and time saving process.
- It is very user friendly.
- User across India come to the portal and register themselves for plantation by giving some details regarding them
- It is eco-friendly as well, as it not only plant trees but also does not involve usage of papers.
- Errors are almost impossible as it requires less human interaction.
- Accuracy in work.
- Easy & fast retrieval of information.
- Decrease the load of the person involve in existing manual system.
- Access to any information individually.
- Work becomes very speedy.
- Easy to update information
- Easy availability.

1.3 IDENTIFICATION OF NEED

User need identification and analysis are concerned with what user needs rather than what he/she wants. Not until the problem has been identified, defined, and evaluated should the analyst think about solutions and whether the problem is worth working. This step intended to help the user and analyst understand the real problem rather than its symptoms. The user or the analyst may identify the need for a candidate system or for enhancement in the existing system.

An analyst is responsible for performing following tasks:

- Studied strength and weakness of the current system.
- Determined “what” must be done to solve the problem.
- Prepared a functional specifications document.

1.4PROBLEM STATEMENT

In the existing system all the work is done manually. This is chance of committing errors and it will take more time to perform or checkout any information. There are so many limitations in the existing system. So the existing system should be automized. If the system is carried over manually, for everything it take more time. So it is difficult to take immediate decisions.

- In the traditional system, if you wish to analyze any record you have to turn pages many time.
- Existing systems are time consuming as it requires too much planning and so much human involvement.
- As it involves much human involvement, the cost of the system automatically gets increased.
- Existing systems require paper use, which isn't good for the environment.
- With too much human involvement, there are high chances of risk as well.
- There is too much of paper work too, which makes the tasks in the existing system, very tedious.

1.5 HARDWARE / SOFTWARE USED IN PROJECT

1.5.1 HARDWARE REQUIREMENT

Hardware	Configuration
Processor	Above 4.4
Ram	4GB
Monitor	Normal

1.5.2 SOFTWARE REQUIREMENT

Software	Configuration
Database	Firebase
Language	Java

1.5.3 SOME REQUIREMENTS

Performance Requirements:

To achieve good performance the following requirements must be satisfied

- **Scalability:** The ease with which a system or component can be modified to fit the problem area.
- **Portability:** The ease with which a system or component can be transferred from one hardware or software environment to another.
- **Security:** It is the ideal state where all information can be communicated across the internet / company secure from unauthorized persons being able to read it and/or manipulate it..
- **Maintainability:** The ease with which a software system or component can be modified to correct faults, improve performance, or other attributes, or adapt to a changed environment.
- **Reliability:** The ability of a system or component to perform its required functions under stated conditions for a specified period of time.
- **Reusability:** The degree to which a software module or other work product can be used in more than one computing program or software system.

Safety Requirements:

In case scenarios where data integrity can be compromised, measures should be taken to ensure that all changes are made before system is shutdown. The user must have a registered account to use all facility of the web application.

1.5.4 OTHER REQUIREMENTS

1.5.4.1 ANDROID

Android is a software platform and operating system for mobile devices. Android is available as open source. It allows developers to write managed code in the Java language, controlling the device via Google-developed Java libraries.

Android SDK was released by Open Handset Alliance in the month of November of the year 2007. Android was actually developed using the kernel of Linux 2.6 and the highlighting features of Android include the relationship (CRM) suite offering applications for small, midsize and enterprise organizations, with a focus on sales and support. Android ships with a set of core applications including an email client, SMS program, calendar, maps, browser, contacts and others. All applications are built using the Java. Each of the applications aims at performing a specific task that it is actually intended to do.

Applications Layer

Android ships with a set of core applications including an email client, SMS program, calendar, maps, browser, contacts and others. All applications are built using the Java. Each of the applications aims at performing a specific task that it is actually intended to do.

Application Framework Layer

The next layer is the application framework. This includes the programs that manage the phone's basic functions like resource allocation, telephone applications, switching between processes or programs and keeping track of the phone's physical location. Application developers have full access to Android's application framework. This allows them to take advantage of Android's processing capabilities and support features when building an Android application. We can think of the application framework as a set of basic tools with which a developer can build much more complex tools.

Libraries Layer

The next layer contains the native libraries of Android. These shared libraries are all written in C or C++, compiled for the particular hardware architecture used by the phone and preinstalled by the phone vendor.

Android Runtime Layer

Android Runtime layer includes Dalvik Virtual Machine (DVM) and a set of core Java libraries. Every Android app gets its own instance of DVM. Dalvik has been written so that a device can run multiple virtual machines efficiently and it executes files with dex (Dalvik Executable Format) extension optimized for minimum memory.

1.5.4.2 Firebase Cloud Messaging (FCM)

Firebase Cloud Messaging (FCM), formerly known as Google Cloud Messaging (GCM), is a cross-platform cloud solution for messages and notifications for Android, iOS, and web applications, which as of 2021 can be used at no cost. Firebase Cloud Messaging allows third-party application developers to send notifications or messages from servers hosted by FCM to users of the platform or end users. The service is provided by Firebase, a subsidiary of Google. On October 21, 2014, Firebase announced it had been acquired by Google for an undisclosed amount. The official Google Cloud Messaging website points to Firebase Cloud Messaging (FCM) as the new version of GCM. Firebase is a mobile platform which supports users in developing mobile and web applications. Firebase Cloud Messaging is one of many products which are part of the Firebase platform. On the platform users can integrate and combine different Firebase features in both web and mobile applications. It is strongly typed object-oriented, on-demand programming language.

- Easy to use
- Data Focused
- Hosted
- Multitenant aware
- Easy to test

1.6 PROJECT SCHEDULE

The objective of software project planning is to provide a framework that enables the manager to make reasonable estimates of resources, costs and schedule. These estimates are made within a limited time frame at the beginning of a software project and should be updated regularly as the project progresses. In addition, estimates should attempt to define “best case” and “worst case” scenarios so that project outcomes can be bounded.

The first activity in software project planning is the determination of software scope. Function and performance allocated to software during system engineering should be assessed to establish a project scope that is ambiguous and understandable at Presidency and technical levels. Software scope describes function, performance, constraints, interfaces and reliability.

During early stages of project planning, a microscopic schedule is developed. This type of schedule identifies all major software engineering activities and the product functions to which they are applied. As the project gets under way, each entry on the macroscopic schedule is refined into detailed schedule. Here specific software tasks are identified and scheduled.

Scheduling has following principles:

1. Compartmentalization: the project must be compartmentalized into a number of manageable activities and tasks.
2. Interdependency: the interdependencies of each compartmentalized activity or tasks must be determined.
3. Time allocation: each task to be scheduled must be allocated some number of work units.
4. Effort validation: every project has a defined number of staff members.
5. Defined responsibilities: every task that is scheduled should be assigned to a specific team member.
6. Defined outcomes: every task that is scheduled should have a defined outcome.

1.6.1 Pert chart

Program evaluation and review technique (pert) is a project scheduling method that is applied to software development.

Pert provide quantitative tool that allow the software planner to-Determine the critical path-the chain of tasks that determines the duration of the project;Establish “most likely” time estimates for individual tasks by applying statistical models; and

Calculate “boundary times” that defines a time “window” for a particular task.

Pert chart(program evolution review technique) for project-

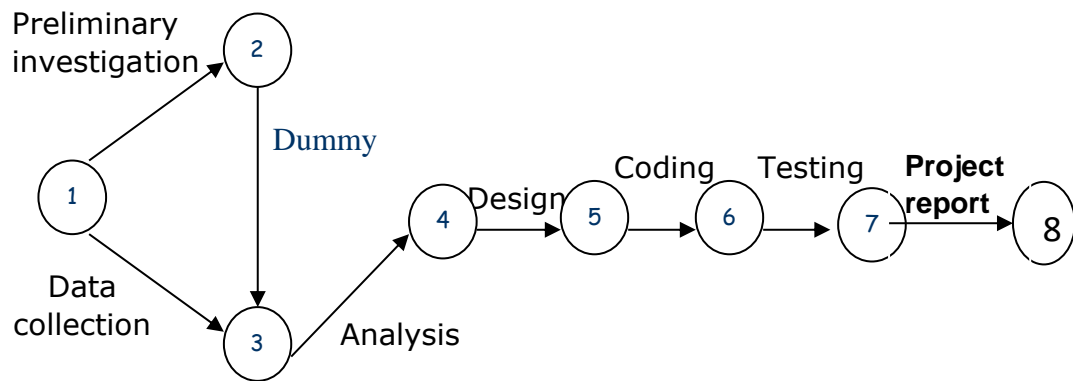


Figure 1.1 Pert chart

1.6.2 Gantt Chart

When creating a project schedule, the planner begins with a set of tasks (the work breakdown structure). If automated tools are used, the work breakdown is input as a task network. Effort, duration and start dates are input are each task network. As a consequence of this input, a timeline chart also called a Gantt chart is generated. A timeline chart is developed for entire project.

Gantt chart for project:

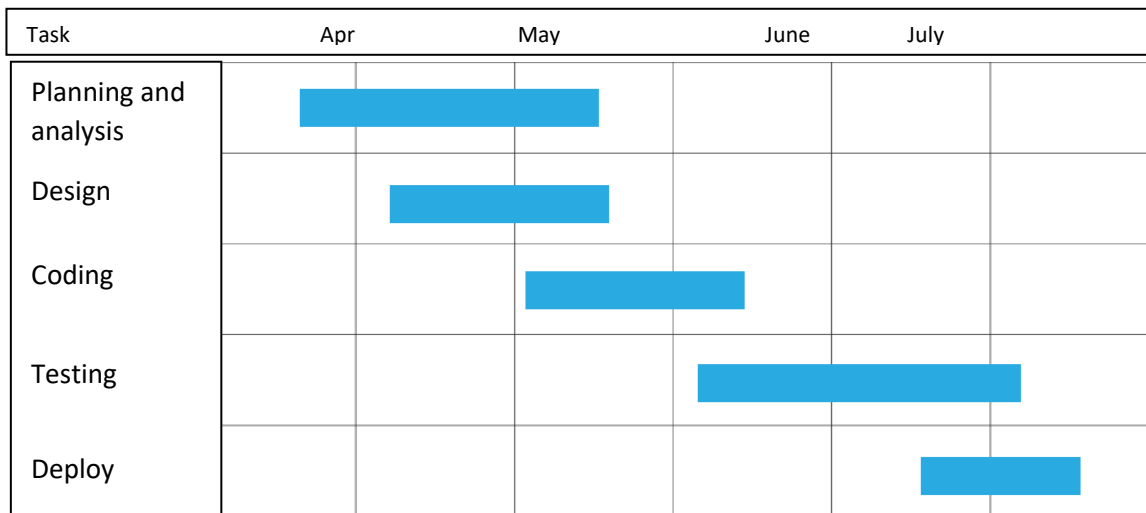


Figure 1.2:Gannt chart for project

Here horizontal bars indicate the duration of each task.

CHAPTER 2

LITERATURE REVIEW

ABSTRACT

Our software is planned for the developing up an easy to use registration and payment Application for tree plantation, that will help to register and pay for the services provided by the administration. Android is one of the most trending technologies, It's a software through which we can do our work with ease and flexibility as well as availability. It helps us building applications with a relative ease and makes them not only easy to use for the users of non-technical background but also makes the application portable as well. Our software mainly. Our application mainly registers the users and ask them to make payments so that the administration can not only plant a tree but also take care of it for them.

Our project is for people who are unable to plant a tree because of lack of space in the high rise buildings or over congestion in the apartments. Our system is for managing, registering, payment, information for the plants etc. For accessing this the user should register themselves, after that they could study the data about the plants that can grow in local terrain. Trees is an attempt to save the environment with like minded people.

KEYWORDS:- Android, Java, Trees, Plants, Administrator, Space, Time, convenience, Lifestyle, Android Studio.

2.1 INTRODUCTION

The Aim of this system is to making an application which allow a normal user who is even not familiar with programming concept can use our system easily[10].

"Trees" is an android application. It can work on android 4.4or above. It is developed using Android studio. It is very easy to use and portable.

Firebase Cloud Messaging is one of many products which are part of the Firebase platform[6]. On the platform users can integrate and combine different Firebase features in both web and mobile applications. It is strongly typed object-oriented, on-demand programming language[15].

Android is a software platform and operating system for mobile devices. Android is available as open source. It allows developers to write managed code in the Java language, controlling the device via Google-developed Java libraries. Android SDK was released by Open Handset Alliance in the month of November of the year 2007[14].

So, the maintenance and maintenance of institute became very easy, it provide:-

- Easy accessibility.

- Easy way to plant trees and have them taken care of.
- It makes searching of data easier and faster.
- Easy availability.
- liberates lengthy manual records.

So our project is providing convenience in different aspects.

2.2 RELATED WORK

2.2.1 College Management System

It's a system which manages the information of college, information of student, information of placement, various event going in college. It keep tracking record of all the data. It has also a notice board which have data about various programme either its cultural or any sports which is supposed to be held soon or technical[5] .

2.2.2 Nature Apps: Waiting for the revolution

Apps are small task-orientated programs with the potential to integrate the computational and sensing capacities of smartphones with the power of cloud computing, social networking, and crowdsourcing. They have the potential to transform how humans interact with nature, cause a step change in the quantity and resolution of biodiversity data, democratize access to environmental knowledge, and reinvigorate ways of enjoying nature. [1].

2.2.3Institute Management System (Ims)

Institute Management System (IMS), a system developed with the help of Java,that could be used for presidency of different things in various firms like colleges, schools and universities. It

have different access level for people of different categories like administrators, special users, normal users. It use SQL for management of database and can be connected so that accessible from any location ,that is to cloud database storage. It will be developed using Netbeans IDE. It contains different module which is available for different section[7].

2.2.4Cloud Based College Management System

This system work mainly on Cloud Platform Techniques, Microsoft Azure to handle the database and code with monitoring of network for reasons of security and handling PHP extensions using PHP libraries inbuilt and Database MySQL and windows server 2016,virtual server. The now work can improve by PHP Code of each page method of ajax calling technique of JavaScript/JQuery Programming Language[8].

2.2.5 Wildlife in the cloud: A new paradigm

Research in wildlife management increasingly relies on quantitative population models. However, a remaining challenge is to have end-users, who are often alienated by mathematics, benefiting from this research. I propose a new approach, 'wildlife in the cloud,' to enable active learning by practitioners from cloud-based ecological models whose complexity remains invisible to the user. The tool takes the form of a user-friendly cloud-app through which users can compare the effects of alternative management decisions, and may feed into adjustment of their hunting strategy.[2].

2.2.6 Waste Connection's app

It lets users keep track of their garbage and recycling schedules in a simple format. Users can also use the “waste wizard” to find the proper way to dispose of a particular material, such as plastic cutlery, food waste, or water bottles. it is currently being used by various municipalities such as Vancouver, Denver etc[11].

The implementation of the system is done with the help of Android Studio and Java and establishing connectivity using firebase. With the help of application the user can easily register themselves and select which trees they can plant in their terrain, they can opt to plant it themselves in the nursery space provided to them or can chose to let the Admin to do it.



Figure 2.2 flow of website

2.3 DESIGN

2.3.1 Design

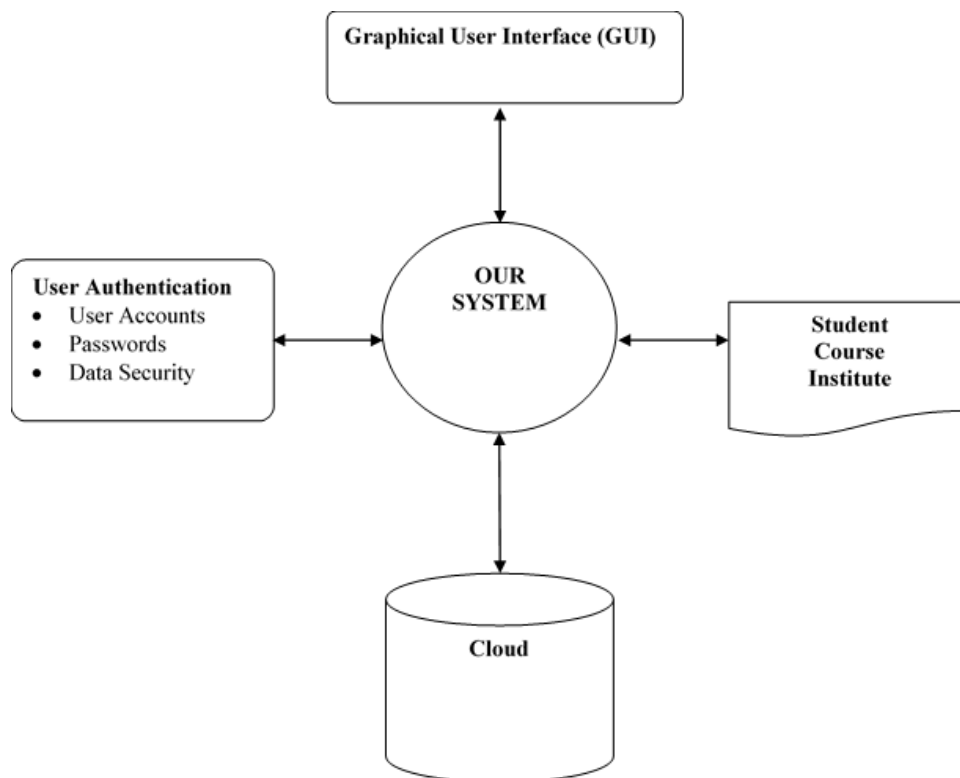


Figure 2.3 Design

In this we define the projects structure, feature, criteria for success the main objective is to create the desired project. our system is based on cloud.

2.3.2 Implementation And Evaluation

2.3.2.1 Android Studio

Android Studio is the official Integrated development environment for Google's Android operating system. Once an app

has been compiled with Android Studio, it can be published on the Google Play Store.

Android Studio supports all the same programming languages such as Java, C++, Go etc. Android Studio 3.0 or above supports Kotlin.

Android Studio supports Gradle-based build support. Lint tools to catch performance, usability, version compatibility etc. A rich layout editor that allows the user drag and drop UI components and many other options.

2.3.2.2 Java

Java is object-oriented strongly typed, programming language, it is on-demand. It is stored, compiled, and run on Java Virtual Machine.

Java programs use the Java virtual machine as abstraction and do not access the operating system directly. This makes Java programs highly portable. A Java program can run unmodified on all supported platforms.

Java source code is transferred into the byte code format which does not depend on the target platform. These byte code instructions will be interpreted by the JVM. The JVM contains a so called Hotspot-Compiler which translates performance critical byte code instructions into native code instructions. As a programming language java is Easy to use, object oriented, platform independent, interpreted, compiled and supports automatic memory management.

2.3.3 GUI

The graphical user interface is one important factor by which the user can easily access the website to understand what he need to do in order to use the

software effectively[3]. A easy and effective GUI is good for both user as well as developer.

Some of the GUI are as follow:

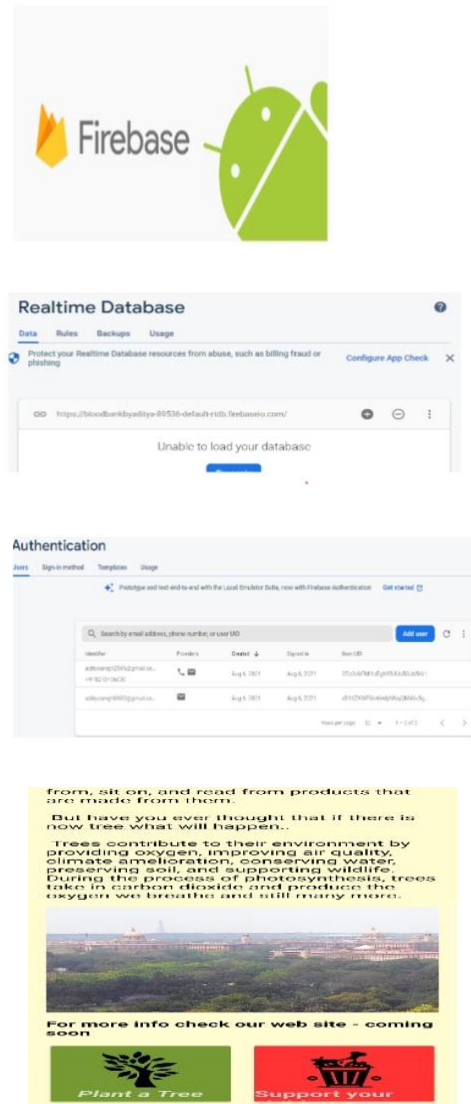


Figure 2.3 some screens

2.4 RESULT

We have come on result that our system working fine in each case. It is user friendly as well as efficient to use. we have done different things to verify the performance.

TEST CASE RESULT

Test Case#	Description	Result
TC#1	Splash screen	Passed
TC#2	Login	Passed
TC#3	Validating	Passed
TC#4	Content	Passed
TC#5	Registration	Passed
TC#6	Payment	Passed
TC#7	Logout	Passed

2.5 DISCUSSION

System was completely done and was duly coded. Each module of the project is checked to ensure they are fully functional units.[4] This was done by checking each unit to give assurance that it functions as required and that it performed exactly as defined. The success of each individual gave us go ahead to carryout testing properly[9].

The defined system was validated by the using a series of short questionnaire that was completely filled by representatives users who have used the system and give suggestion according to the need .This was done to the assess if the system met their respective needs and requirements. It was also find that it is easy to access the data as well as available when needed. With the flexibility of the cloud it is quite useful and better version of management system.

2.6 CONCLUSION & FUTURE SCOPE

2.6.1 CONCLUSION

A system means a lot of experience. I learned a lot of thing this project development. This project has also sharpened my concept of android development.

2.6.2 FUTURE SCOPE

To make our system more user friendly. I plan to add a waste management system.

2.7 REFERENCES

- [1] Paul R Jepson, Richard James Ladle, "Nature apps: Waiting for revolution", published on october 2015 .
- [2] Chapron G., "Wildlife in the cloud: A new paradigm", published in 2012.
- [3] Zilli, O. Parson, G. Merrett and A. Rogres, "A hidden markov model-based acoustic cicada detector for crowdsourced smartphone biodiversity monitoring," in 23rd International joint conference on Artificial Intelligence, Beijing, CN, 2945-2951.
- [4] R. M. Leod, management Information Systems, Third Ed., Science Research Associates, 1986, pp. 17-19.
- [5] L. Long, college management System, Prentice Hall, 1989, pp.116-117.
- [6] Research on Development of android applications – Jianye Liu, Jiankun Yu.
- [7] Institute Management System (Ims) Ijarce ,MustafizSharique pp 1-6
- [8] A Research Paper On College Management System Lalit Mohan Joshi M.Tech Scholar.
- [9] IJCSMC, Vol. 7, Issue. 7, July 2018, pg.20 – 31 Design and Implement a Novel Student Information

Management System – Case Study
Ihsan Ali Hassan

- [10] Shimozono K., Itsukiy M., Harasaka Y. and Furukawa Z., (2010), “User Management in an Educational Computer System: Personal Information Management,” Fukuoka, Japan, pp. 1-6.
- [11] Rakesh kumar,sonu, Agarwal, pragya -Extremely Effective Crm Solution Using Salesforce, JETIR,
- [12] Norasiah M. A. and Norhayati A., (2003), “Intelligent Student Information System,” in 4th National Conference on Telecommunication Technology Proceedings, Shah Alam, Malaysia, pp. 212-215
- [13] Bharamagoudar S. R., Geeta R. B. and Totad S. G., (2013), “Web Based Student Information Management System,” International Journal of Advanced Research in Computer and Communication Engineering, vol. 2, no. 6, pp. 2342-2348.
- [14] Jensen, J. Schwenk, N. Gruschka, & L. LO. Iacono (2009). On technical security issues in cloud computing. 2009 - 2009 IEEE International Conference on Cloud Computing, 109–116.
- [15] Connolly T. M. and Begg C. E., (2010), Databases Systems: A Practical Approach to Design, Implementation and Management, 5th ed., Boston: Pearson, pp. 340-344.
- [16] OL. Google Android Developers, Android Develop Guide, <http://developer.android.com/guide/topics/fundamentals.html>
- [17] M. Fengsheng Yang, Android Application Development Revelation, China Machine press, 2010.
- [18] M. Junmin Ye, Software Engineering, Tsinghua University Press,2006.

CHAPTER 3

FEASIBILITY STUDY

3.1 INTRODUCTION

Feasibility of the system in an important aspect, which is to be considered. The system needs to satisfy the law of economic, which states that the maximum output should be yielded in minimum available resources.

A feasibility analysis evaluates the project's potential for success; therefore, perceived objectivity is an essential factor in the credibility of the study for potential investors and lending institutions. There are five types of feasibility study—separate areas that a feasibility study examines, described below.

1. Technical Feasibility

This assessment focuses on the technical resources available to the organization. It helps organizations determine whether the technical resources meet capacity and whether the technical team is capable of converting the ideas into working systems. Technical feasibility also involves the evaluation of the hardware, software, and other technical requirements of the proposed system. As an exaggerated example, an organization wouldn't want to try to put Star Trek's transporters in their building—currently, this project is not technically feasible.

2. Economic Feasibility

This assessment typically involves a cost/ benefits analysis of the project, helping organizations determine the viability, cost, and benefits associated with a project before financial resources are allocated. It also serves as an independent project assessment and enhances project credibility—helping decision-makers determine the positive economic benefits to the organization that the proposed project will provide.

3. Legal Feasibility

This assessment investigates whether any aspect of the proposed project conflicts with legal requirements like zoning laws, data protection acts or social media laws. Let's say an organization wants to construct a new office building in a specific location. A feasibility study might reveal the organization's ideal location isn't zoned for that type of business. That organization has just saved considerable time and effort by learning that their project was not feasible right from the beginning.

4. Operational Feasibility

This assessment involves undertaking a study to analyze and determine whether—and how well—the organization's needs can be met by completing the project. Operational feasibility

studies also examine how a project plan satisfies the requirements identified in the requirements analysis phase of system development.

5. Scheduling Feasibility

This assessment is the most important for project success; after all, a project will fail if not completed on time. In scheduling feasibility, an organization estimates how much time the project will take to complete.

When these areas have all been examined, the feasibility analysis helps identify any constraints the proposed project may face, including:

- Internal Project Constraints: Technical, Technology, Budget, Resource, etc.
- Internal Corporate Constraints: Financial, Marketing, Export, etc.
- External Constraints: Logistics, Environment, Laws, and Regulations, et

3.2 MAIN ASPECTS

There are three aspects of feasibility to be considered namely.

1. Technical
2. Operational
3. Economical

TECHNICAL:

In the technical aspects one may consider the hardware equipment for the installation of the software. The system being centralized will required very little hardware appliances. Hence this helps the system to work smoothly with limited amount of working capitals.

OPERATIONAL:

In the operational aspects may think of the benefits of the workload that many a personal may have to share. This is eased out and the required output may be retrieved in a very short time. Thus there is accuracy in the work on time is also saved there will be very little work that needs to be performed.

ECONOMICAL:

Economical system is definitely feasible because the hardware requirement is less and the operational working for the system requires less number of recruits. This help introduction over-staffing and wastage funds.

We studied on the position to evaluate solution. Most important factors in this study were tending to overlook the confusion inherent in system Development the constraints and the assumed studies. It can be started that it the feasibility study is to serve as a decision document it must answer three key questions.

1. Is there a new and better way to do the job that will benefit the user?
2. What are the costs and savings of the alternatives?
3. What is recommended?

On these questions it can be explained that feasibility study of the system includes following different angles.

3.2.1 Technical feasibility:

This centers on the existing computer system (hardware, software etc.) and to what extent it can support the proposed additional equipment .in this stage of study, we have collected information about technical tools available by which I could decide my system design as the technical requirements.

3.2.2 Operational Feasibility:

In this stage of study we have checked the staff availability. I concentrate on knowledge of end users that are going to use the system. This is also called as behavioral feasibility in which I have studied on following aspects; people are inherently resistant to change, and computers have been known to facilitate change .An estimate has been made to how strong a reaction the user staff is having toward the development of a computerized system. It is common knowledge that computer installations have something to do with turnover. I had explained that there is need to educate and train the staff on new ways of conducting business.

3.2.3Economical feasibility:

Economical analysis is the most frequently used method for evaluating the effectiveness of candidate system. More commonly known as cost\benefit analysis, the procedure is to determine the benefits and savings that benefits outweigh costs. The decision was to design and implement system because it is for having chanced to be approved. This is an on going effort that improves the accuracy at each phase of the system life cycle. In developing cost estimates for a system I need to consider several cost elements. Among these is hardware personal facility. Operating and supply costs.

3.3BENEFITS

Benefits of conducting a feasibility study:

- Improves project teams' focus
- Identifies new opportunities
- Provides valuable information for a "go/no-go" decision
- Narrows the business alternatives
- Identifies a valid reason to undertake the project
- Enhances the success rate by evaluating multiple parameters
- Aids decision-making on the project
- Identifies reasons not to proceed

3.4SYSTEM REQUIREMENT SPECIFICATION

Any system can be designed after specifies the requirement of the user about that system. For this first of all gathered information from user by the preliminary investigation which is starting investigation about user requirement..

The data that the analysts collect during preliminary investigation are gathered through the various preliminary methods.

Documents Reviewing Organization

The analysts conducting the investigation first learn the organization involved in, or affected by the project. Analysts can get some details by examining organization charts and studying written operating procedures.

Collected data is usually of the current operating procedure:

- The information relating to clients, projects and students and the relationship between them was held manually.
- Managing of follow-ups was through manual forms.
- Complaints require another tedious work to maintain and solve.
- Payments details had to be maintained differently.

Gathering Information By Asking Questions

Interviewing is the most commonly used techniques in analysis. It is always necessary first to approach someone and ask them what their problems are, and later to discuss with them the result of your analysis.

Questionnaires

Questionnaires provide an alternative to interviews for finding out information about a system. Questionnaires are made up of questions about information sought by analyst. The questionnaire is then sent to the user, and the analyst analyzes the replies.

Electronic Data Gathering

Electronic communication systems are increasingly being used to gather information. Thus it is possible to use electronic mail to broadcast a question to a number of users in an organization to obtain their viewpoint on a particular issue.

In my project, with the help of Marg software solutions, I have send questionnaire through electronic mail to twenty employees of the company and retrieved the information regarding the problem faced by existing system.

Interviews

Interview allows the analysts to learn more about the nature of the project request and reason of submitting it. Interviews should provide details that further explain the project and show whether assistance is merited economically, operationally or technically.

One of the most important points about interviewing is that what question you need to ask.

It is often convenient to make a distinction between three kinds of question that is

- Open questions
- Closed question
- Probes

Open questions are general question that establish a persons view point on a particular subject.

Closed questions are specific and usually require a specific answer.

Probes are question that follow up an earlier answer.

CHAPTER 4

DESIGN

4.1 INTRODUCTION

System is created to solve problems. One can think of the systems approach as an organized way of dealing with a problem. In this dynamic world, the subject system analysis and design, mainly deals with the software development activities.

Since a new system is to be developed, the one most important phases of software development life cycle is system requirement gathering and analysis. Analysis is a detailed study of various operations performed by a system and their relationship within and outside the system. Using the following steps it becomes easy to draw the exact boundary of the new system under consideration.

All procedures, requirements must be analysed and documented in the form of detailed DFDs, logical data structure and miniature specifications.

System analyses also include sub-dividing of complex process involving the entire system, identification of data store and manual processes.

4.2 SYSTEM DESIGN

System design is the process of planning a new system or to replace the existing system. Simply, system design is like the blueprint for building, it specifies all the features that are to be in the finished product.

System design phase follows system analysis phase. Design is concerned with identifying functions, data streams among those functions, maintaining a record of the design decisions and providing a blueprint the implementation phase.

Design is the bridge between system analysis and system implementation. Some of the essential fundamental concepts involved in the design of application software are:

- Abstraction
- Modularity
- Verification

Abstraction is used to construct solutions to problem without having to take account of the intricate details of the various component sub problems. Abstraction allows system designer to make step-wise refinement, which at each stage of the design may hide, unnecessary details associated with representation or implementation from the surrounding environment.

Modularity is concerned with decomposing of main module into well-defined manageable units with well-defined interfaces among the units. This enhances design clarity, which in turn eases implementation, Debugging, Testing, Documenting and Maintenance of the software product. Modularity viewed in this sense is a vital tool in the construction of large software projects.

Verification is fundamental concept in software design. A design is verifiable if it can be demonstrated that the design will result in implementation that satisfies the customer's requirements. Verification is of two types namely.

- Verification that the software requirements analysis satisfies the customer's needs.
- Verification that the design satisfies the requirement analysis.

Some of the important factors of quality that are to be considered in the design of application software are:

Reliability:

The software should behave strictly according to the original specification and should function smoothly under normal conditions.

Extensibility:

The software should be capable of adapting easily to changes in the specification.

Reusability:

The software should be developed using a modular approach, which permits modules to be reused by other application, if possible.

The System Design briefly describes the concept of system design and it contains four sections. The first section briefly describes the features that the system is going to provide to the user and the outputs that the proposed system is going to offer.

The second section namely Logical Design describes the Data Flow Diagrams, which show clearly the data movements, the processes and the data sources, and sinks, E-R diagrams which represent the overall logical design of the database, and high-level process structure of the system.

Preliminary Design:

Preliminary design is basically concerned with deriving an overall picture of the system. Deriving entire system into modules and sub-modules while keeping Cohesion and Coupling factors in mind. Tools, which assist in preliminary design process, are Data Flow Diagrams.

Code design:

The purpose of code is to facilitate the identification and retrieval for items of information. A code is an ordered collection of symbols designed to provide unique identification of an entity or attribute. To achieve unique identification there must be only one place where the identified entity or the attribute can be entered in the code; conversely there must be a place in the code for every thing that is to be identified. This mutually exclusive feature must be built into any coding system.

The codes for this system are designed with two features in mind. Optimum human oriented use and machine efficiency. They are also operable i.e., they are adequate for present and anticipate data processing both for machine and human use.

Input /Output design:

is a part of overall system design, which requires very careful attention. The main objectives of input design are:

- To produce a cost-effective method of input.
- To achieve the highest possible level of accuracy.
- To ensure that the input is acceptable to and understood by the user staff.

Outputs from computer systems are required primarily to communicate the results of processing to users. They are also to provide a permanent hard copy of these results for later consultation.

The various types of outputs are required by this system are given below:

- External outputs, whose destination is outside the concern and which require special attention because they, project the image of the concern.
- Internal outputs, whose destination is within the concern and which require careful design because they are the user's main interface within the computer.
- Operation outputs, whose use is purely within the computer department, E.g., program listings, usage statistics etc,

4.3 SDLC

Software Development Life Cycle (SDLC) is a framework that defines the steps involved in the development of software at each phase. It covers the detailed plan for building, deploying and maintaining the software.

SDLC defines the complete cycle of development i.e. all the tasks involved in planning, creating, testing, and deploying a Software Product.



Figure 4.1: Above image depicting the planning step

SDLC Phases

Given below are the various phases:

- Requirement gathering and analysis
- Design
- Implementation or coding
- Testing
- Deployment
- Maintenance

Requirement Gathering and Analysis

During this phase, all the relevant information is collected from the customer to develop a product as per their expectation. Any ambiguities must be resolved in this phase only. Business analyst and Project Manager set up a meeting with the customer to gather all the information like what the customer wants to build, who will be the end-user, what is the purpose of the product. Before building a product a core understanding or knowledge of the product is very important.

Once the requirement gathering is done, an analysis is done to check the feasibility of the development of a product. In case of any ambiguity, a call is set up for further discussion. Once the requirement is clearly understood, the SRS (Software Requirement Specification) document is created. This document should be thoroughly understood by the developers and also should be reviewed by the customer for future reference.

Design

In this phase, the requirement gathered in the SRS document is used as an input and software architecture that is used for implementing system development is derived.

Implementation or Coding

Implementation/Coding starts once the developer gets the Design document. The Software design is translated into source code. All the components of the software are implemented in this phase.

Testing

Testing starts once the coding is complete and the modules are released for testing. In this phase, the developed software is tested thoroughly and any defects found are assigned to developers to get them fixed.

Deployment

Once the product is tested, it is deployed in the production environment or first UAT (User Acceptance testing) is done depending on the customer expectation.

Maintenance

After the deployment of a product on the production environment, maintenance of the product i.e. if any issue comes up and needs to be fixed or any enhancement is to be done is taken care by the developers.

4.4 SOFTWARE ENGG. PARADIGM APPLIED

Software engineering is a layered technology. The foundation for software engineering is the process layer. Software engineering processes the glue that holds the technology layers together and enables ratios and timely development of computer software. Process defines a framework for a set of key process areas that must be established for effective delivery of software engineering technology.

Software engineering methods provide the technical how-to's for building software. Methods encompass a broad array of tasks that include requirements analysis, design, program construction, testing and support. Software engineering tools provide automated or semi-automated support for the process and the methods. When tools are integrated so that information created by one tool can be used by another tool, a system for the support of software development, called computer-aided software engineering is established.

The following paradigms are available:

1. The Waterfall Model
 2. The Prototyping Model
 3. The Spiral model
- Etc.

4.4.1 The Prototype model

The prototype model requires that before carrying out the development of actual software, a working prototype of the system should be built. A prototype is a toy implementation of the system. A prototype usually turns out to be a very crude version of the actual system, possibly exhibiting limited functional capabilities, low reliability, and inefficient performance as compared to actual software. In many instances, the client only has a general view of what is expected from the software product. In such a scenario where there is an absence of detailed information regarding the input to the system, the processing needs, and the output requirement, the prototyping model may be employed.

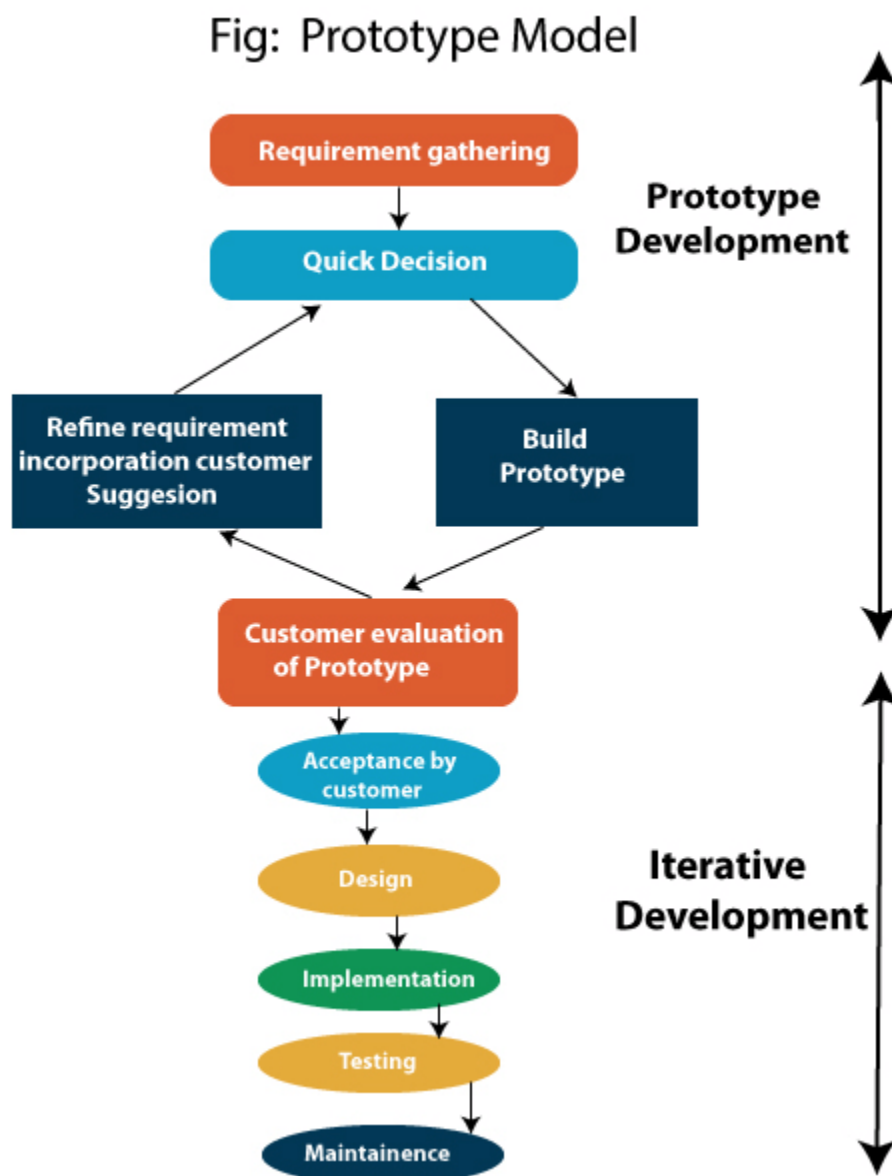


Figure 4.2:Prototype model

4.4.1.1 Advantage of Prototype Model

1. Reduce the risk of incorrect user requirement
2. Good where requirement are changing/uncommitted
3. Regular visible process aids Presidency
4. Support early product marketing
5. Reduce Maintenance cost.
6. Errors can be detected much earlier as the system is made side by side.

4.4.1.2 Disadvantage of Prototype Model

1. An unstable/badly implemented prototype often becomes the final product.
2. Require extensive customer collaboration
 - Costs customer money
 - Needs committed customer
 - Difficult to finish if customer withdraw
 - May be too customer specific, no broad market
3. Difficult to know how long the project will last.
4. Easy to fall back into the code and fix without proper requirement analysis, design, customer evaluation, and feedback.
5. Prototyping tools are expensive.
6. Special tools & techniques are required to build a prototype.
7. It is a time-consuming process.

4.5 DFD

DFD is the abbreviation for **Data Flow Diagram**. The flow of data of a system or a process is represented by DFD. It also gives insight into the inputs and outputs of each entity and the process itself. DFD does not have control flow and no loops or decision rules are present. Specific operations depending on the type of data can be explained by a flowchart. Data Flow Diagram can be represented in several ways.

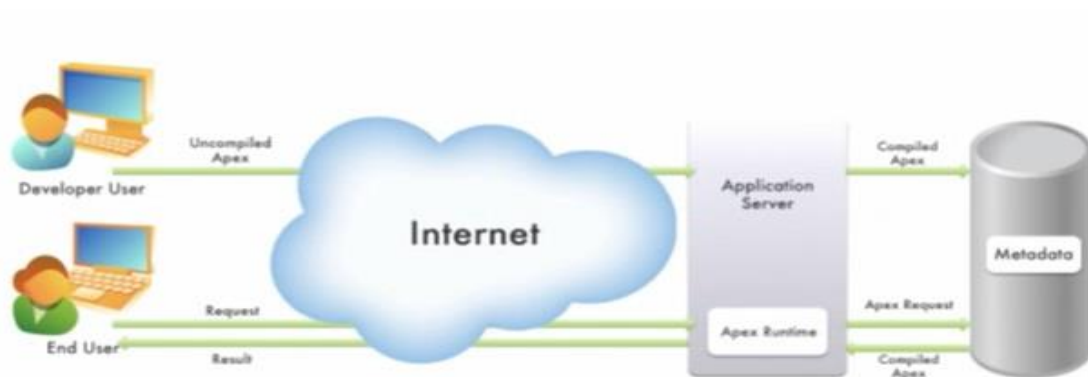


Figure 4.3 Apex working

Level 0 DFD :

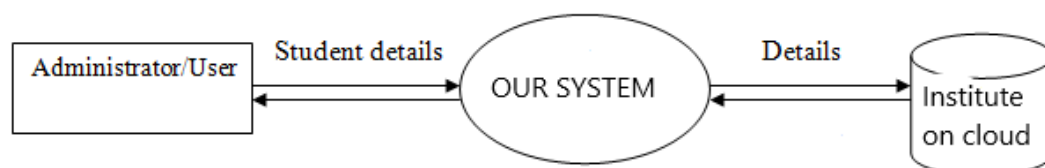


Figure 4.4 Context level

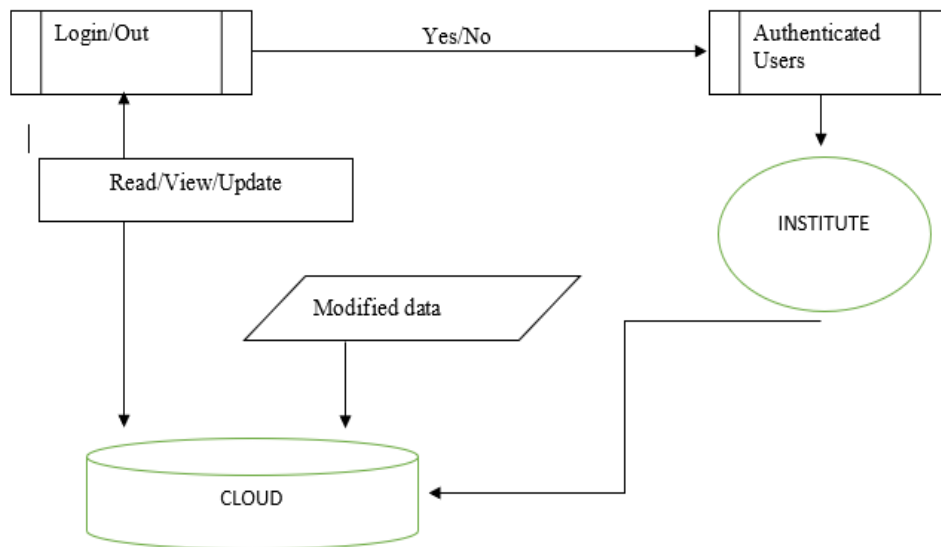


Figure 4.5Dfd 1 level

4.6 UML use case diagram

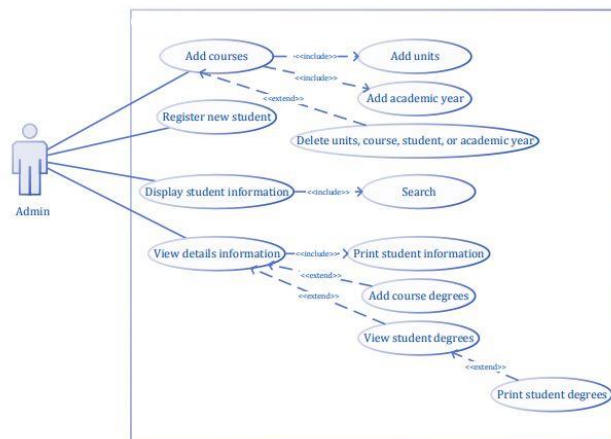


Figure 4.6 Uml use case

4.7ER DIAGRAM

An Entity–relationship model (ER model) describes the structure of a database with the help of a diagram, which is known as Entity Relationship Diagram (ER Diagram). An ER model is a design or blueprint of a database that can later be implemented as a database. The main components of E-R model are: entity set and relationship set.

An ER diagram has three main components:

1. Entity
2. Attribute
3. Relationship

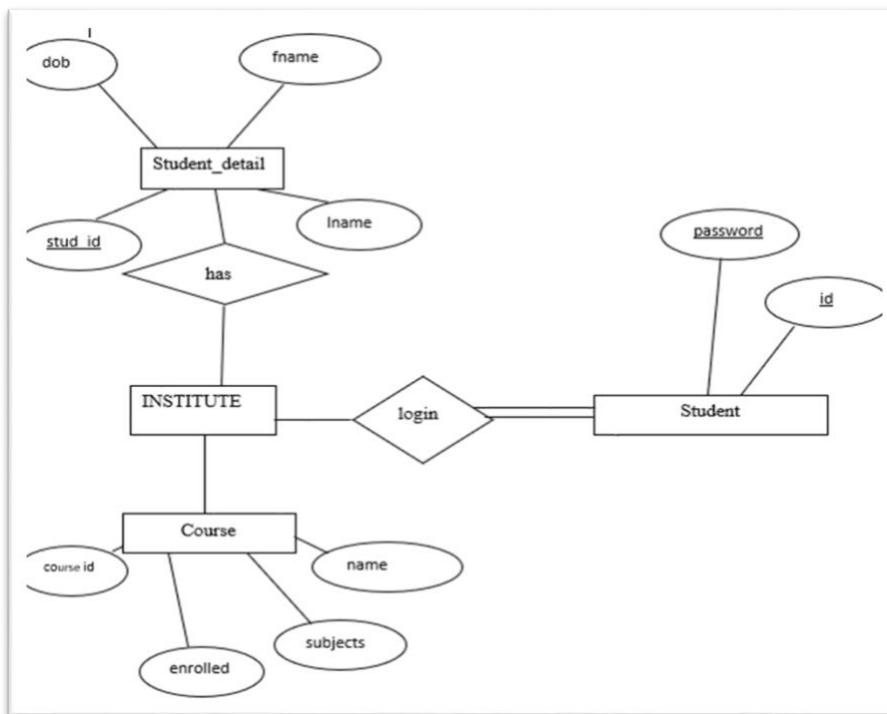


Figure 4.7:ER diagram of system

4.7.1 ER- Diagram Notations

ER- Diagram is a visual representation of data that describe how data is related to each other.

- **Rectangles:** This symbol represent entity types
- **Ellipses :**Symbolrepresent attributes
- **Diamonds:** This symbolrepresents relationship types
- **Lines:** It links attributes to entity types and entity types with other relationship types
- **Primary key:** attributes are underlined
- **Double Ellipses:** Represent multi-valued attributes

CHAPTER 5

REPORT

5.1GIST

The diagram **figure 5.1**,depicting our system.
We have designed and developed an easy, Useful, reliable system.

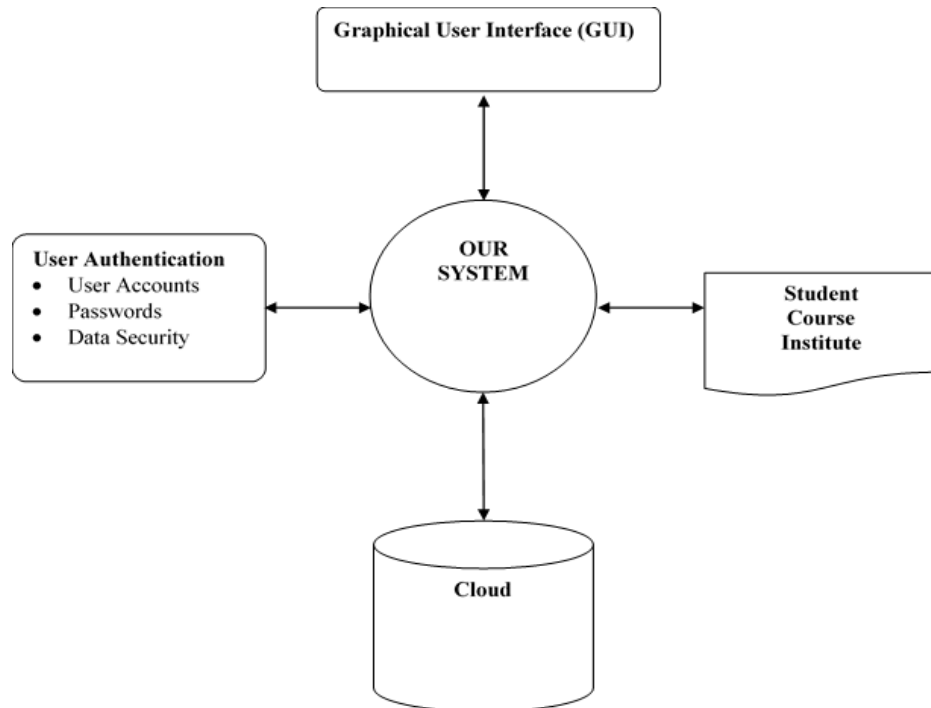


Figure 5.1 System

This gives a high level view of the system with the main components and the services they provide and how they communicate. It consists of the general graphical user interface facilities.

5.2 SOME SCREENSHOTS

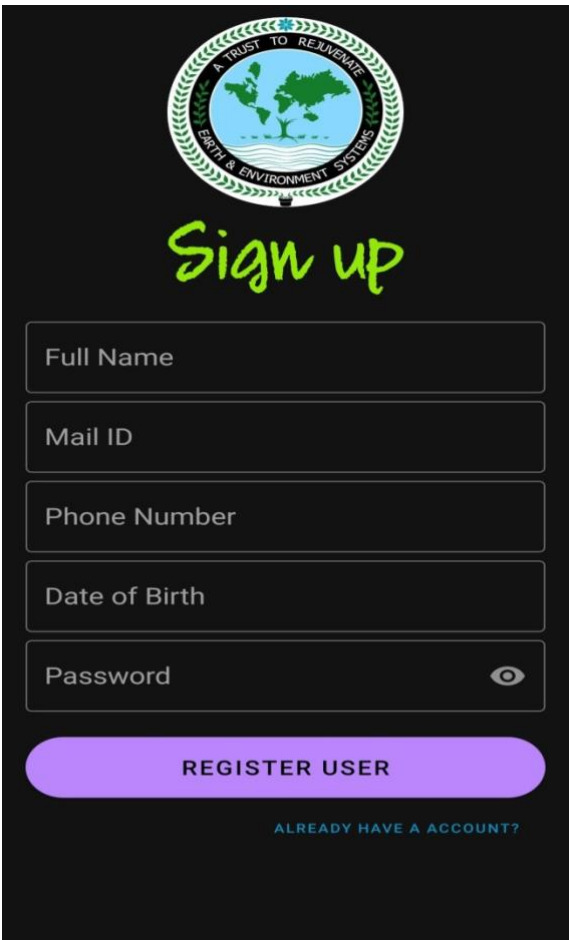
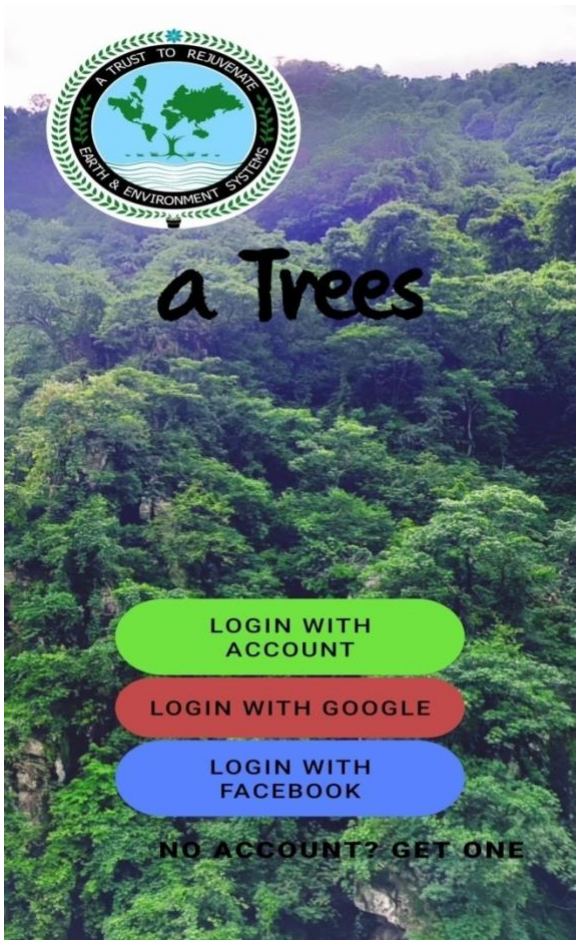
HOME PAGES



SPLASH SCREEN



REGISTRATION



PLANT A TREE

Plant a Tree

*What we learn from
a tree is rise from the
Earth to be able to talk
with the sky*

Want to plant yourself : ☐ Yes ☐ No

NEXT STEP

CHAPTER 6

CODING

This chapter contains some codes of the project. The goal of the coding is to translate the design of the system into code in a given programming language. For a given design, the aim of this phase is to implement the design in the best possible manner. The coding phase affects both testing and maintenance profoundly.

Some Codes are as Written below:

#Dashboard

```
<?xml version="1.0" encoding="utf-8"?>
<ScrollViewxmlns: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="match_parent"
tools:context=".Dashboard"
android:background="#ffdd0">
```

```
<androidx.constraintlayout.widget.ConstraintLayout
android:layout_width="match_parent"
android:layout_height="match_parent">
```

```
<ImageView
android:id="@+id/imageView3"
android:layout_width="175dp"
android:layout_height="175dp"
android:src="@drawable/logo"
```

```
android:layout_marginTop="15dp"
app:layout_constraintEnd_toEndOf="parent"
app:layout_constraintStart_toStartOf="parent"
app:layout_constraintTop_toTopOf="parent" />
```

```
<LinearLayout
android:id="@+id/dash_layer"
android:layout_width="match_parent"
android:layout_height="wrap_content"
android:orientation="vertical"
app:layout_constraintEnd_toEndOf="parent"
app:layout_constraintStart_toStartOf="parent"
app:layout_constraintTop_toBottomOf="@+id/imageView3">
```

```
<TextView
android:id="@+id/textView4"
android:layout_width="match_parent"
android:layout_height="wrap_content"
android:layout_marginStart="30dp"
android:layout_marginEnd="30dp"
android:textSize="25dp"
android:gravity="center"
android:textColor="#000"
android:textStyle="bold|italic"/>
```

```
<TextView
android:layout_width="match_parent"
android:layout_height="wrap_content"
android:text="Our Team Welcomes You to join the Noble Cause"
android:textSize="25dp"
android:textStyle="bold|italic"
android:layout_marginTop="5dp"
android:layout_marginLeft="30dp"
android:layout_marginRight="30dp"
android:gravity="center"
android:textColor="#000" />
</LinearLayout>
```

```
<LinearLayout
android:id="@+id/layer1"
android:layout_width="0dp"
android:layout_height="wrap_content"
android:orientation="horizontal"
app:layout_constraintEnd_toEndOf="parent"
```

```
android:padding="9dp"
app:layout_constraintStart_toStartOf="parent"
app:layout_constraintTop_toBottomOf="@+id/dash_layer">
```

```
<androidx.cardview.widget.CardView
android:id="@+id/d1"
android:layout_width="0dp"
app:cardBackgroundColor="#779933"
android:layout_height="match_parent"
android:layout_margin="10dp"
android:layout_weight="1"
android:padding="200dp">
```

```
<RelativeLayout
android:layout_width="match_parent"
android:layout_height="match_parent">
```

```
<ImageView
android:id="@+id/imageView4"
android:layout_width="92dp"
android:layout_height="83dp"
android:layout_marginTop="12dp"
android:layout_centerHorizontal="true"
app:srcCompat="@drawable/plant" />
```

```
<TextView
android:id="@+id/textView6"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:layout_below="@id/imageView4"
android:layout_centerHorizontal="true"
android:text="Plant Tree Project"
android:textSize="20dp"
android:gravity="center"
android:textStyle="bold|italic" />
```

```
<View
android:layout_width="match_parent"
android:layout_height="1dp"
android:background="@color/black"
android:layout_below="@+id/textView6"
android:layout_marginBottom="10dp"/>
```

```
</RelativeLayout>
```

```
</androidx.cardview.widget.CardView>
```

```
<androidx.cardview.widget.CardView  
android:layout_width="0dp"  
android:id="@+id/d2"  
app:cardBackgroundColor="#FF3333"  
android:layout_height="match_parent"  
android:layout_weight="1"  
android:layout_margin="10dp">
```

```
<RelativeLayout  
android:layout_width="match_parent"  
android:layout_height="match_parent">
```

```
<ImageView  
android:id="@+id/imageView5"  
android:layout_width="92dp"  
android:layout_height="83dp"  
android:layout_marginTop="12dp"  
android:layout_centerHorizontal="true"  
app:srcCompat="@drawable/waste" />
```

```
<TextView  
android:id="@+id/textView7"  
android:layout_width="wrap_content"  
android:layout_height="wrap_content"  
android:layout_below="@id/imageView5"  
android:layout_centerHorizontal="true"  
android:text="Waste Manage"  
android:textSize="20dp"  
android:gravity="center"  
android:textStyle="bold" />
```

```
<View  
android:layout_width="match_parent"  
android:layout_height="1dp"  
android:background="@color/black"  
android:layout_below="@+id/textView7"  
android:layout_marginBottom="10dp"/>
```

```
</RelativeLayout>
```

```
</androidx.cardview.widget.CardView>
```

</LinearLayout>

```
<LinearLayout
android:id="@+id/layer2"
android:layout_width="0dp"
android:layout_height="wrap_content"
android:orientation="horizontal"
app:layout_constraintEnd_toEndOf="parent"
android:paddingLeft="5dp"
android:paddingBottom="5dp"
android:paddingEnd="5dp"
android:padding="5dp"
app:layout_constraintStart_toStartOf="parent"
app:layout_constraintTop_toBottomOf="@+id/layer1">
```

```
<androidx.cardview.widget.CardView
android:id="@+id/d3"
android:layout_width="0dp"
app:cardBackgroundColor="#9966cc"
android:layout_height="match_parent"
android:layout_weight="1"
android:layout_marginLeft="10dp"
android:layout_marginRight="10dp"
android:layout_marginBottom="10dp">
```

```
<RelativeLayout
android:layout_width="match_parent"
android:layout_height="match_parent">
```

```
<ImageView
android:id="@+id/imageView7"
android:layout_width="92dp"
android:layout_height="83dp"
android:layout_marginTop="12dp"
android:layout_centerHorizontal="true"
app:srcCompat="@drawable/member" />
```

```
<TextView
android:id="@+id/textView9"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:layout_below="@id/imageView7"
android:layout_centerHorizontal="true"
```

```
android:text="Become member"
android:textSize="20dp"
android:gravity="center"
android:textStyle="bold" />
```

```
<View
android:layout_width="match_parent"
android:layout_height="1dp"
android:background="@color/black"
android:layout_below="@+id/textView9"
android:layout_marginBottom="10dp"/>
```

```
</RelativeLayout>
```

```
</androidx.cardview.widget.CardView>
```

```
<androidx.cardview.widget.CardView
android:id="@+id/d4"
android:layout_width="0dp"
app:cardBackgroundColor="#FFA500"
android:layout_height="match_parent"
android:layout_marginLeft="10dp"
android:layout_marginRight="10dp"
android:layout_marginBottom="10dp"
android:layout_weight="1">
```

```
<RelativeLayout
android:layout_width="match_parent"
android:layout_height="match_parent">
```

```
<ImageView
android:id="@+id/imageView8"
android:layout_width="92dp"
android:layout_height="83dp"
android:layout_marginTop="12dp"
android:layout_centerHorizontal="true"
app:srcCompat="@drawable/about_us" />
```

```
<TextView
android:id="@+id/textView10"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:layout_below="@id/imageView8"
```

```
android:layout_centerHorizontal="true"
android:text="About Us"
android:gravity="center"
android:textSize="20dp"
android:textStyle="bold|italic" />
```

```
<View
android:layout_width="match_parent"
android:layout_height="1dp"
android:background="@color/black"
android:layout_below="@+id/textView10"
android:layout_marginBottom="10dp"/>
```

```
</RelativeLayout>
</androidx.cardview.widget.CardView>
```

```
</LinearLayout>
```

```
<Button
android:id="@+id/logout"
android:textStyle="bold"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:text="Log out"
app:layout_constraintEnd_toEndOf="parent"
app:layout_constraintStart_toStartOf="parent"
app:layout_constraintTop_toBottomOf="@+id/layer2" />
```

```
</androidx.constraintlayout.widget.ConstraintLayout>
```

```
</ScrollView>
```

```
##dashboard
```

```
package com.example.projecttrees;
```

```
import androidx.annotation.NonNull;
import androidx.appcompat.app.AppCompatActivity;
import androidx.cardview.widget.CardView;
```

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

```

import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.TextView;
import android.widget.Toast;

import com.google.firebase.auth.FirebaseAuth;
import com.google.firebase.auth.FirebaseUser;
import com.google.firebase.database.DataSnapshot;
import com.google.firebase.database.DatabaseError;
import com.google.firebase.database.DatabaseReference;
import com.google.firebase.database.FirebaseDatabase;
import com.google.firebase.database.ValueEventListener;

public class Dashboard extends AppCompatActivity implements View.OnClickListener
{

    CardView c1,c2,c3,c4;

    private FirebaseUser user;

    private DatabaseReference reference;

    private String userID;

    Button logout;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_dashboard);

        //logout
        logout=(Button) findViewById(R.id.logout);

        logout.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                FirebaseAuth.getInstance().signOut();
                startActivity(new Intent(Dashboard.this,Welcome.class));
            }
        });

        //data by user

```



```

        user = FirebaseAuth.getInstance().getCurrentUser();
        reference = FirebaseDatabase.getInstance().getReference("Users");
        userID = user.getId();

        final TextView clint = (TextView) findViewById(R.id.textView4);

        reference.child(userID).addListenerForSingleValueEvent(new ValueEventListener() {
            @Override
            public void onDataChange(@NonNull DataSnapshot snapshot) {
                User userProfile = snapshot.getValue(User.class);

                if (userProfile != null) {
                    String clint_name = userProfile.name;
                    clint.setText("Welcome " + clint_name);
                }
            }

            @Override
            public void onCancelled(@NonNull DatabaseError error) {
                Toast.makeText(Dashboard.this, "Something Wrong happened!",
                    Toast.LENGTH_LONG).show();
            }
        });

        //Activity handler
        c1 = (CardView) findViewById(R.id.d1);
        c1.setOnClickListener(this);

        c2 = (CardView) findViewById(R.id.d2);
        c2.setOnClickListener(this);

        c3 = (CardView) findViewById(R.id.d3);
        c3.setOnClickListener(this);

        c4 = (CardView) findViewById(R.id.d4);
        c4.setOnClickListener(this);
    }

    //card click
    @Override
    public void onClick(View v) {
        switch (v.getId()) {
            case R.id.d1:
                startActivity(new Intent(this, Plant_Menu.class));

```

```
        break;

        case R.id.d2:
startActivity(new Intent(this,Waste.class));
        break;

        case R.id.d3:
startActivity(new Intent(this,Members.class));
        break;

        case R.id.d4:
startActivity(new Intent(this,About_us_board.class));
        break;
    }
}
}
```

CHAPTER 7

TESTING

7.1 INTRODUCTION

Software Testing is a critical element of software quality assurance and represents the ultimate review of specification, design and code generation. The increasing visibility of software as a system element and the attendant “costs” associated with a software failure are motivating forces for well planned, thorough testing.

7.1.1 Testing Objectives

The following are the testing objectives:

- Testing is a process of executing a program with the intent of finding an error.
- A good test case is one that has a high probability of finding an as-yet-undiscovered error
- successful test is one that uncovers an as yet undiscovered error.

7.1.2 Testing Principles

The basic principles that guide software testing are as follows:

- All tests should be traceable to customer requirements.
- Tests should be planned long before testing begins.
- The parate principle applies to software testing.

Pareto principle states that 80 percent of all errors uncovered during testing will likely be traceable to 20 percent of all program components.

Testing should begin “in the small “and progress toward testing “in the large.”

Exhaustive testing is not possible.

7.2 LEVEL OF TESTING

There are different levels of testing

->Unit Testing

->Integration Testing

->System Testing

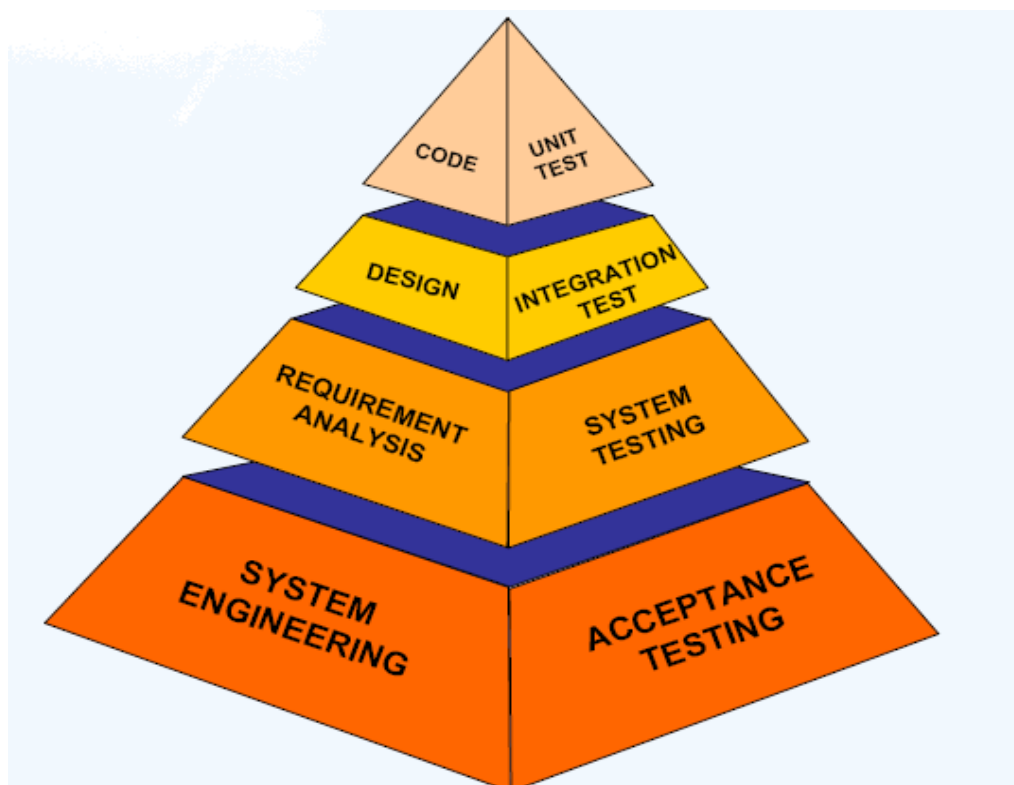


Figure 7.1:Testing pyramid

7.2.1 Unit testing

Unit testing focuses verification effort on the smallest unit of software design, the module. The important control parts are tested to uncover with in the boundary of the module. The module interface is tested to ensure that the information properly flows into and out of the program unit and boundary conditions are tested to ensure that the modules operate properly at boundaries established to limit or restrict processing. Test date is provided through testing screens.

7.2.2 Integration testing

Integrating testing is a systematic technique for constructing Program structure while conducting tests to uncover error associates with interfacing .The objective is to take unit modules and built a program structure that has been directed by design.

- Integration Testing will test whether the modules work well together.
- This will check whether the design is correct.
- Integration can be done in 4 different ways:

7.2.3 System testing

System testing is the process of testing the completed software as a part of the environment it was created for. It is done to ensure that all the requirements specified by the customer are met. System testing involves functional testing and performance testing.

- System Testing will contain the following testing :
 - Functional Testing.
 - Performance Testing.
- Function Testing will test the implementation of the business needs.
- Performance Testing will test the non-functional requirements of the system like the speed, load etc

7.3 SOME IMPORTANT OBSERVATIONS

7.3.1 System Testing and Validation Results.

System testing was done after the system was duly coded. Individual modules of the system were checked to ensure they are fully functional units before the integrating them. This was done by examining each unit; each script was checked to ensure that it functions as required and that it performed exactly as intended. The success of each individual unit gave us the go ahead to carryout integration testing.

The system was validated using a short questionnaire that was filled by representatives of the users who were let to interact with the system using test data and provided feedback about the system features. This was done to assess if the system met their needs and requirements as regards. It was found out that the system performed in conformance to the then defined user needs and requirements. Results of the validation are shown as percentages of respondents against each requirement.

7.3.2 Testing Test Scenarios

1. Check if the page load time is within the acceptable range.
2. Check the page load on slow connections.
3. Check the response time for any action under a light, normal, moderate, and heavy load conditions.
4. Check the performance of database stored procedures and triggers.
5. Check the database execution time.
6. Check for load testing of the application.
7. Check for the Stress testing of the application.
8. Check CPU and memory usage under peak load conditions.

We have checked for scenarios and find that our system performing well in the circumstances.

7.4 TEST CASE RESULT SUMMARY

Test Case #	Description	Result
TC#1	Loading the homepage	Passed
TC#2	Login	Passed
TC#3	Validating	Passed
TC#4	Content	Passed
TC#5	Course page loading	Passed
TC#6	Report page loading	Passed
TC#7	Logout	Passed

CHAPTER 8

CONCLUSION AND FUTURE SCOPE

8.1 CONCLUSION

A software project means a lot of experience. I learned a lot through this project. This project has sharpened our concept cloud computing. It provides easy methods to manage the load of work easily for the users. It is much fast and more efficient as the data once entered can be used and accessed easily. This project has given me an ample opportunity to design, code, test and implements an application. This has helped in putting into practice of various Software Engineering principles concepts like maintaining integrity and consistency of data.

8.2 FUTURE SCOPE

- The Future scope is to make the system more user friendly and enhanced.
- And we will make waste management system to integrate in our existing system.
- I will add Helping BOT in the system.
- Further, the faculty can upload the videos of their plantations on to this application and people can view those videos for the better understanding of our working.

BIBLIOGRAPHY

The books , which are referred and which really helped me in building this system in time, are as follows:-

Books:

MOBILE APPLICATIONS DEVELOPMENT	Taylor & Francis
ANDROID APP-HOOK AND PLUG-IN	Francis Buttle
SYSTEM ANALYSIS &DESIGN	Ellias M. Award
SOFTWARE ENGINEERING	Roger S. Pressman
FUNDAMENTALS OF JAVA PROGRAMMING	MitsunoriOgihara

Web Sites:

[Android Tutorial: http://developer.android.com/](http://developer.android.com/)
<http://www.w3schools.in/>
<http://www.geeksforgeeks.com/>
[Java Tutorial: http://www.w3schools.com/java/](http://www.w3schools.com/java/)
<http://docs.oracle.com/javase/tutorial/>
<http://www.geeksforgeeks.com/>