**A**

**PROJECT REPORT**

**On**

**PRATTLE**

***AN APP FOR YOUR GUFTGU***

**at TECHPILE TECHNOLOGY PVT. LTD., LUCKNOW**



**Submitted Towards Partial Fulfillment of**

**Three-Year diploma in**

# COMPUTER SCIENCE & Engineering

**Under the supervision of**

**Mr. Rahul Soni & Miss Divya Rai**

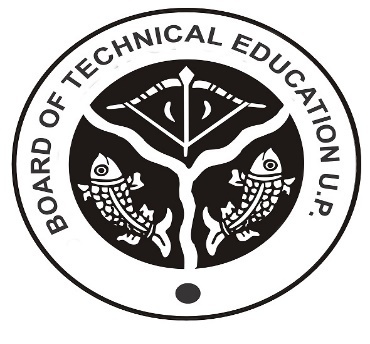
**SUBMITTED TO: SUBMITTED BY:**

**Mr.Vaibhav Kishore ARPAN KHARE**

**Mr. Ajeet Kumar Gautam**

**GOVERNMENT POLYTECHNIC UNNAO**

**Session 2022-2023**



**Enroll: TechpileST22**

**COMPLETION CERTIFICATE**

This is to certify that **ARPAN KHARE** of **DIPLOMA (COMPUTER SCIENCE AND ENGINEERING)** from **GOVERNMENT POLYTECHNIC UNNAO**(Institute/University) was working on the project entitled “**Prattle**” developed on **"Android with Java"** in Techpile Technology Pvt. Ltd. He was engaged with us during **25 July** to **8 September** for a period of **45days.**

He has done an excellent job during his engagement with the Software Development & Testing Division of the company. He has completed his project during the training tenure. His performance has been good and satisfactory.

I would like to take this opportunity to express my appreciation to **ARPAN KHARE** for his work and wish him all the very best for his future endeavors.

**Regards,**

**Divya Rai**

**HR MANAGER**

**Techpile Technology Pvt. Ltd.**

**Lucknow(U.P.)Signature**

# PREFACE

Summer training is an important part of the engineering curriculum. The Diploma course summer training helps a student in getting acquainted with the manner in which his knowledge is being practically used outside his institute and this is normally different from what he has learnt from books. Hence, when the student switches from the process of learning to that of implementing his knowledge,he finds an abrupt change.This is exactly why summer training session during the Diplomacurriculum becomes all the more important. Summer training is prescribed for the student of Technical College as a part of the three-year degree course of engineering by the BTEUP. We are required to undergo summer training for a period of 45 days after the completion of the 1styear.

This training report describes in detail the training after the 2 nd year session, which I completed at the ***Techpile Technology Pvt. Ltd...*** This report also gives the information about the organization and it’s working along with the project undertaken in the trainingperiod.

The fundamental step used in **SDLC** process is based on the ISO 9001 guidelines. My aim was to follow the ISO guidelines and develop a perfect system.

The system development was organized into 5 major parts:

1. **Requirement Gathering**
2. **Documentation/Design**
3. **Development**
4. **Coding**
5. **Testing**

**ACKNOWLEDGEMENT**

Apart from my effort, the success of the project depends largely on the encouragement and guidelines of many others. We take this opportunity to express our gratitude to the people who express have been instrumental in the successful completion of this project.

I would like to express my deep and sincere gratitude to my supervisor Mr. **Rahul Soni** Sir **(**Techpile Technology Pvt. Ltd.)***,***who gave me his full support and encouraged me to work in an innovative and challenging project for Educational field. His wide knowledge and logical thinking gave me right direction all the time.

I am deeply grateful my project coordinator for his help and support provided at every step of the project.

Last but not the least, I thank to all employees of **Techpile Technology Pvt. Ltd.** for their support andco-operation.

***ARPAN KHARE***

**DECLARATION**

This is to certify that the project report entitled “**PRATTLE**” is done by me is an authentic work carried out for the partial fulfillment of the requirements for the award of the Diploma in **“(COMPUTER SCIENCE AND ENGINEERING)”**under the guidance of Mr.**Rahul Soni**. The matter embodied in this project work has not been submitted earlier for award of any degree or diploma to the best of my knowledge and belief.

`

**ARPAN KHARE**

**INDEX**

**Page**

1. **Introduction………………………………..…………..................................10-12**
   1. Overview of Organization…………………………………………………………….…….10-10
   2. Project Introduction……………………………………………………………….………...10-10
   3. Objectives…………………………………………………………………………...........…11-11
   4. Problem Definition……………………………………….……………………………..…..11-12
2. **System Analysis……………………………….…………………………… 13-21**
   1. Objective……………………………………………………………………………..….….13
   2. SDLC Phases…………………………………………………………………....…….….…13
      1. Preliminary Investigation……………………………………………………….………..…13-15
      2. System Analysis………………………………………………………………….…………15
      3. System Design…………………………………………………………........................…… 15
      4. Coding……………………………………………………………...…………….…………15
      5. Testing………………………………………………………………………….....……..….15
      6. Implementation………………………………………………..……………........................15-16
      7. Maintenance…………………………………………………………………………....…...16
   3. Process Description………………………………………………………….……………...16-17
   4. Project Model Used…………………………………………………………….……..…….17-18
   5. ER-Diagram…………………………………………………………………………..…….18-19
   6. Data Flow Diagram……………………………………………………….…………..…….20-21

1. **Software Hardware Requirement Specification……………..………..… 22-23**
   1. Hardware Requirement…………………………………………………………………..… 22
   2. Serverside Software Requirement………………………………………………………… 22
   3. Client sideSoftware Requirement………………………………..…….………………….. 22-23
   4. Support Maintenance………………………………………………….…….……………... 23
2. **System Design Approach……………………………………..……………. 24**
   1. Top-Down Designing……………………………………………………………….…..…. 24
   2. Bottom –Up Designing…………………………………………………………..………… 24
   3. Following Approach……………………………………………………………..………… 24
3. **Configuration Design (**manifest.xml**)…………………................................25-31**
4. **Data Modeling…………………………………………………………...… 32-33**
   1. List of Tables
   2. Structure of Tables
5. **Testing…………………………………………………….. ………………. 34-37**
6. **Input-Output Forms……………………………………………………… 38-54**
   1. Project Screenshot
   2. Project Coding (XML & Java)
7. **Future Scope…………………………………………….………………….. 55**
8. **Conclusion………………………………………………………………….. 55**

**LIST OF TABLES**

**Page**

1. **Screenshot of Firebase Realtime Database…………………................................33**
2. **MessageTable.………………………………………………………...………33**
3. **StoryTable……………………………………………….................................34**
4. **UserTable…………………………………………………................................34**

**LIST OF FIGURES & SCREENSHOTS**

**Page**

1. **SOFTWARE DEVELOPMENT LIFE CYCLE ……………………………….16**
2. **DEVELOPMENT PHASES ……………………………………………………..18**
3. **E-R DIAGRAM …………………………………………………………...…..18-19**
4. **ZERO LEVEL DATA FLOW DIAGRAM …………………………………….21**
5. **ONE LEVEL DATA FLOW DIAGRAM……………………………………….21**
6. **TOP-DOWN DESIGNING……………………………………………………….24**
7. **BOTTOM-UP DESIGNING…………………………………...............................24**
   1. **INTROUCTION**
   2. **Overview of Organization**

Techpile is founded by some young engineers who have mastered the IT sector, whose objective is to achieve the highest position in the IT sector across the country, who are trying to achieve this objective by cooperating in various fields.

Techpile is an organization working in both software development and software training, which aims to make all the specials of its client successful through their coding as well as to make students more competent to work with a well reputed organization.

We are proud of our high-quality standards. These standards allow us to provide our customers with reliable and error-free software applications, regardless of complexity. Our top-notch developers use the latest software methodologies and technologies. This means that they can concentrate on our clients' business goals and keep them involved in every stage through the entire project. Our meticulous approach has helped us build our excellent track record with no failed or aborted projects. We are in the business of change, managing complexity with an unparalleled insight, looking beyond the horizon of IT with resources focused on solutions. Becoming successful is a skill but one cannot perfect it without practice.

* 1. **PROJECT INTRODUCTION:--**

TPTChat rooms are areas in which people can gather to engage in real-time conversations, generally using text-based communication.

The chat room works as a virtual room, where groups of people send messages that others can read instantaneously.

* 1. **OBJECTIVE**
* It is time saving and energy saving.
* Person can easily login to site while sitting on home.
* Person can buy more than one product without going anywhere.
* The transactions are executed in off-line mode, hence on-line data for Shopping, Internet capture and modification is not possible.
* Manage the information of Internet
* Shows the information and description of the Shopping.
* To increase efficiency of managing the Shopping
* It deals with monitoring the information and transactions of Bills.
  1. **PROBLEM DEFINITION**

In this section we shall discuss the limitation and drawback of the existing system that forced us to take up this project. Really that work was very typical to manage the daily errors free records and adding or removing any node from server. This problem produces a need to change the existing system. Some of these shortcomings are being discussed below: -

* **Low Functionality**

With the existing system, the biggest problem was the low functionality. The problem faced hampered the work. For small task like adding any new node to server or deleting a node or keeping daily record we have to appoint minimum two or three employee.

* **Erroneous Input and Output**

In the existing system, humans performed all the tasks. As in the human tendency, error is also a possibility. Therefore, the inputs entered by the person who is working in the Company, in the registers may not be absolutely foolproof and may be erroneous. As a result of wrong input, the output reports etc. Will also be wrong which would in turn affect the performance.

* **Portability Problem**

System that existed previously was manual. As a result, the system was less portable. One has to carry the loads of many registers to take the data from one place to another. A big problem was that the system was less flexible and if we wanted to calculate yearly or monthly maintenance report or efficiency report, then it was a big headache.

* **Security-**

Security concerns were also one of the motives of the Company for the need of software. In the registers, the data is not secure as anybody can tamper with the data written in the registers. While in this software, just a password makes it absolutely secure from the reach of unauthorized persons.

* **Data Redundancy**

In the case of manual system, the registers are maintained in which, a lot of data is written.

* **Processing Speed**

In manual system maintaining a register and performing the necessary calculation has proved to be a troublesome job, which takes a lot of time and may affect the performance of the Company. But with this software we can have all the tasks performed in a fraction of second by a single click thus making the troublesome job much easier.

* **Manual Errors**

When a number of tough tasks are prepared by the humans like preparation of reports, performing long calculation then some human error are obvious due to a number of factors like mental strain, tiredness etc. But as we all know that computer never get tired irrespective of the amount of work it has to do. So this software can nullify the probability of manual error that improve the performance.

* **Complexity in Work**

In manual system whenever a record is to be updated or to be deleted a lot of cutting and overwriting needs to be done on the registers that are concerned that are deleted or updated record, which makes the work very complex.

1. **SYSTEM ANALYSIS**
   1. **Objective:**

It is a process of collecting and interpreting facts, identifying the problems, and decomposition of a system into its components.

System analysis is conducted for the purpose of studying a system or its parts in order to identify its objectives. It is a problem solving technique that improves the system and ensures that all the components of the system work efficiently to accomplish their purpose.

**Analysis specifies what the system should do.**

* 1. **SDLC Phases:**

System Development Life Cycle (SDLC) mainly consists of the following 7 phases which can be detailed: -

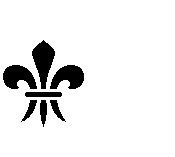
* + 1. **Preliminary Investigation: -**

This is the first phase of the system development life cycle. In this phase we tend to find out the needs of the client –what exactly does the client want? Before the development of any system the important point is to know the needs, objectives and scope of the system.

* **Feasibility Study**: -

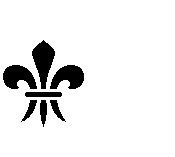
Feasibility study is the step of preliminary study of the system development life cycle. Things are always easy at the beginning in any software process. In fact nothing is in feasible with unlimited time and resources. But it is not the fact. So, practically we have to do in limited resources in a restricted time margin. So for the system to be feasible, following points we have to consider.

The feasibility study is conducted to check whether the candidate system is feasible. The system which is selected to be the best against the criteria is there after designed and developed. The feasibility study takes in to consideration, the risks involved in the project development beforehand. Therefore in this phase we have to do feasibility study which is the test of the website according to its work ability, impact on the organization, ability to meet user need and effective use of resources. We do the feasibility study for website to analyze the risks, costs and benefits relating to economics, technology and user organization. There are several types of feasibility depending on the aspect they cover. Import of theseincludes:

** Technical Feasibility:**

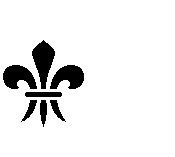
This is an important outcome of preliminary investigation. It comprise of following questions:-

* Can the work of project bed one with the current equipment, existing software and available man powerresource?
* If Technology is required what are the possibilities that it can bedeveloped?

**Economic Feasibility:**

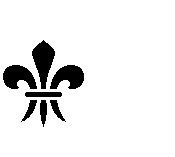
It deals with question related to the economy. It comprise of the following questions:-

* Are there sufficient benefits in creating the system to make the costacceptable?
* Are the costs of not creating the system so great that the project must beundertaken?

**Legal Feasibility:**

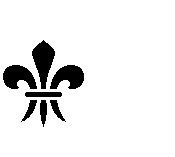
It deals with the question related to the legal issues. It comprise of the followingquestions: -

* ContractSigning
* Software License agreement
* Issues related to cyber laws.
* Legal issues relating to the man power contract.

**Operational Feasibility:**

The operational feasibility consists of the following activity:

* Will the system be useful if it is developed &implemented?
* Will there be resistance from employee?

**Social & Behavioral Feasibility:**

It deals with the various issues related to the human behavior like: -

* Whether the user be able to adapt a new change or not?
* Whether the ambiance we are providing suits the user or not?
* **Request Approval: -**

Request approval is the preliminary investigation phase of system development lifecycle.Request approval is the phase in which all the requirements which would be provide in the system are stated. The request approval is a sort of agreement between the client and the company which is building this software.

Both the parties should be mutually agreed on the statedrequirements.

* + 1. **System Analysis:**-

System analysis is the phase following the phase of the request approval. In this phase we tend to analyze the overall system which we have to build. System analysis is the crucial part in SDLC.

* + 1. **System Design:**-

System design means the designing of the system. The System can be done in either of the following two ways:-

* + Logical SystemDesign
  + Physical SystemDesign
    1. **Coding:**-

Coding is the phase in which a developer codes using any programming languages. Coding constitutes only 20% of the whole project and which is easier to write. The coding work is also done in the teams; development of the system is usually done under the modular programming style, which can be either top-down approach or bottom-up approach.

* + 1. **Testing:**-

Testing is the phase in which the system that has been developed is tested. Testing comprises of the 60% of the overall development of the system. Testing of the system is important because testing aims to uncover the different errors in the system. There are various different testing techniques that can be used for the testing of the system.

* + 1. **Implementation:-**

Implementation process involved the installation of software on user’s side. Implementation process actually depends on type of a system & various. Opting for suitable conversion approach is a step implementation. The conversion processes are as follows:-

* + - ParallelConversion
    - Direct ConversionApproach
    - Pilot ConversionApproach
    - Phase In ConversionApproach
    1. **Maintenance**: -

Merely developing the system is not important but also maintenance is important. The company that has built the system provides for some time free of cost maintenance to the client and after that period it is usually a paid service.



* 1. **Process Description**

Gantt charts mainly used to allocate resources to activities. The resources allocated to activities include staff, hardware, and software. Gantt charts (named after its developer Henry Gantt) are useful for resource planning. A Gantt chart is special type of bar chart where each bar represents an activity. The bars are drawn along a timeline. The length of each bar is proportional to the duration of the time planned for the corresponding activity.

Gantt chart is a project scheduling technique. Progress can be represented easily in a Gantt chart, by coloring each milestone when completed. The project will start in the month of January and end after 4 months at the beginning of April.

* 1. **PROJECT MODEL USED**

**Iterative Enhancement Model**

* This model has the same phases as the waterfall model, but with fewer restrictions. Generally the phases occur in the same order as in the waterfall model, but they may be conducted in several cycles.
* Useable product is released at the end of the each cycle, with each release providing additional functionality. Customers and developers specify as many requirements as possible and prepare a SRS document. Developers and customers then prioritize these requirements. Developers implement the specified requirements in one or more cycles of design, implementation and test based on the definedpriorities.

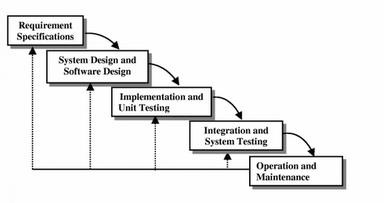
The procedure itself consists of the initialization step, the iteration step, and the Project Control List. The initialization step creates a base version of the system. The goal for this initial implementation is to create a product to which the user can react. It should offer a sampling of the key aspects of the problem and provide a solution that is simple enough to understand and implement easily. To guide the iteration process, a project control list is created that contains a record of all tasks that need to be performed. It includes such items as new features to be implemented and areas of redesign of the existing solution. The control list is constantly being revised as a result of the analysisphase.

The iteration involves the redesign and implementation of iteration is to be simple, straightforward, and modular, supporting redesign at that stage or as a task added to the project control list. The level of design detail is not dictated by the iterative approach. In a light-weight iterative project the code may represent the major source of [documentation](http://en.wikipedia.org/wiki/Software_documentation)of the system; however, in a critical iterative project a formal [Software Design Document](http://en.wikipedia.org/wiki/Software_Design_Document)may be used. The analysis of iteration is based upon user feedback, and the program analysis facilities available. It involves analysisof the structure, modularity, [usability](http://en.wikipedia.org/wiki/Usability), reliability, efficiency, & achievement of goals. The project control list is modified in light of the analysis results.

**PHASES:**

Incremental development slices the system functionality into increments (portions). In each increment, a slice of functionality is delivered through cross- discipline work, from the requirements to the deployment. The unified process groups increments/iterations into phases: inception, elaboration, construction, and transition.

* Inception identifies project scope, requirements (functional and non-functional) and risks at a high level but in enough detail that work can be estimated.
* Elaboration delivers a working architecture that mitigates the top risks and fulfills the non-functional requirements.
* Construction incrementally fills-in the architecture with production-ready code produced from analysis, design, implementation, and testing of the functional requirements.
* Transition delivers the system into the production operating environment.



* 1. **ER-Diagram**

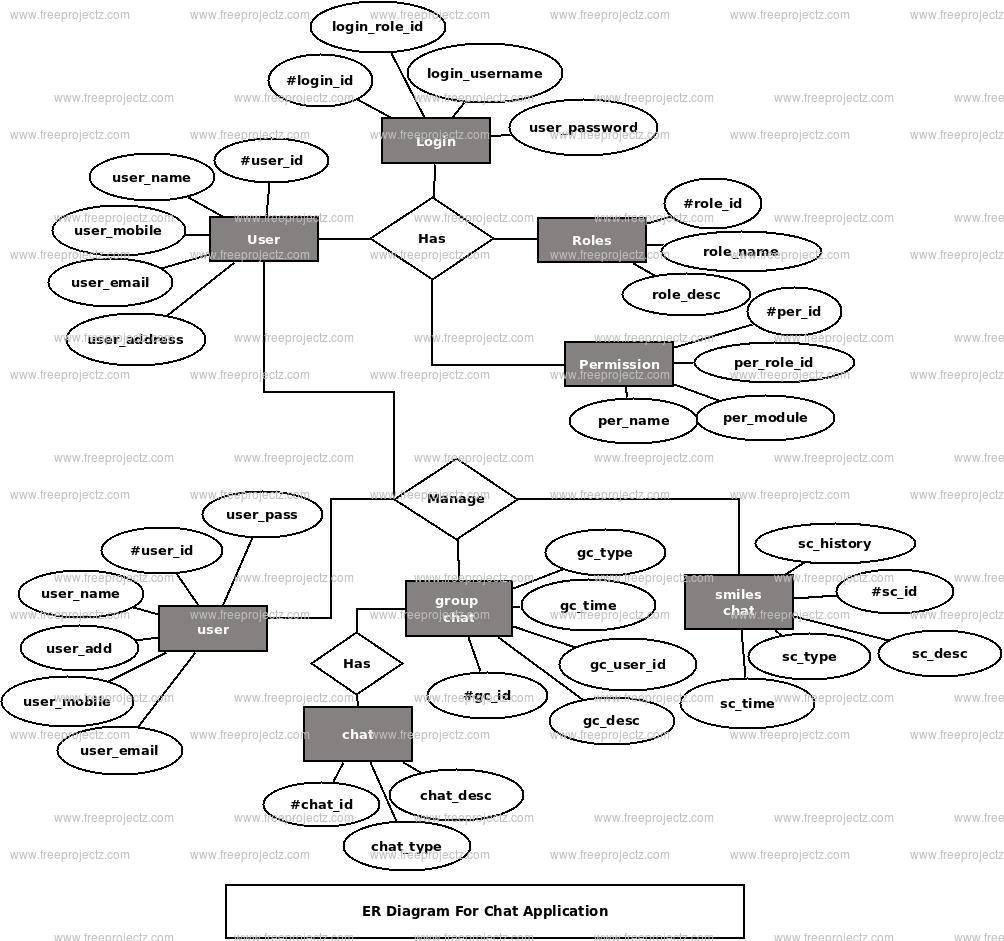
**Introduction: -**

I[n software engineering, an](http://en.wikipedia.org/wiki/Software_engineering) entity-relationship model (ERM) is an abstract and conceptual representation of [data.](http://en.wikipedia.org/wiki/Data) Entity-relationship modeling is a [database modelingmethod,](http://en.wikipedia.org/wiki/Database_model) used to produce a type [of conceptual schema](http://en.wikipedia.org/wiki/Conceptual_schema) o[r semantic data model of](http://en.wikipedia.org/wiki/Semantic_data_model) a system, often [a relational database, a](http://en.wikipedia.org/wiki/Relational_database)nd its require[ments in a top-down](http://en.wikipedia.org/wiki/Top-down) fashion. Diagrams created by this process are called entity-relationship diagrams, ER diagrams, or ERDs. ER Diagrams depicts relationship between data objects. The attribute of each data objects noted in the entity-relationship diagram can be described using a data object description. Entity relationship diagram is very basic, conceptual model of data and it is fundamental to the physical database design. This analysis is then used to organize data as relations, normalizing relations, and obtaining a Relationaldatabase.

The entity-relationship model for data uses three features to describe data. Theseare:

1. Entities which specify distinct real-world items in anapplication.
2. Relationship, which connect entities and represent meaningful dependencies betweenthem.
3. Attributes which specify properties of entities &relationships.

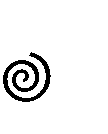
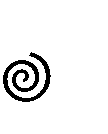
**E-R Diagram**



* 1. **Data Flow Diagram**

**Introduction:**-

DFD is anacronym for theword Data FlowDiagram. DFD is a pictorial representation of thesystem. DFD is agraphical representation ofthe flowof data throughthe information system. DFD are also used for the visualization of data processing (structured design). ADFD provides no information about the timings of the process, or about whether process will operate in parallel or sequence. DFD is an important technique for modeling system’s high-level detail by showing how input datais transformed to output results through a sequence of functional transformations. DFD reveal relationships among between the various components in a program or system. The strength of DFD lies in the fact that using few symbols we are able to express program design in an easiermanner. ADFD can beusedto represent thefollowing:-

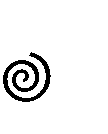
External Entity sending and receiving data. Process that change the data.

Flow of data within the system. Data Storage locations.

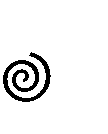
**Uses of DFD:-**

Themain uses of data flow diagrams are as follows: -

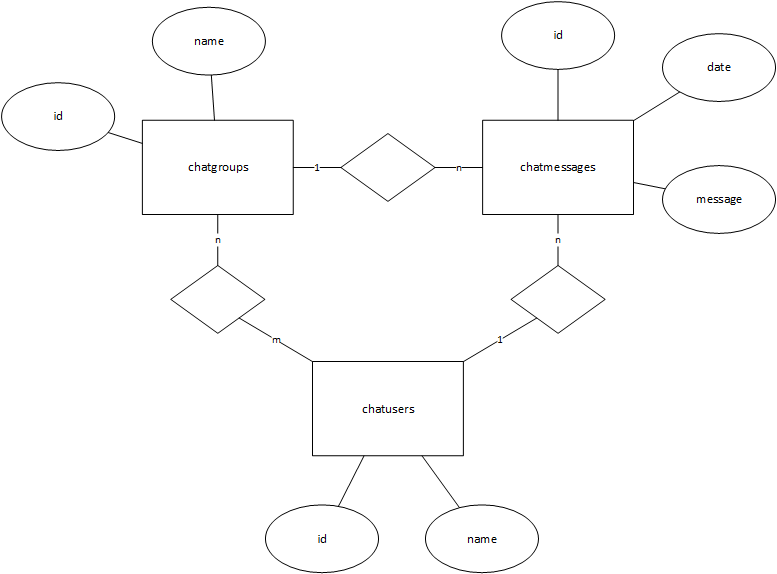
DFD is amethod ofchoice forrepresentation of showingof information througha system because of the followingreasons:-

DFDs areeasierto understand bytechnical and non-technicalaudiences.

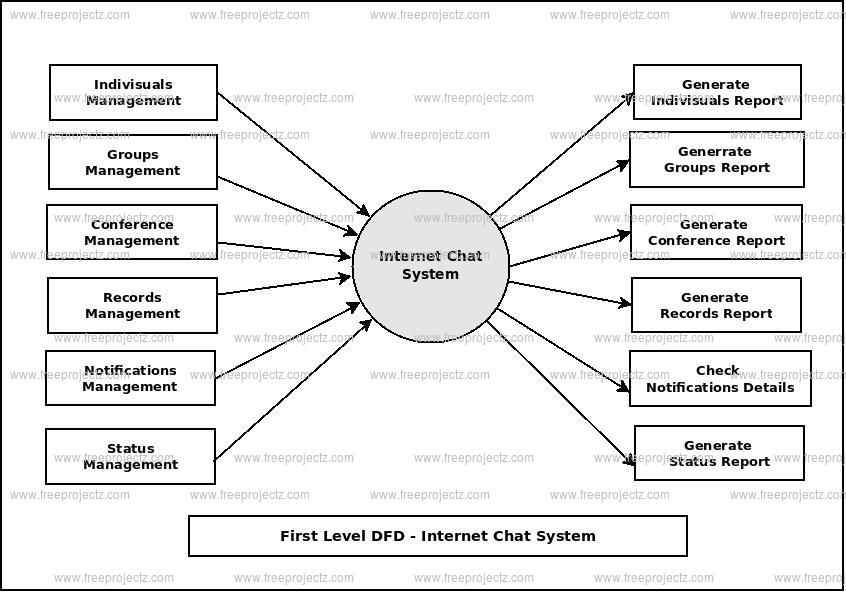
DFDs can provide a high level system overview, complete with boundaries and connections toother system.

DFDs can provideadetailed representation of system components.

**0.Level DFD**



1. **Level DFD:**



1. **SOFTWARE HARWARE REQUIREMENT SPECIFICATION**

A requirements specification for a software system is a complete description of the behavior of a system to be developed and it includes a set of use cases that describe all the interactions the users will have with the software. In addition to use cases, the SRS also contains non-functional requirements.

Non-functional requirements are requirements which impose constraints on the design or implementation (such as performance engineering requirements, quality standards, or design constraints).Requirements are a sub-field of software engineering that deals with the elicitation, analysis, specification, and validation of requirements for software.

The software requirement specification document enlists all necessary requirements for project development. To derive the requirements we need to have clear and thorough understanding of the products to be developed. This is prepared after detailed communications with project team and the customer.

* 1. **Hardware Requirement:**
* Microsoft Windows 7/8/10 (32 or 64 bit)
* 2 GB RAM minimum, 8 GB recommended
* 2 GB of available disk space minimum, 4 GB recommended (500 MB for IDE + 1.5 for Android SDK and emulator system image)
* 1280 x 800 minimum screen resolution
* Graphics - none (Integrated graphics 64MB)
* 250GB SSD (the main reason for a laptop to run as fast as light)
  1. **Server side Software Requirement:**
* Java Development Kit (JDK) 8
* Android Studio | 2020.3.1
* Firebase Database
  1. **Client side Hardware requirements**:
* A fast speed USB cable
* Android Phone for Testing (Version 11 / 10 / 9)

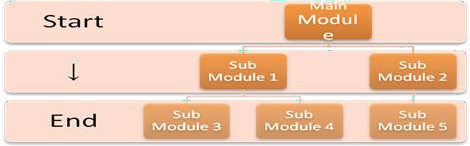
**To develop this project the various Software resources are used:**

* Front End                       :  XML
* Back End                        :  Firebase Browser Based
* Technology                     : Android
* Code-Behind Language   :  Java
* IDE                                 :  Android Studio
  1. **SUPPORT AND MAINTENANCE:-**

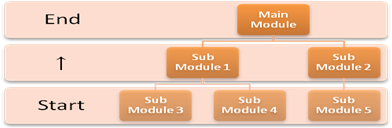
One year free support for rectifying system bugs including front end and beck end will be provided. During warranty period Software Engineers will be responsible for removing bugs and improving it. After one year support can be extended @ 20% of the total product deployment cost.

1. **SYSTEM DESIGN APPROACH**
   1. ***Top – Down designing*:**

The top - down designing approach started with major components of the system. It is a stepwise refinement which starts from an abstract design, in each steps the design is refined two or more concrete levels until we reach a level where no – more refinement is possible or not needed.

* 1.  ***Bottom – Up designing:***

In bottom – up designing the most basic and primitive components are designed first, and we proceed to higher level components. We work with layers of abstractions and abstraction are implemented until the stage is reached where the operations supported by the layer is complete.

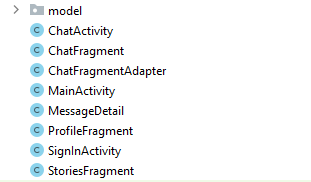


* 1. **Following Approach:**

In this project we are following **Mixed Approach** i.e. A combination of top – down and bottom – up. We are developing some of the components using top – down designing approach (e.g. the WebPages) and the some components in bottom – up designing approach (e.g. the middle tier classes).

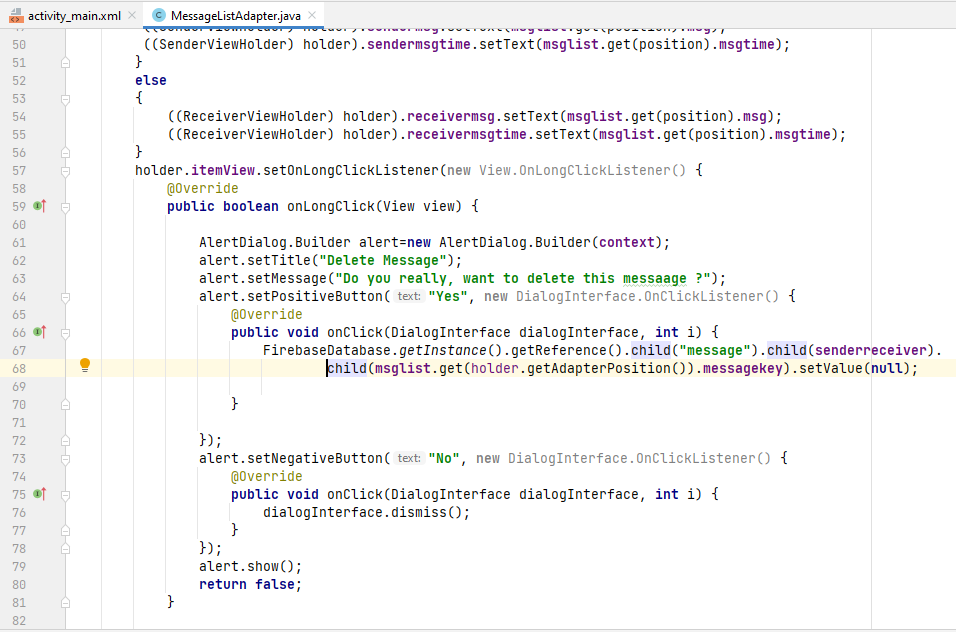
1. **CONFIGURATION DESIGN(**manifest.xml**)**

**Models**



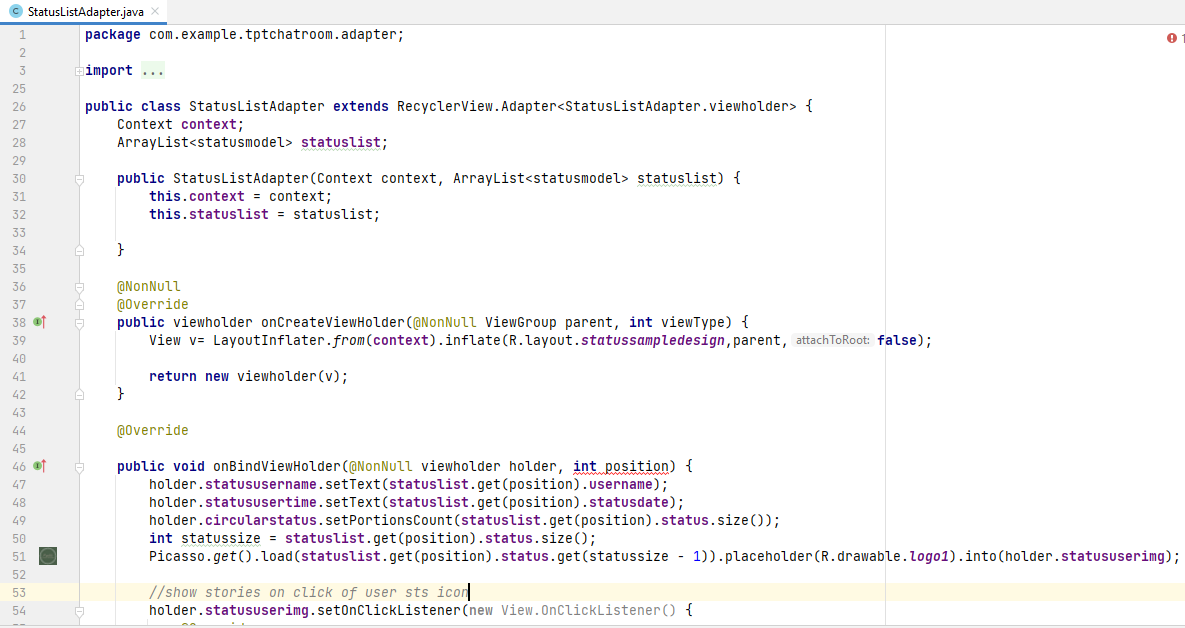
**MessageList Adapter.java**

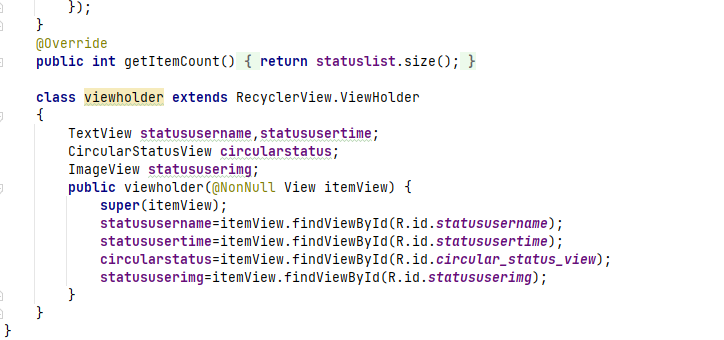
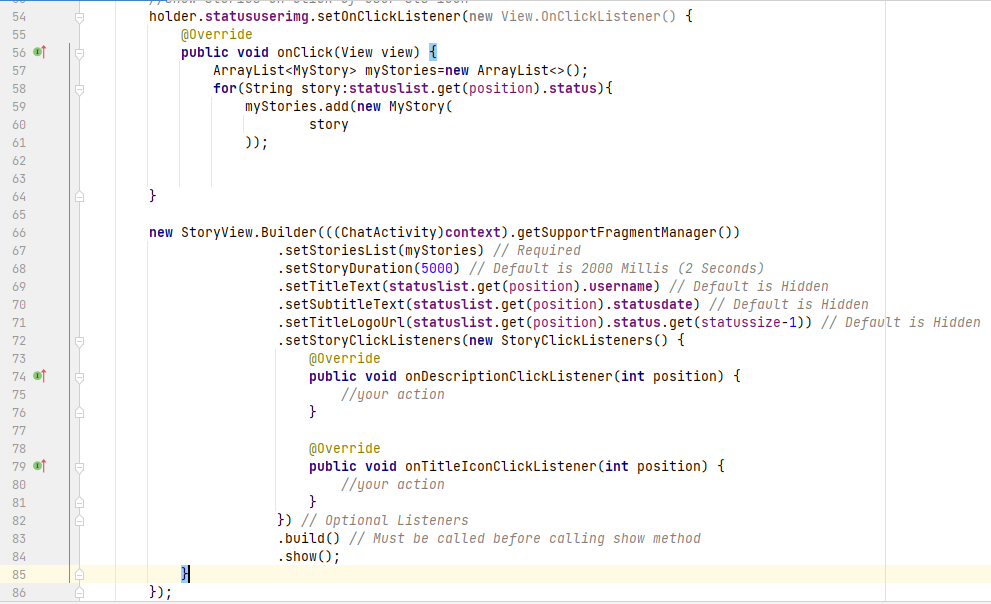
****

****

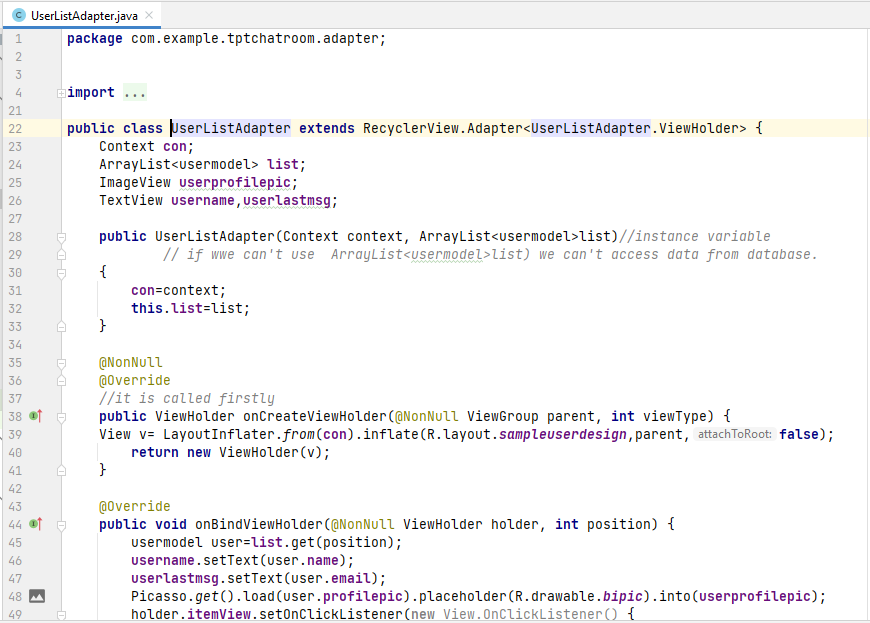
****

**StatusListAdapter.java**

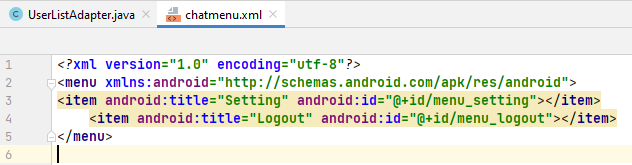
****

****

**UserList.Adapter.java**

****

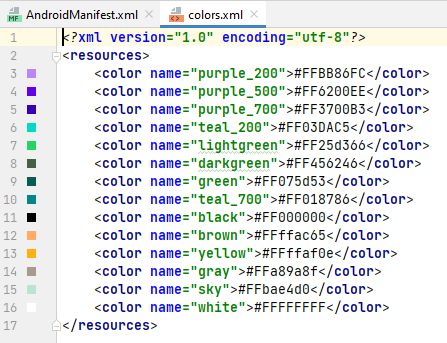
**Chat menu.xml**

****

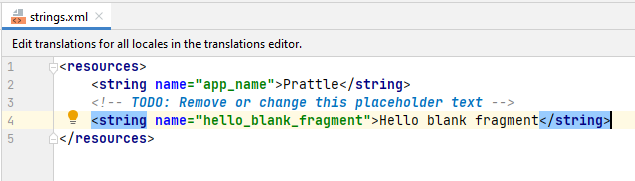
**Manifest.xml**

****

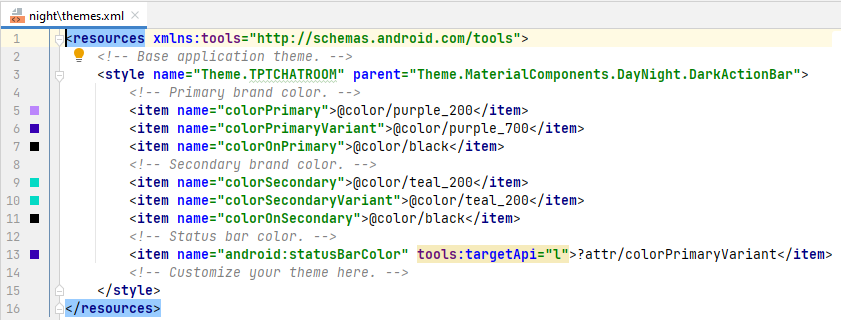
**Color.xml**

****

**Strings.xml**

****

**Themes.xml**

****

1. **DATA MODELING**

**6.1 List of Table**

**1. RealTime Database**

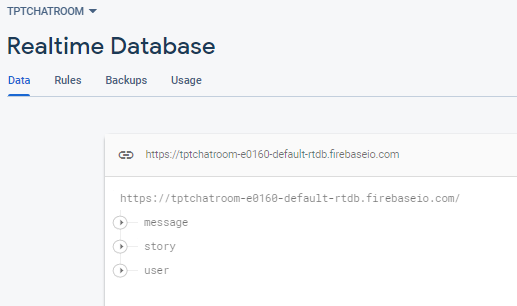
**2. Message Table**

**3. Story Table**

**4. User Table**

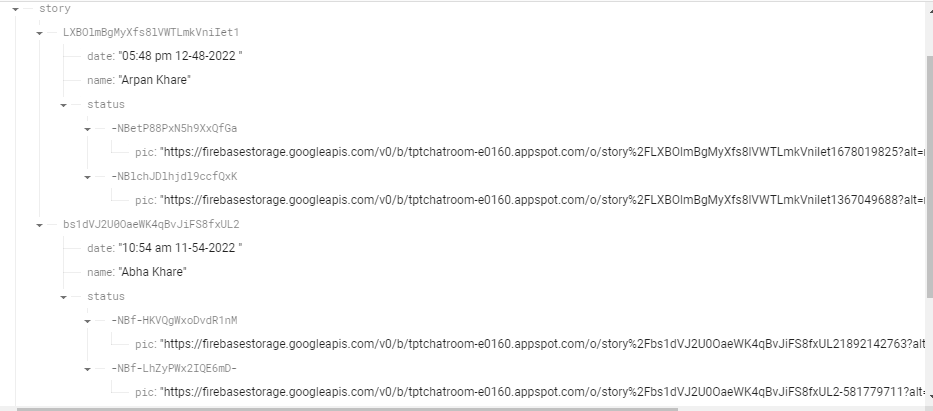
**6.2 Structure of Table**

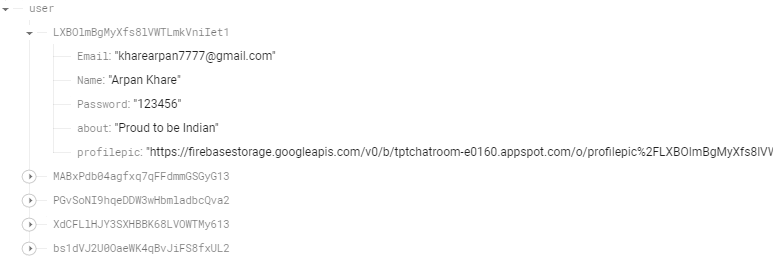
6.2.1 SCREENSHOT OF FIREBASE REALTIME DATABASE



6.2.2 MessageTable

6.2.3 StoryTable



6.2.4 UserTable

1. **TESTING**

Testing is the integral part of any System Development Life Cycle insufficient and interested application tends to crash and result in loss of economic and manpower investment besides user’s dissatisfaction and downfall of reputation.

“Software Testing can be looked upon as one among much process, an organization performs, and that provides the last 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 the plan that all aspect of system .Complements, Correctness, Reliability and Maintainability.

Software is to be tested for the best quality assurance, an assurance that system meets the specification and requirement for its intended use and performance.

System Testing is the most useful practical process of executing the program with the implicit intention of finding errors that makes the programfail.

**Types of Testing:**

**Black Box (Functional) Testing:**

Testing against specification of system or component. Study it by examining its inputs and related outputs. Key is to devise inputs that have a higher likelihood of causing outputs that reveal the presence of defects. Use experience and knowledge of domain to identify such test cases. Failing this a systematic approach may be necessary. Equivalence partitioning is where the input to a program falls into a number of classes,

e.g. positive numbers vs. negative numbers. Programs normally behave the same way for each member of a class. Partitions exist for both input and output. Partitions may be discrete or overlap. Invalid data (i.e. outside the normal partitions) is one or more partitions that should be tested.

Internal System design is not considered in this type of testing. Tests are based on requirements andfunctionality.

This type of test case design method focuses on the functional requirements of the software, ignoring the control structure of the program. Black box testing attempts to find errors in the following categories:

* + Incorrect or missing functions.
  + Interface errors.
  + Errors in data structures or external database access.
  + Performance errors.
  + Initialization and termination errors.

**White Box (Structural) Testing:**

Testing based on knowledge of the structure of component (e.g. by looking at source code). The advantage is that structure of the code can be used to find out how many test cases need to be performed. Knowledge of the algorithm (examination of the code) can be used to identify the equivalence partitions. Path testing is where the tester aims to exercise every independent execution path through the component. All conditional statements were tested for both true and false cases.

program units than large ones. Flow graphs are a pictorial representation of the paths of control through a program (ignoring assignments, procedure calls and I/O statements). Use flow graph to design test cases that execute each path. Static tools may be used to make this easier in programs that have a complex branching structure. Tools support. Dynamic program analysers instrument a program with additional code. Typically this will count how many times each statement is executed. At end print out report showing which statements have and have not been executed. Problems with flow graph derived testing:

* Data complexity could not take intoaccount.
* We cannot test all paths incombination.
* In really only possible at unit and module testing stages because beyond that complexity is toohigh.

This testing is based on knowledge of the internal logic of an application’s code. Also known as a Glass Box Testing .Internal software and code working should be known for this type of testing. Tests are based on coverage of code statements, branches, paths, conditions.

**Unit Testing:**

Unit testing concentrates on each unit of the software as implemented in the code. This is done to check syntax and logical errors in programs. At this stage, the test focuses on each module individually, assuring that it functions properly as a unit. In our case, we used extensive white-box testing at the unit testing stage.

A developer and his team typically do the unit testing do the unit testing is done in parallel with coding; it includes testing each function and procedure.

**Incremental Integration Testing:**

Bottom up approach for testing i.e. continuous testing of an application as new functionality is

added; Application functionality and modules should be independent enough to test separately done

by programmers or by testers.

**Integration Testing:**

Testing of integration modules to verify combined functionality after integration

.Modules are typically code modules, individual applications, client and server and distributed systems.

**FunctionalTesting:**

This type of testing ignores the internal parts and focus on the output is as per requirement or not .Black box type testing geared to functionality requirements of an application.

**System Testing:**

Entire system is tested as per the requirements. Black box type test that is based on overall requirement specifications covers all combined parts of a system.

**End-to-End Testing:**

Similar to system testing ,involves testing of a complete application environment in a situation that mimics real-world use, such as interacting with a database ,using network communications, or interacting with hardware, applications, or system if appropriate.

**Regression Testing:**

Testing the application as a whole for the modification in any module or

functionality.Difficult to cover all the system in regression testing so typically automation tools are used for these testing types.

**Acceptance Testing:**

Normally this type of testing is done to verify if system meets the customer specified requirements. User or customers do this testing to determine whether to accept application.

**Performance Testing:**

Term often used interchangeably with “stress” and “load” testing, To check whether system meets performance requirements, Used different performance and load tools to do this.

**Alpha Testing:**

In house virtual user environment can be created for this type of testing. Testing is done at the end of development .Still minor design changes may be made as a result of such testing.

**Beta Testing:**

Testing typically done by end-users or others. This is final testing before releasing application for commercial purpose.

# Input Output Forms

# 8.1 Project screenshot

# Sign in Page Sign up Page

# WhatsApp Image 2022-11-09 at 6.25.28 AM.jpeg WhatsApp Image 2022-11-09 at 6.25.27 AM (2).jpeg

# Chat Page or Home Page Chatting Page

# WhatsApp Image 2022-11-09 at 6.25.27 AM (1).jpeg WhatsApp Image 2022-11-09 at 6.25.27 AM.jpeg

# Stories Page Profile Page WhatsApp Image 2022-11-09 at 6.25.27 AM.jpeg WhatsApp Image 2022-11-09 at 6.25.25 AM.jpeg

# 8.2Project Coding (XML & Java)

# activity\_chat.xml

# ac1.PNGac2.PNG

# ChatActivity.java

# ca java 1.PNGca java 2.PNGca java 3.PNG

# activity\_main.xml

# am1.PNGam 2.PNGam 3.PNGam 4.PNG

# MainActivity.java

# ma java 1.PNG ma java 2.PNG

# Activity\_sign\_in.xml

# as 1.PNGas 2.PNGas 3.PNGas 4.PNG

# SignInActivity.java

# sa 1.PNGsa 2.PNG

# Fragment\_stories.xml

# sa 1.PNG

# sa 2.PNG

# StoriesFragment.java

# sf 1.PNGsf 2.PNGsf 3.PNGsf 4.PNG

# Activity\_message\_detail.xml

# amd 1.PNGamd 2.PNGamd 3.PNG

# MessageDetail.java

# md 1.PNGmd 2.PNGmd 3.PNGmd 4.PNG

## FUTURE SCOPE

Following modification or upgrades can be done in system.

* 1. More than one company can be integrated through this software.
  2. Web services can be used to know exact donation status of packets.
  3. Client can check there donation delivery status online.

1. **CONCLUSION:**

At the last the Conclusion of project is to develop a mobile-application which the help programmer to get help from the site, so that they can develop their project and application Different Technologies and make a group of programmer. A **Group**is a social unit of any size that shares common values, ideas and code and queries. The portal doesn’t have to be expensive. It supports multiple programmer goals.

**PRATTLE**is not only a App portal; it is a live product of board of technical education. In future we will add more and more features on it.