

# MAJOR PROJECT REPORT

***iREQ***

*Submitted By*

***AMRITHA K (12133642)***

***ANJANA B (12133647)***

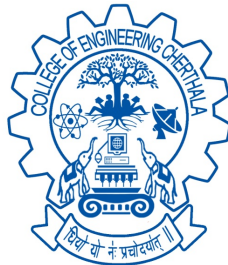
***BIBINRAJ B S (12133613)***

***HARIVISHNU R (12133615)***

***VISHNUJITH N (12133635)***

*Under the guidance of*

***Mrs. MEERA M***



**March 2016**

**Department of Computer Science and Engineering**

**College of Engineering, Cherthala**

**Pallippuram P O, Alappuzha-688541**

**Phone: 0478 2553416, Fax: 0478 2552714**

**<http://www.cectl.ac.in>**

# MAJOR PROJECT REPORT

## ***iREQ***

*Submitted By*

**AMRITHA K (12133642)**

**ANJANA B (12133647)**

**BIBINRAJ B S (12133613)**

**HARIVISHNU R (12133615)**

**VISHNUJITH N (12133635)**

*Under the guidance of*

***Mrs. MEERA M***

*In partial fulfilment of the requirements for the award of the degree  
of*

*Bachelor of Technology*

*in*

*Computer Science and Engineering*

*of*

*Cochin University Of Science And Technology*



**March 2016**

**Department of Computer Science and Engineering**

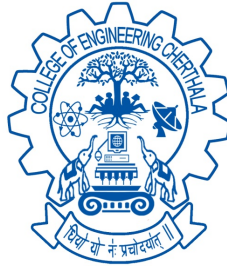
**College of Engineering, Cherthala**

**Pallippuram P O, Alappuzha-688541**

**Phone: 0478 2553416, Fax: 0478 2552714**

**<http://www.cectl.ac.in>**

**DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING**  
**COLLEGE OF ENGINEERING CHERTHALA**  
**ALAPPUZHA-688541**



**C E R T I F I C A T E**

This is to certify that, the Major Project Report submitted by

Name :

Date: 23-03-2016

Reg.No :

***iREQ***

In partial fulfilment of the requirement of course of the Bachelor of Technology, (B. Tech) in **Computer Science & Engineering** prescribed by **Cochin University of Science & Technology** during the academic year 2015-2016.

**Guide**

**Co-ordinator**

**HoD**

**Mrs. Meera M**

Assistant Professor

Computer Science&Engg

**Ms. Asha Thomas**

Assistant Professor

Computer Science&Engg

**Mr. Muhammed Ilyas H**

Assistant Professor

Computer Science&Engg

## ACKNOWLEDGEMENT

We take this opportunity to express our sincere gratitude to the people who have been instrumental in the successful completion of our major project.

For every endeavour in our life, the help of the Lord has always followed. We have yet again experienced his loving kindness while preparing for this project.

We would like to express our sincere thanks to the Principal, **Dr. Mini M. G**, for her valuable support and advice. We would also like to thank our Head Of Computer Science Department, **Mr. Muhammed Ilyas H** for his support. We express our heartfelt gratitude to our project coordinator, **Ms. Asha Thomas** (Assistant Professor, Department of Computer Science) for her valuable help and support. We would also like to thank our project Guide, **Mrs. Meera M** (Assistant Professor, Department of Computer Science) for her support and guidance.

We are indebted to all teaching and non-teaching staff of the Department of Computer Science & Engineering, friends and family for their co-operation and support, with out which we could never have completed the project this well.

## **ABSTRACT**

The usual method of document handling requires authorization from many officials before the application is finally processed. This project aims at replacing the traditional method of transferring the files manually for authorization from each concerned official which often results in delayed processing of documents. This project helps in optimizing or automating this procedure. The system relieves the applicant from having to pay daily visits to office to know the status of their application. Users will be notified by mail or SMS when their application status changes. The users, after logging in can access a portal where they can submit applications which is electronically forwarded to concerned staff. The staff is provided a portal where they can access the applications or requests submitted for them to sanction or forward to higher officials after verifying and making necessary comments on it. The users can also upload or download the necessary certificates and other documents instead of having to physically submit it. The application is not limited to any particular organization and can be modified accordingly for any new organization.

# Contents

|          |                                    |           |
|----------|------------------------------------|-----------|
| <b>1</b> | <b>INTRODUCTION</b>                | <b>1</b>  |
| 1.1      | PURPOSE . . . . .                  | 1         |
| 1.2      | SCOPE . . . . .                    | 2         |
| <b>2</b> | <b>SYSTEM ANALYSIS</b>             | <b>3</b>  |
| 2.1      | EXISTING SYSTEM . . . . .          | 3         |
| 2.2      | PROPOSED SYSTEM . . . . .          | 4         |
| 2.3      | LIFE CYCLE MODEL . . . . .         | 5         |
| 2.4      | FEASIBILITY STUDY . . . . .        | 6         |
| 2.4.1    | Technical Feasibility . . . . .    | 7         |
| 2.4.2    | Economical Feasibility . . . . .   | 7         |
| 2.4.3    | Operational Feasibility . . . . .  | 7         |
| 2.4.4    | Social feasibility . . . . .       | 7         |
| 2.5      | GANTT CHART . . . . .              | 8         |
| 2.6      | COST ESTIMATION . . . . .          | 10        |
| 2.7      | SYSTEM REQUIREMENT STUDY . . . . . | 11        |
| 2.7.1    | Hardware Requirements . . . . .    | 11        |
| 2.7.2    | Software Requirements . . . . .    | 11        |
| 2.7.3    | Safety Requirements . . . . .      | 11        |
| <b>3</b> | <b>SYSTEM DESIGN</b>               | <b>12</b> |
| 3.1      | DESIGN . . . . .                   | 12        |
| 3.2      | MODULES . . . . .                  | 13        |

|          |                                       |           |
|----------|---------------------------------------|-----------|
| 3.2.1    | Login module . . . . .                | 13        |
| 3.2.2    | Applicant module . . . . .            | 14        |
| 3.2.3    | Official module . . . . .             | 14        |
| 3.2.4    | Administrator module . . . . .        | 14        |
| 3.3      | USE CASE DIAGRAM . . . . .            | 15        |
| 3.3.1    | Purpose . . . . .                     | 15        |
| 3.3.2    | Diagram . . . . .                     | 15        |
| 3.3.3    | Description . . . . .                 | 16        |
| 3.4      | SEQUENCE DIAGRAM . . . . .            | 17        |
| 3.4.1    | Purpose . . . . .                     | 17        |
| 3.4.2    | Diagram . . . . .                     | 17        |
| 3.4.3    | Description . . . . .                 | 18        |
| 3.5      | CLASS DIAGRAM . . . . .               | 18        |
| 3.5.1    | Purpose . . . . .                     | 18        |
| 3.5.2    | Diagram . . . . .                     | 19        |
| 3.5.3    | Description . . . . .                 | 19        |
| 3.6      | ER DIAGRAM . . . . .                  | 20        |
| 3.6.1    | Purpose . . . . .                     | 20        |
| 3.6.2    | Diagram . . . . .                     | 20        |
| 3.6.3    | Description . . . . .                 | 21        |
| 3.7      | DATA FLOW DIAGRAM . . . . .           | 21        |
| 3.7.1    | Purpose . . . . .                     | 21        |
| <b>4</b> | <b>IMPLEMENTATION</b>                 | <b>29</b> |
| 4.1      | LANGUAGES AND PLATFORM USED . . . . . | 29        |
| 4.1.1    | Java . . . . .                        | 30        |
| 4.2      | SCREEN SHOTS . . . . .                | 33        |
| 4.2.1    | Login . . . . .                       | 33        |
| 4.2.2    | User Registration . . . . .           | 34        |
| 4.2.3    | Admin . . . . .                       | 35        |

|          |                               |           |
|----------|-------------------------------|-----------|
| 4.2.4    | Staff . . . . .               | 36        |
| 4.2.5    | End User . . . . .            | 38        |
| 4.2.6    | Request . . . . .             | 41        |
| <b>5</b> | <b>TESTING</b>                | <b>42</b> |
| 5.1      | CODE TESTING . . . . .        | 43        |
| 5.2      | PASSWORD TESTING . . . . .    | 43        |
| 5.3      | UNIT TESTING . . . . .        | 43        |
| 5.4      | INTEGRATION TESTING . . . . . | 44        |
| 5.5      | VALIDATION TESTING . . . . .  | 44        |
| 5.6      | SYSTEM TESTING . . . . .      | 45        |
| 5.7      | OUTPUT TESTING . . . . .      | 45        |
| 5.8      | GOAL OF TESTING . . . . .     | 46        |
| 5.9      | PASS/FAIL CRITERIA . . . . .  | 46        |
| 5.10     | PASS CRITERIA: . . . . .      | 46        |
| 5.11     | FAIL CRITERIA . . . . .       | 46        |
| 5.12     | TEST REPORTS . . . . .        | 47        |
| 5.13     | BLACK BOX TESTING . . . . .   | 48        |
| <b>6</b> | <b>FUTURE SCOPE</b>           | <b>49</b> |
| <b>7</b> | <b>CONCLUSION</b>             | <b>50</b> |
| <b>8</b> | <b>REFERENCES</b>             | <b>51</b> |
| <b>9</b> | <b>GLOSSARY</b>               | <b>52</b> |
|          | <b>INDEX</b>                  | <b>53</b> |



## List of Figures

|      |                                      |    |
|------|--------------------------------------|----|
| 2.1  | Prototype Life Cycle Model . . . . . | 6  |
| 2.2  | Gantt Chart . . . . .                | 9  |
| 3.1  | Use Case Diagram . . . . .           | 15 |
| 3.2  | Sequence Diagram . . . . .           | 17 |
| 3.3  | Class Diagram . . . . .              | 19 |
| 3.4  | ER Diagram . . . . .                 | 20 |
| 3.5  | Level 0 DFD . . . . .                | 23 |
| 3.6  | Level 1.1 DFD . . . . .              | 24 |
| 3.7  | Level 1.2 DFD . . . . .              | 24 |
| 3.8  | Level 1.3 DFD . . . . .              | 25 |
| 3.9  | Level 2.1 DFD . . . . .              | 26 |
| 3.10 | Level 2.2 DFD . . . . .              | 26 |
| 3.11 | Level 2.3 DFD . . . . .              | 27 |
| 3.12 | Level 2.4 DFD . . . . .              | 27 |
| 3.13 | Level 2.5 DFD . . . . .              | 28 |
| 4.1  | Login page . . . . .                 | 33 |
| 4.2  | Registration Page . . . . .          | 34 |
| 4.3  | Pending Request . . . . .            | 35 |
| 4.4  | Admin Hierarchy . . . . .            | 35 |
| 4.5  | Admin request . . . . .              | 36 |
| 4.6  | Staff - History . . . . .            | 36 |

|      |  |    |
|------|--|----|
| 4.7  | Staff- Request under process . . . . . | 37 |
| 4.8  | Staff - History . . . . .              | 37 |
| 4.9  | Staff- Request inbox . . . . .         | 38 |
| 4.10 | User - history . . . . .               | 38 |
| 4.11 | User - Request under process . . . . . | 39 |
| 4.12 | User -Details . . . . .                | 39 |
| 4.13 | User - send request . . . . .          | 40 |
| 4.14 | User upload page . . . . .             | 40 |
| 4.15 | Request page . . . . .                 | 41 |

## List of Tables

|     |                                     |    |
|-----|-------------------------------------|----|
| 2.1 | COCOMO Model Coefficients . . . . . | 10 |
| 5.1 | Test Report . . . . .               | 47 |
| 5.2 | Test Cases . . . . .                | 48 |

# **Chapter 1**

## **INTRODUCTION**

### **1.1 PURPOSE**

Proper document processing techniques should be carried out in any organization, to handle various requests, process them based on priority basis, produce adequate results in the form of message formats, documents etc. In any system, whenever a request for a document is encountered, control for issuing the specific documents is transferred from one official to the next higher official, until the highest official provides specific commands such as issuing the document to the required individual, rejecting the proposal etc. The current system, which focuses on manual document processing suffers from a large number of disadvantages. Since the process is manual, we have to keep separate files for each activity .Since there is a large volume of data and humans do all the handling stand alone, they are prone to errors. Moreover, a lot of time and effort is needed for proper functioning of the system especially in the case of communication between the users. Under these circumstances, a need for an automated document processing system is advised that assist in the creation of electronic documents. This process is increasingly used within certain industries to assemble legal documents, contracts and letters. In this project it is proposed to design a system which is modelled on an automated document processing such that it allows companies to minimize data entry, reduce the time spent proof-reading, and reduce the risks associated with human error.

## 1.2 SCOPE

In the proposed system all details about the system is designed into an application. It is designed as a web based application, which will work smoothly on any computer system with internet connection. For using/operating the application, there will be no need of additional official. Any person with basic computer knowledge can use this application. The applications own operational official is found to be sufficient and efficient to handle the system. Once the system is hosted on the web, the system will be automatically operated. Nowadays people are familiar with internet and e-mail, and hence, they can easily use the options from the home page of the new system. So the proposed system is operationally feasible. The document automation software is also supported by a powerful database where the documents are arranged, making updates and collaboration easy and fast.

The system proposed here is meant for use in any kind of organization and for other small scale industries. The basic steps in application processing is same everywhere, and hence the extended version of this system be used in advanced areas too. It can be used at private/public firms. It can even be used at Government offices where a large number of requests for documents are submitted by the people on a daily basis. At such places, the delay in processing increases with the number of requests, because most of the work like the transfer of the files has to be done manually by the official.

## **Chapter 2**

# **SYSTEM ANALYSIS**

### **2.1 EXISTING SYSTEM**

The process of submitting a request for say, a certain document, like the income certificate, has to go through a certain number of officials before it is finally processed. The person has to first get a written request, and then consult corresponding official for recommending or forwarding the request before it can be given to the office for further processing. During busy days, finding the official or concerned official can also turn out to be a hard task. Once it has been recommended or forwarded, the applicant can submit it to the office personnel. After submission, the applicant has to visit the office almost on a daily basis to know its status. Sometimes, it may take days because the file was not moved when it was supposed to be.

One of the main disadvantages of the existing system is the time it takes to process and physically move the request/file. If the official supposed to move the file forgets about it, count in another day for the request to be processed. The applicant (in this case) also has to move around a lot, to find the concerned official to get the request approved, which again results in a lot of wasted time and energy. Also, most of the time, the office official may not be able to predict when the requested document will be available for the applicant to collect, and hence the applicant will have to visit the office every day, till he/she gets the document(s).

## 2.2 PROPOSED SYSTEM

iREQ is a replacement for the ordinary request/ application document processing systems which depend on paper work and manual labor for processing, accepting/ rejecting each request/application. In any system, whenever a request for a document is encountered, control for issuing the specific documents is transferred from one official to the next higher official, until the highest official provides specific commands such as issuing the document to the required individual, rejecting the proposal etc. iREQ will provide an advanced request/application processing mechanism, and will make it easy to submit, process, and know the status of the request with the click of a mouse. Applicants can log in to the application with a username and password provided by the administrator after verifying the identity of the applicant. He/she will be able to access a portal to submit the applications, which will then be forwarded to the concerned officials. The applicant can also view the status of the application, like Recommendation pending or Approved. The application also provides a different login portal for the official, who, again, are provided login credentials by the administrator. The official will be able to access the applications submitted to them, and take necessary actions. Options for making comments, rejecting and forwarding (if required) the requests will be available.

The proposed system being a fully automated one, reduces the chances of errors and provides trustworthy information. In the proposed system, data lies at a one click distance whereas in the existing system shows the high speed of the proposed system. It creates a solution for the problem by enquiring what the user need and then providing the information. This creates a better feeling for the user that his/her requirement is being considered also makes the user to update and manage data easily without much training. The iREQ will save one the time typically spent entering data, tracking data, updating spreadsheets and pulling reports. This system can save a lot of time by automating the document control effort. Organizations will be able to reduce the workload of three full-time employees down to that of half of a full time employees time. Of course, each company/ workplace is different, but it is safe to say that the time saved, whatever it may be, can be better allocated to things that are more critical to the business.

## 2.3 LIFE CYCLE MODEL

The paradigm chosen for this project is prototype life cycle model. Prototype model should be used when the desired system needs to have a lot of interaction with the end users. Prototype ensures that the end users constantly works with the system and provide a feedback which is incorporated in the prototype to result in a usable system. They are excellent for designing good human computer interface systems. For the entire graphical user interfacing prototype life cycle model is used. The basic idea here is that instead of freezing the requirements before a design or coding can proceed, a through way prototype is built to understand the requirements. By using this prototype, the client can get an “actual feel” of the system, since the interactions with the prototype can enable the client to better understand the requirements of the desired system. Prototyping is an attractive idea for complicated and large systems for which there is no manual process or existing system to help determining the requirements. The prototype is usually not complete systems and many of the details are not built in the prototype. The goal is to provide a system with overall functionalities.

Users are actively involved in the development. Since in this methodology a working model of the system is provided, the users get a better understanding of the system being developed. Errors can be detected much earlier. Quicker user feedback is available leading to better solutions. Missing functionality can be identified easily. Confusing or difficult functions can be identified. Required validation, quick implementation of, incomplete, but functional, application.

As a part of applying prototyping in our project, first we generated a rough interface of our iREQ. Later on, based on the requirements alterations were made. After the customer accepts rest comes the design phase, coding, testing and then maintenance. The disadvantages of the prototype life cycle model is developing time is high and compromising in the quality and performance of the product for the sake of the customer’s satisfactions. Since the developing time is high in this type of life cycle, the developing cost is will also be high.



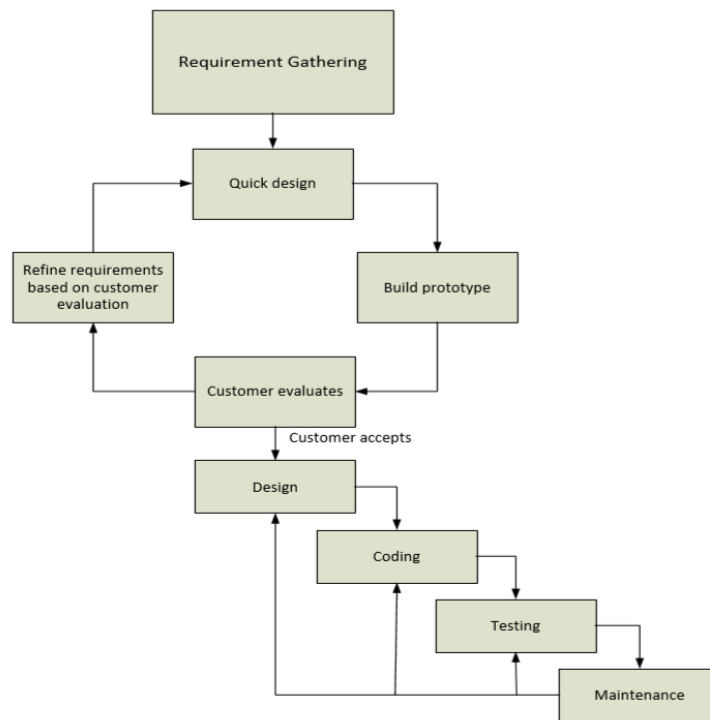


Fig. 2.1: Prototype Life Cycle Model

## 2.4 FEASIBILITY STUDY

The main objective of this study is to determine whether the proposed system is feasible or not. Mainly there are three types of feasibility study to which the proposed system is subjected as described below:

Four key considerations are involved in this feasibility.

- Economic Feasibility
- Technical Feasibility
- Operational Feasibility
- Social Feasibility

The proposed system must be evaluated from a technical viewpoint first, and if technically feasible, their impact on the organization must be assessed. If compatible, the operational system can be devised. Then those must be tested for economic feasibility.

#### **2.4.1 Technical Feasibility**

The technology required for developing the driver is identified. It has technical capability to initialize the system and perform data transfer. It also provides technical guarantee of assurance, reliability, easy access and security. Thus, since both hardware and software requirements are satisfied it is technically feasible.

#### **2.4.2 Economical Feasibility**

The system is developed at reasonable cost with the available hardware, software and manpower. So its benefits outweigh the cost. So it is economically feasible.

#### **2.4.3 Operational Feasibility**

The proposed project is beneficial because this driver software is the first of its class, so the users are encouraged to use it, and is expected to serve the user's needs on request. The user interface is designed in such a way that the users are not bound to have any doubts to use the interface.

#### **2.4.4 Social feasibility**

The proposed project will be socially feasible as the contents being shared is only inside a friend's circle. Such that it would not be used for any offensive purposes. The social feasibility determines whether the project would be accepted by the people. This assumption would in general examine the probability that the project would have to be accepted by the group of people that are directly affected by the proposed system.

## 2.5 GANTT CHART

Gantt chart is a graphical representation of allocation of resources to the activities. Here our resource is time. A Gantt chart is a type of bar chart that illustrates a project schedule. Gantt charts illustrate the start and finish dates of the terminal elements and summary elements of a project. Terminal elements and summary elements comprise the work breakdown structure of the project. Some Gantt charts also show the dependency (i.e., precedence network) relationships between activities.

Gantt charts have become a common technique for representing the phases and activities of a project work breakdown structure (WBS), so they can be understood by a wide audience. Although a Gantt chart is useful and valuable for small projects that fit on a single sheet or screen, they can become quite unwieldy for projects with more than about 30 activities. Larger Gantt charts may not be suitable for most computer displays. A related criticism is that Gantt charts communicate relatively little information per unit area of display. That is, projects are often considerably more complex than can be communicated effectively with a Gantt chart. Although project management software can show schedule dependencies as lines between activities, displaying a large number of dependencies may result in a cluttered or unreadable chart.

Because the horizontal bars of a Gantt chart have a fixed height, they can misrepresent the time-phased workload (resource requirements) of a project, which may cause confusion especially in large projects. A related criticism is that all activities of a Gantt chart show planned workload as constant. In practice, many activities (especially summary elements) have front-loaded or back-loaded work plans, so a Gantt chart with percent-complete shading actually may lead to miscommunications on the true schedule performance status.

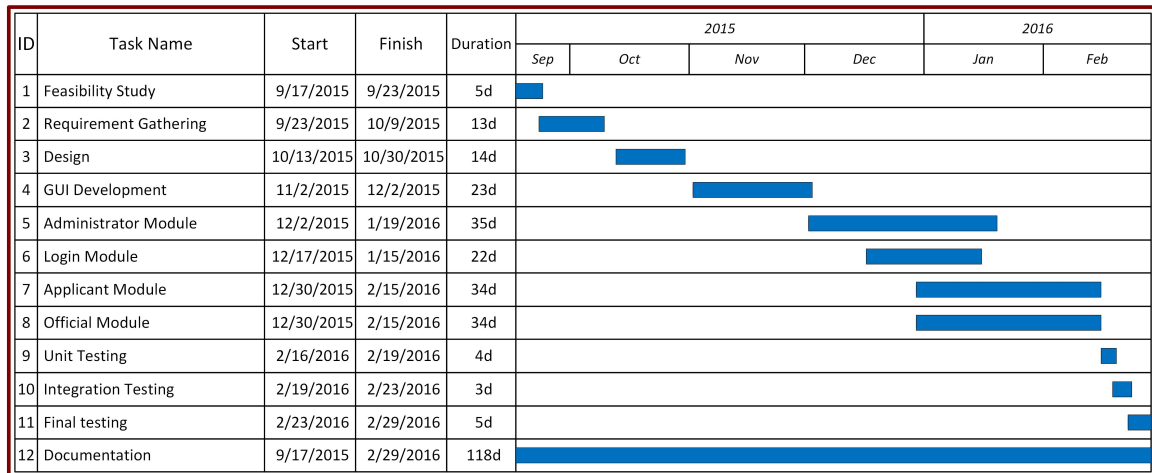


Fig. 2.2: Gantt Chart

In the Gantt Chart, as per the module name each dates are given. For the feasibility study five days are given and parallel to that the requirement analysis has been started. After this comes the design phase, during this phase it took for over fourteen days for coming up with the suitable design interface according to our project. Next as a part of the implementation stage it took twenty eight days for GUI development. Then comes the implementation of each modules according to our project. First in the Client Terminal Module it took thirty five days for its completion and for Server Management module it took twenty eight days and our last module that is the Connection Establishment module it took thirty four days for its completion. Rest comes the testing phase for double checking the performance analysis of our project by underlying them over various perspective by Unit testing, Integration testing, Final testing. At last comes the thorough documentation of each and every nook and corner of the project done, evaluated and analysed.

## 2.6 COST ESTIMATION

Basic COCOMO computes software development effort (and cost) as a function of program size. Program size is expressed in estimated thousands of lines of code (KLOC).

COCOMO applies to three classes of software projects:

- Organic projects - “small” teams with “good” experience working with “less than rigid” requirements
- Semi-detached - “medium” teams with mixed experience working with a mix of rigid and less than rigid requirements
- Embedded projects - developed within a set of “tight” constraints (hardware, software, operational ...)

The basic COCOMO equations take the form:

Effort Applied =  $a(KLOC)^b$  [ person-months ]

Development Time =  $c(EffortApplied)^d$  [months]

People required = Effort Applied / Development Time [count]

The coefficients a, b, c and d are given in the following table.

Table 2.1: COCOMO Model Coefficients

| Software project | a   | b    | c   | d    |
|------------------|-----|------|-----|------|
| Organic          | 2.4 | 1.05 | 2.5 | 0.38 |
| Semi-detached    | 3.0 | 1.12 | 2.5 | 0.35 |
| Embedded         | 3.6 | 1.20 | 2.5 | 0.32 |

### *Cost Estimation of Our Project:*

Effort for Development: 6PM

Time for development is expected to be 3months

## **2.7 SYSTEM REQUIREMENT STUDY**

### **2.7.1 Hardware Requirements**

The recommended minimum system should have 2 GHz processor, 1 GB RAM, and disk space as required.

### **2.7.2 Software Requirements**

- Tools : NetBeans IDE 8.0.2
- Front End : Java
- Back End : MySQL
- Database : MySQL
- Platform : Windows, Linux
- Web browser : Internet Explorer 6.0 or above

### **2.7.3 Safety Requirements**

Complete access to the system is allowed only to the administrator of this system. Clients can do nothing other than using the applications provided.

## **Chapter 3**

# **SYSTEM DESIGN**

### **3.1 DESIGN**

Design of the system includes mainly two steps:

- System design
- Detailed design.

In System design a structural framework for the entire system is created. It is done in such a way that related part come under particular groups. Thus after the system design, a network of different groups is obtained. It is the high-level strategy for solving the problem and building a solution. It includes the decision about the organization of the system into subsystems, the allocation of subsystems to hardware and software components, and major conceptual and policy decisions that form the framework for the detailed design.

In detailed design, each group is studied in detail and the internal operations are decided. Based on this, the data structures and the programming language to be used are decided. Apart from detailed design, the system design can be grouped into physical design and structural design. The physical design maps out the details of the physical system and plans the system implementation and specifies the hardware and software requirements.

Structured design is an attempt to minimize the complexity and make a problem manageable by subdividing into smaller segments, which is called modularization or decomposition. In this way structuring minimizes intuitive reasoning and promotes maintainable provable of

systems. The structured design partitions a program into small, independent modules. They are arranged in a hierarchy that approximates a model of the business are and is organized in a top-down manner.

Logical design proceeds in a top-down manner. General features, such as reports and inputs are identified first. Then each is studied individually and in more detail. Hence the structured design is an attempt to minimize the complexity and make a problem.

## **3.2 MODULES**

iREQ can be divided into the following modules:

- Login module
- Applicant module
- Official module
- Administrator module

### **3.2.1 Login module**

This is the first module of the iREQ. Anyone who wishes to use the application needs to pass through this module. In other words, it can be said that the login module is the authentication module. The module provides separate login screens for applicants and official users. The module also provides the option for new user registrations. On entering the correct username and password, a verification code will be sent to the phone number of the user, which the user has to enter for accessing the application. The two step verification has been provided to protect sensitive user data from unauthorized access. Digital signature will further improve the security of the system.



### **3.2.2 Applicant module**

The applicants, after successful login can access the applicant module. This module provides options for submitting a request, changing user details, viewing status of submitted requests, downloading scanned copies of official documents, and viewing the history. On submission of the request, a confirmation will be send to the mobile device, informing the user that a request has been submitted. The user should use this confirmation code to pursue the process. The requests submitted by the user are saved into iREQs secure database. Further transfer of the application/documents take place according to the official hierarchy set by the applicant. The hierarchy can be customized according to the user needs. The applicant also gets notified by SMS or Email when the status of the submitted application.

### **3.2.3 Official module**

Similar to applicant module, the official can access the module only after successful login. The module provides options different from applicant module, such as options for viewing pending applications/requests, handling the requests, making comments, and changing the status of the requests. The official will be notified whenever a new application/request is submitted to them via SMS to the mobile number provided at the time of registration. This will help in quicker processing of the requests. If a request is not handled within a specified time, the request will be forwarded to a different official, who is equally qualified to process it.

### **3.2.4 Administrator module**

The administrator module will mainly contain options for adding a new user and deleting an existing user. The administrator can edit and view the user details. The administrator can also view the list of users who accessed the database. The administrator can customize the interface of the application according to the needs of the organization.

## 3.3 USE CASE DIAGRAM

### 3.3.1 Purpose

Use case diagram in the Unified Modelling language (UML) is a type of behavioural diagram. Its purpose is to represent the graphical overview of the functionality provided by a system in terms of actors, their goals (represented use cases), and any dependencies between those use cases. The main purpose of a use case diagram is to show what system functions are performed for which actor.

### 3.3.2 Diagram

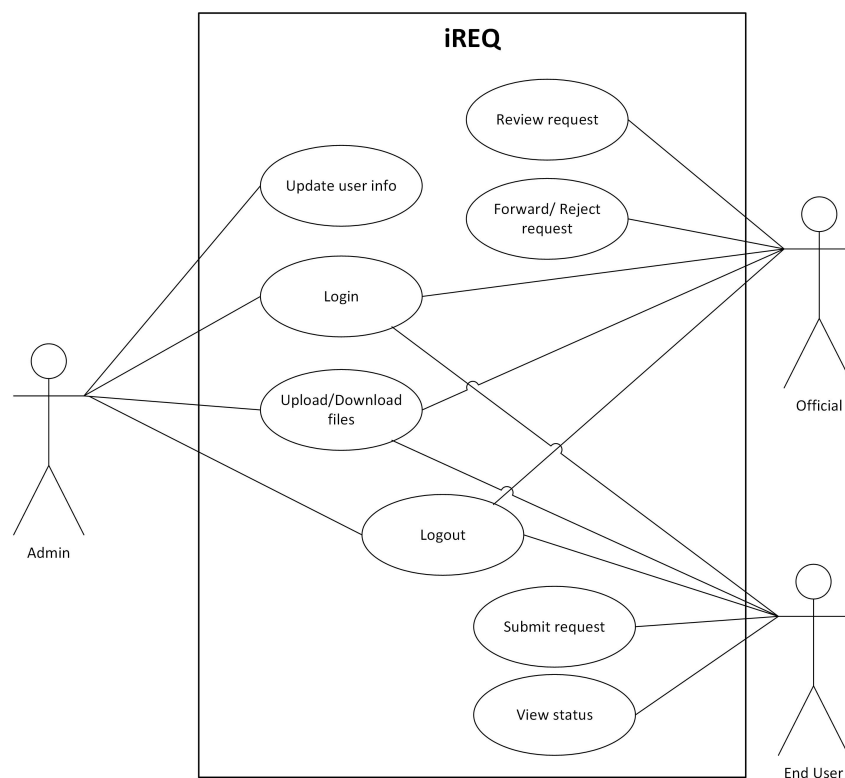


Fig. 3.1: Use Case Diagram

### 3.3.3 Description

**Admin should be able to:**

- Login to the system through the first page of the application with his/ her unique user-name and password.
- Update user information
- Upload or download the necessary files
- Logout when the transaction is completed

**Student should be able to:**

- Login to the system through the first page of the application with his/ her unique user-name and password.
- Submit a request after successful login, and can upload additional files if required
- View the status of the request/ application.
- Download the necessary files.
- Logout when the transaction is completed

**Staff should be able to:**

- Login to the system through the first page of the application with his/ her unique user-name and password.
- Review/forward/reject request after successful login, and can upload additional files if required
- Upload or Download the files.
- Logout when the transaction is completed

## 3.4 SEQUENCE DIAGRAM

### 3.4.1 Purpose

A Sequence diagram depicts the sequence of actions that occur in a system. It portrays the different perspectives of behaviour of the system and different types of inferences can be drawn from them. The invocation of methods in each object, and the order in which the invocation occurs is captured in a Sequence diagram. This makes the Sequence diagram a very useful tool to easily represent the dynamic behaviour of a system.

### 3.4.2 Diagram

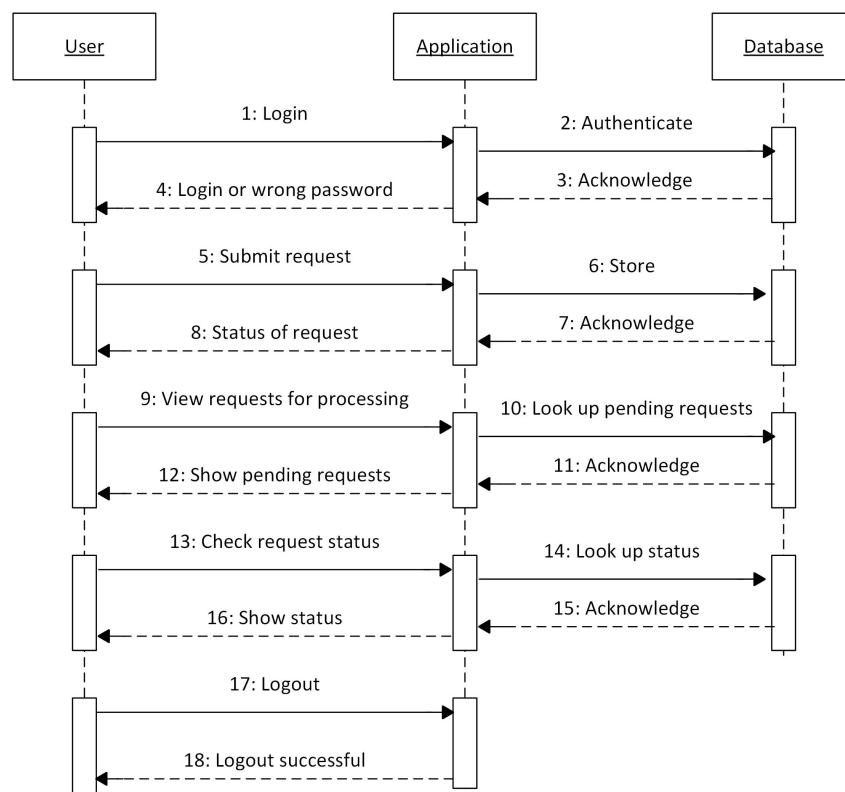


Fig. 3.2: Sequence Diagram

### 3.4.3 Description

After successful authentication with his/ her unique username and password, the user will be able to Submit/review a request, depending on their account type. After submitting a request through the iREQ web portal, User will be able to enquire the status of the pending request. While in the background, each of the requests are stored separately in the database and the processing document statuses are made available to the user. Transaction is completed when user logout from the system.

## 3.5 CLASS DIAGRAM

### 3.5.1 Purpose

Class diagram in the Unified Modelling Language (UML) is a type of static structure diagram that describes the structure of a system by showing the system's classes, their attributes, operations (or methods), and the relationships among objects. The classes in a class diagram represent both the main objects, interactions in the application and the classes to be programmed. A class with three sections. In the diagram, classes are represented with boxes which contain three parts: The top part contains the name of the class. The middle part contains the attributes of the class. The bottom part gives the methods or operations the class can take or undertake.

To specify the visibility of a class member (i.e., any attribute or method), some notations must be placed before the member's name.

- + Public
- - Private
- / Derived (can be combined with one of the others)
- Package

### 3.5.2 Diagram

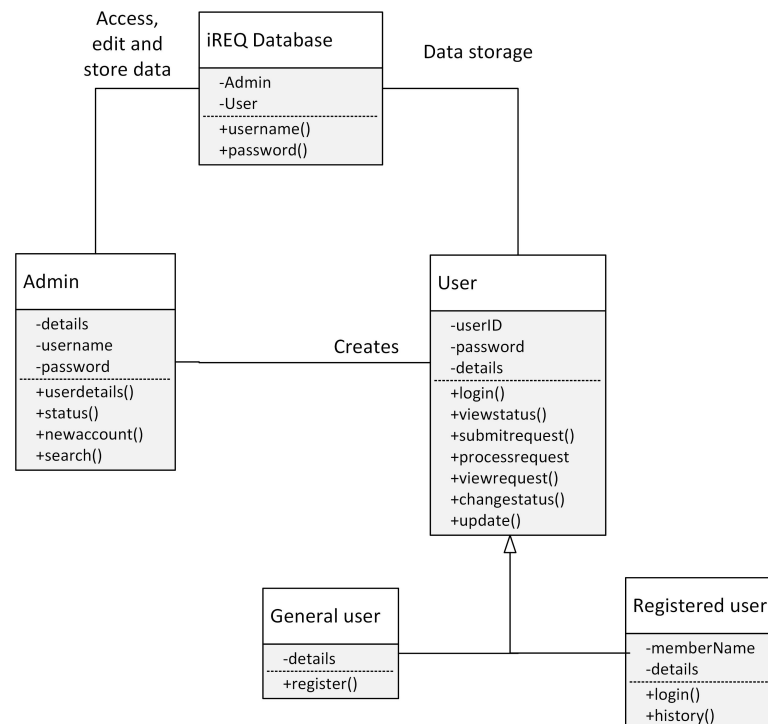


Fig. 3.3: Class Diagram

### 3.5.3 Description

The iREQ system has three classes. The first one is the iREQ Database. Admin and user are the attributes for this class. username() and password() are the methods provided by the class. Admin and user are the next classes, userid and password are their common attributes. but their functions are different.

A user can be general user or a registered user. This is represented by concept of Generalization. The Generalization relationship indicates that one of the two related classes is considered to be a specialized form of the other and the superclass is considered a Generalization of the subclass. The UML graphical representation of a Generalization is a hollow triangle shape on the superclass end of the line (or tree of lines) that connects it to one or more subtypes. The generalization relationship is also known as the inheritance or is a relationship.

## 3.6 ER DIAGRAM

### 3.6.1 Purpose

Database is recognized as a standard and is available virtually for every computer system. The general theme behind every database is to integrate all the information. The database is an integrated collection of data and provides centralized access to the data. The user authentication in the project and share details is implemented using the database.

### 3.6.2 Diagram

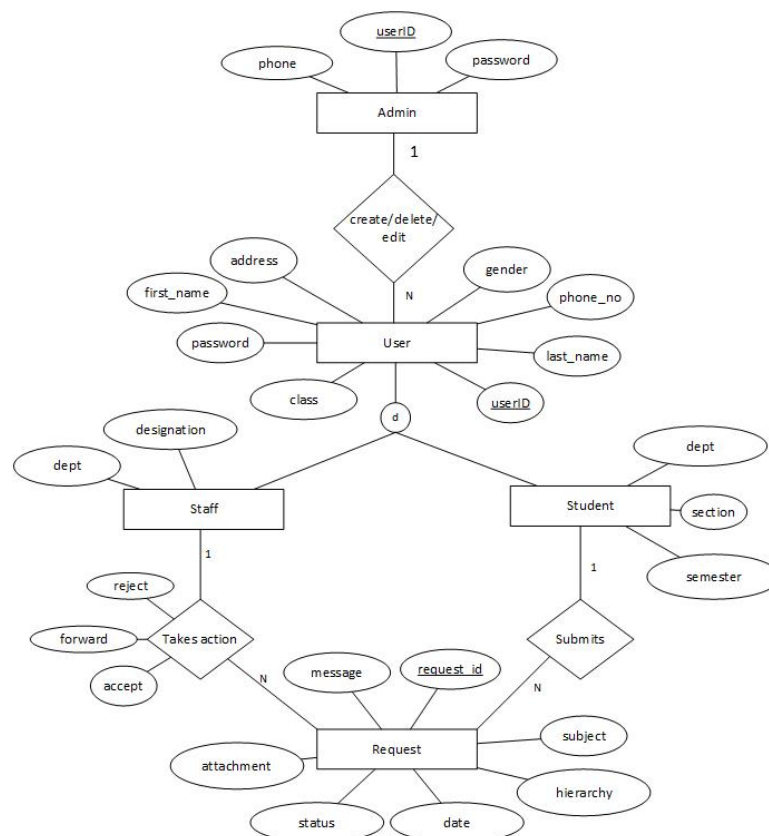


Fig. 3.4: ER Diagram

### 3.6.3 Description

The entity admin has attributes name, admin id and password where admin id is given as primary key. Student and staff are two entities where the admin can add or delete. Student has the attribute such as date of birth, branch, name, phone number, student id, gender, mail id and age where mailed is a multivalued attribute since it can take multiple values. Age is a derived attribute since its value can be derived from date of birth. Staff has attributes such as username, password, staff id and designation where staff id is a primary key.

## 3.7 DATA FLOW DIAGRAM

### 3.7.1 Purpose

A data flow diagram is a simple graphical formalism that represents a system in terms of the input data to the system, various processing carried on these data, and the output data generated by the system. A DFD model represents the data flow aspects and does not show the sequence of execution of the different functions and conditions based on which a function may or may not be executed. In fact it completely ignores aspects such as control flow, the specific algorithms used by the functions etc.

A DFD model uses a very limited number of primitive symbols to represent the functions performed by the system and data flow among these functions. This network is constructed by using a set of symbols that do not imply a physical implementation. It is a graphical tool for structured analysis of the system requirements. DFD models a system by using external entities, from which data flows to a process, which transforms the data and creates, output-data-flows which go to other processor or external entities or files. Data in files may also flow to processes as inputs.

### Notations used in dataflow

- **Process**

Process shows the work of the system. Each process has one or more data inputs and



produce one or more data outputs. Processes are represented by rounded rectangles in Data Flow Diagram. Each process has a unique name and number. This name and number appears inside the rectangle that represents the process in a Data Flow Diagram.



- **Data Stores**

A data store is a repository of data. Processes can enter data, into a store or retrieve the data from the data store. Each data has a unique name.



- **Data Flows**

Data flows show the passage of data in the system and are represented by lines joining system components. An arrow indicates the direction of flow and the line is labelled by name of the data flow.



- **External Entity**

External entities are outside the system but they either supply input data into the system or use other systems output. They are entities on which the designer has control. They may be an organizations customer or other bodies with which the system interacts. External entities that supply data into the system are sometimes called source. External entities

that use the system data are sometimes called sinks. These are represented by rectangles in the Data flow Diagram.



## LEVEL 0

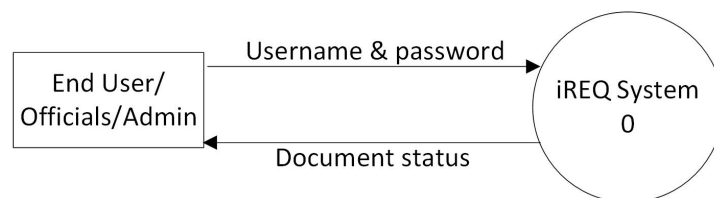


Fig. 3.5: Level 0 DFD

### Description

Level 0 represents the basic level/structure of the iREQ system. It is characterized by a set of user classes involving user, staff and administrator. Any user can access the iREQ system, after completing the secure authentication method, by providing the correct username and password.

## LEVEL 1

### LEVEL 1.1 (USER)

### Description

It represents the decomposed structure of the level 0, which corresponds to the User activities. After successful authentication, user can upload /submit a request, change the password, download a processed document, refer FAQ for answers etc.

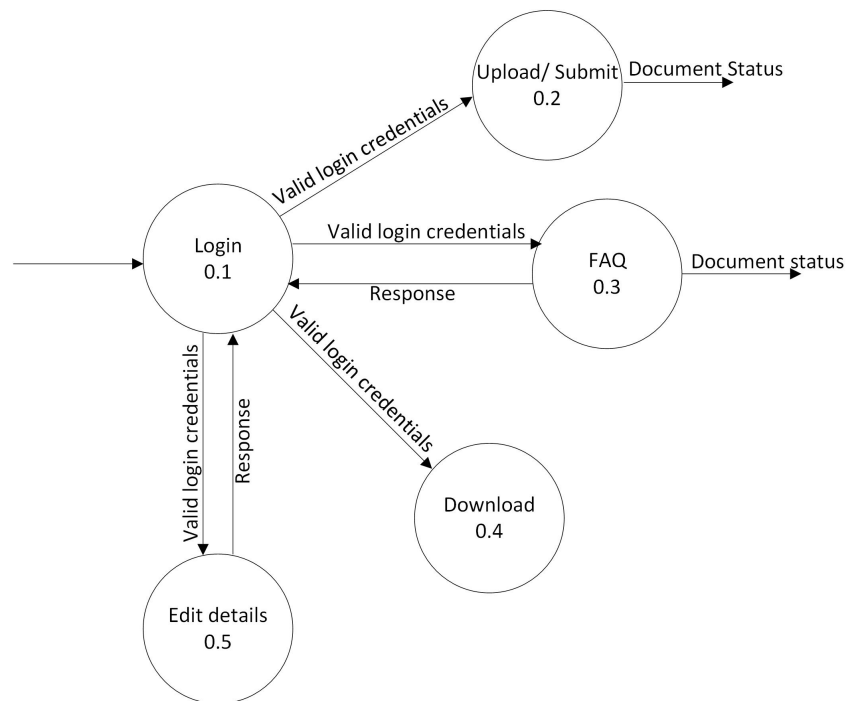


Fig. 3.6: Level 1.1 DFD

**LEVEL 1.2 (Official)****Description**

It represents the decomposed form of the level 0, involving the Staff activities. After successful login, Staff can view notifications relating to the newly submitted or pending user requests. Selected notifications are considered for document review. Staff also has an option to change his/her password.

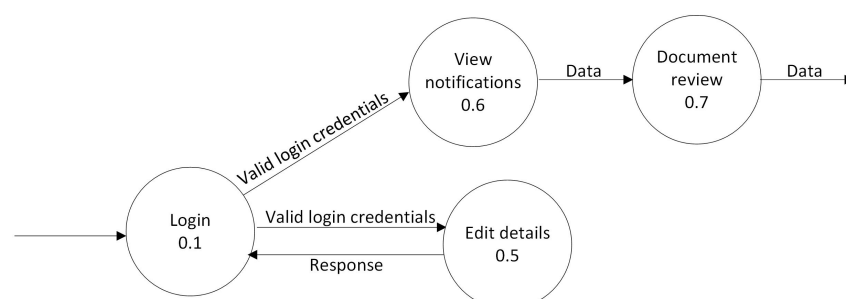


Fig. 3.7: Level 1.2 DFD

### LEVEL 1.3 (Admin)

#### Description

It represents decomposed form of the level 0, involving the admin activities. After successful login, administrator can change the password, modify links in the web-pages, update individual user information etc.

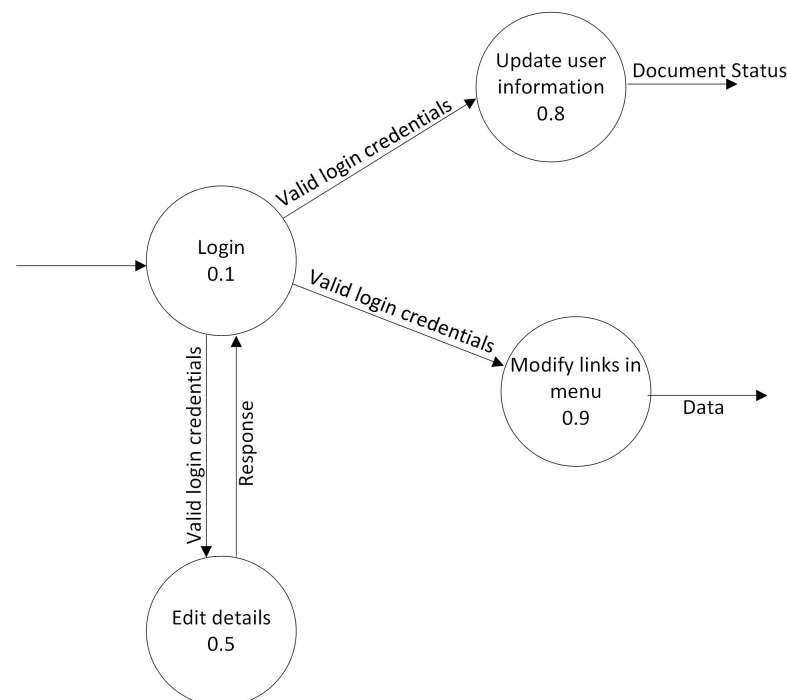


Fig. 3.8: Level 1.3 DFD

### LEVEL 2

#### LEVEL 2.1 (Login)

#### Description

The Login activity is supported by a powerful 2-level authentication mechanism. During each login, the username and password is taken as input and information is validated by referring the database. After successful validation, a confirmation code is sent to the mobile of the

user. User have to use this code to complete the login process. It is common to all user classes.

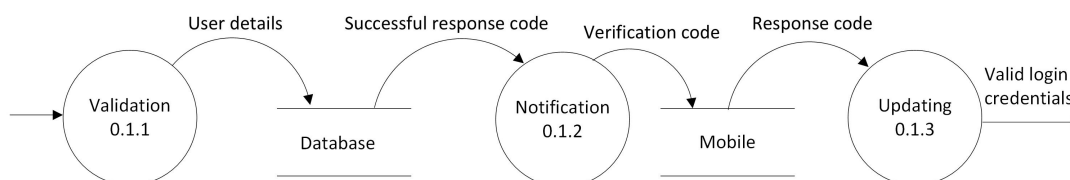


Fig. 3.9: Level 2.1 DFD

## LEVEL 2.2(Download)

### Description

Download activity is concerned with the user. User is provided with a provision for downloading the requested document. During each retrieval, corresponding document id is used for validation from the database and retrieval of the document.

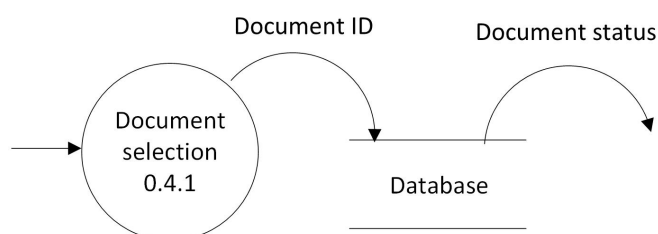


Fig. 3.10: Level 2.2 DFD

## LEVEL 2.3 (Update User Info)

### Description

This activity is concerned with the Administrator. Admin can modify/update individual user information, by retrieving existing user information from the database, making changes to the data and updating the database.

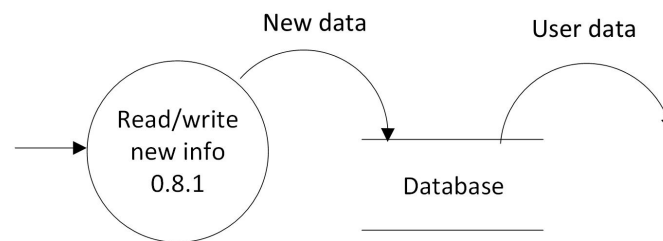


Fig. 3.11: Level 2.3 DFD

## LEVEL 2.4 (Document Review)

### Description

Document review is one of the prime activities in the system, which is concerned with the staff. During document review, Staff can either forward the request to another staff of higher privilege, or issue the document to the user. Staff also have an option for uploading document formats to the user.

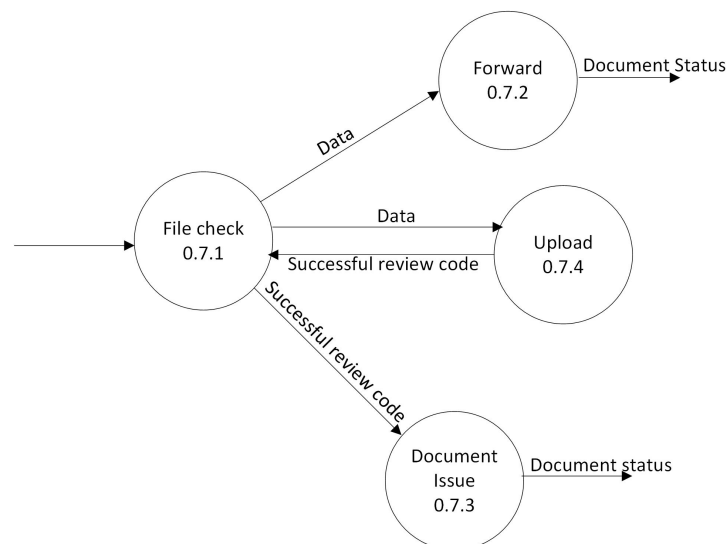


Fig. 3.12: Level 2.4 DFD

**LEVEL 2.5 (Password Change)****Description**

Password change process is common to all the user classes. During each activity, after the successful validation of existing password from the database, new password is read and updated into the database.

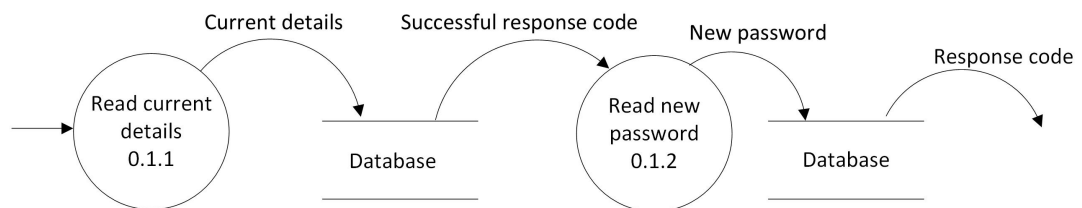


Fig. 3.13: Level 2.5 DFD

## **Chapter 4**

# **IMPLEMENTATION**

Implementation is the stage in the project where the theoretical design is turned into a working system and is giving confidence on the new system for users that it will work effectively and effectively. It involves careful planning, investigation of the current system and its constraints on implementation, design of achieve the changeover, an evolution of change over method. A part from planning major task of preparing the implementation are education and training of users .The more complex system being implemented, the more involved will be the system analysis and the design effort required just for implementation.

The product is developed in java environment in windows platform. This chapter may explain our implementation details.

### **4.1 LANGUAGES AND PLATFORM USED**

The product is developing in Java environment and we are using MySQL for database. Microsoft SQL Server automatically tunes many of the server configuration options, therefore requiring little, if any, tuning by a system administrator. Although these configuration options can be modified by the system administrator, it is generally recommended that these options be left at their default values, allowing SQL Server to automatically tune itself based on run-time conditions.



### 4.1.1 Java

Java is a general purpose, concurrent, class-based, object-oriented computer programming language that is specifically designed to have as few implementation dependencies as possible. It is intended to let application developers write once, run anywhere (WORA), meaning that code that runs on one platform does not need to be recompiled to run on another. Java applications are typically compiled to bytecode (class file) that can run on any Java virtual machine (JVM) regardless of computer architecture. Java is, as of 2012, one of the most popular programming languages in use, particularly for client-server web applications, with a reported 10 million users. Java was originally developed by James Gosling at Sun Microsystems (which has since merged into Oracle Corporation) and released in 1995 as a core component of Sun Microsystems Java platform. The language derives much of its syntax from C and C++, but it has fewer low-level facilities than either of them.

JAVA offers a number of advantages to developers:

- **Simple:** Java was designed to be easy to use and is therefore easy to write, compile, debug, and learn than other programming languages. The reason why Java is much simpler than C++ is because Java uses automatic memory allocation and garbage collection where else C++ requires the programmer to allocate memory and to collect garbage. Because Java is simple, it is easy to read and write. By keeping the language simple Java help programmers write bug-free code. Despite its simplicity Java has considerably more functionality than C, primarily because of the large class library. About half of the bugs in programs are related to memory allocation and reallocation. Therefore the second important addition Java makes to providing bug-free code is automatic memory allocation and reallocation by making use of garbage collection.
- **Platform-independent:** One of the most significant advantages of Java is its ability to move easily from one computer system to another. Java is compiled to an intermediate form called byte-code. A Java program never really executes natively on the host machine. Rather a special native program called the Java interpreter reads the byte code and executes the corresponding native machine instructions. Thus to port Java programs to

a new platform all that is needed is to port the interpreter and some of the library routines. Even the compiler is written in Java. The byte codes are precisely defined, and remain the same on all platforms. The ability to run the same program on many different systems is crucial to World Wide Web software, and Java succeeds at this by being platform-independent at both the source and binary levels.

- **Secure:** Java is one of the first programming languages to consider security as part of its design. The Java language, compiler, interpreter, and runtime environment were each developed with security in mind. Java was designed from the ground up to allow for secure execution of code across a network, even when the source of that code was entrusted and possibly malicious. This required the elimination of many features of C and C++. Most notably there are no pointers in Java. Java programs cannot access arbitrary addresses in memory. All memory access is handled behind the scenes by the (presumably) trusted runtime environment. Furthermore Java has strong typing. Variables must be declared, and variables do not change types when you aren't looking. Casts are strictly limited to casts between types that make sense. Java implements a robust exception handling mechanism to deal with both expected and unexpected errors. Most importantly Java applets can be executed in an environment that prohibits them from introducing viruses, deleting or modifying files, or otherwise destroying data and crashing the host computer.
- **Robust:** Robust means reliable and no programming language can really assure reliability. Java puts a lot of emphasis on early checking for possible errors, as Java compilers are able to detect many problems that would first show up during execution time in other languages.
- **Dynamic:** The compiler searches the current directory and directories specified in the CLASSPATH environment variable to find other classes explicitly referenced by name in each source code file. If the file you're compiling depends on other, non-compiled files the compiler will try to find them and compile them as well. The compiler is quite smart, and can handle circular dependencies as well as methods that are used before they're declared. It also can determine whether a source code file has changed since the last time

it was compiled. More importantly, classes that were unknown to a program when it was compiled can still be loaded into it at runtime.

- **Multithreaded:** Java is inherently multi-threaded. A single Java program can have many different threads executing independently and continuously. Three Java applets on the same page can run together with each getting equal time from the CPU with very little extra effort on the part of the programmer. This makes Java very responsive to user input. It also helps to contribute to Javas robustness and provides a mechanism whereby the Java environment can ensure that a malicious applet doesnt steal all of the hosts CPU cycles.
- **Object oriented:** In object-oriented programs data is represented by objects. Objects have two sections, fields (instance variables) and methods. Fields tell you what an object is. Methods tell you what an object does. These fields and methods are closely tied to the objects real world characteristics and behavior. When a program is run messages are passed back and forth between objects. When an object receives a message it responds accordingly as defined by its methods.
  - Object oriented programming is alleged to have a number of advantages including:
    - \* Simpler, easier to read programs.
    - \* More efficient reuse of code.
    - \* Faster time to market.
    - \* More robust, error-free code.

## 4.2 SCREEN SHOTS

### 4.2.1 Login

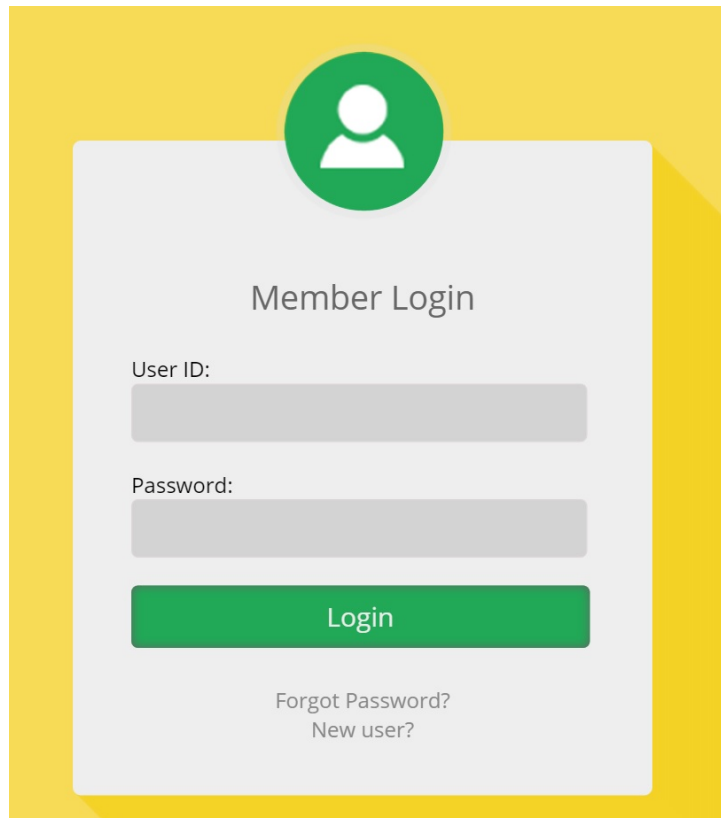


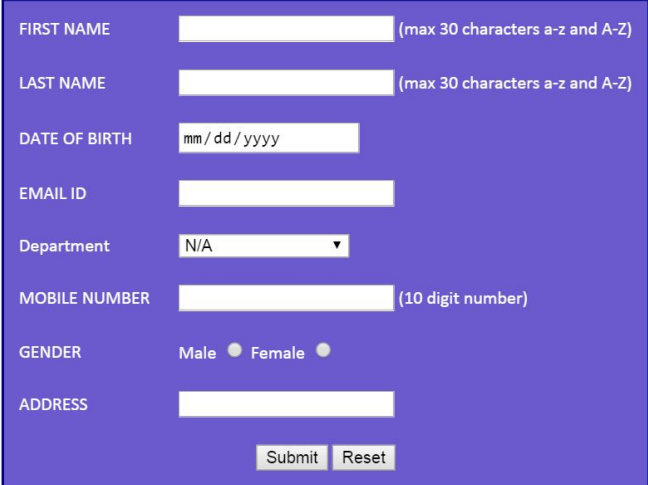
Fig. 4.1: Login page

#### Description

The login page consists of two text fields where the users need to enter their login credentials to login to the iREQ system. Two links for new user registration and retrieving forgotten passwords are also provided in the page. On logging in with correct credentials, the user has to go through two step verification to proceed.

### 4.2.2 User Registration

#### USER REGISTRATION FORM



A user registration form with a light blue background. It contains the following fields and controls:

- FIRST NAME**: Text input field with a placeholder "(max 30 characters a-z and A-Z)".
- LAST NAME**: Text input field with a placeholder "(max 30 characters a-z and A-Z)".
- DATE OF BIRTH**: Text input field with a placeholder "mm/dd/yyyy".
- EMAIL ID**: Text input field.
- Department**: Dropdown menu with "N/A" selected.
- MOBILE NUMBER**: Text input field with a placeholder "(10 digit number)".
- GENDER**: Radio buttons for "Male" and "Female".
- ADDRESS**: Text input field.
- Submit** and **Reset** buttons at the bottom right.

Fig. 4.2: Registration Page

#### Description

The user enter their details.

### 4.2.3 Admin

iREQ | Admin

Welcome Admin [Settings](#) [Logout](#)

Main page Request details Hierarchy

Wed Mar 09 2016 07:36:57 GMT+0530 (India Standard Time)

Today!

Today is  
Wednesday  
9 Mar 2016  
Saint Joseph's Day is on  
19th March 2016

Pending user registrations

No new entries!

Create selected Delete selected

User details New users

Enter details to see user information

User ID:

OR

First name:

Last name:

Fig. 4.3: Pending Request

### Description

Pending request are shown.

iREQ | Admin

Welcome Admin [Settings](#) [Logout](#)

Main page Request details Hierarchy

Wed Mar 09 2016 07:40:21 GMT+0530 (India Standard Time)

Today!

Today is  
Wednesday  
9 Mar 2016  
Saint Joseph's Day is on  
19th March 2016

Set or view hierarchy

Department:

Designation:

Fig. 4.4: Admin Hierarchy

### Description

This page show the admin hierarchy page. The admin has the power to set the hierarchy.

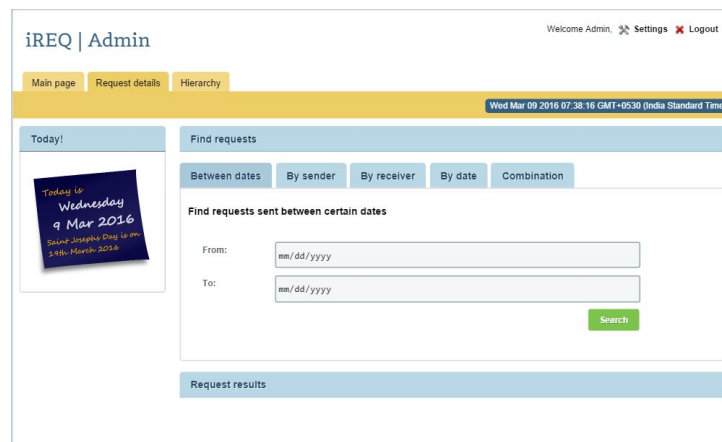


Fig. 4.5: Admin request

## Description

The page shows the staff request details like the send date receive date etc.

### 4.2.4 Staff

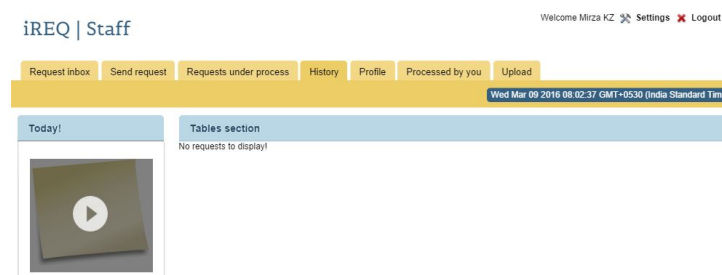


Fig. 4.6: Staff - History

## Description

This page show the staff history page .Request details will be shown here.

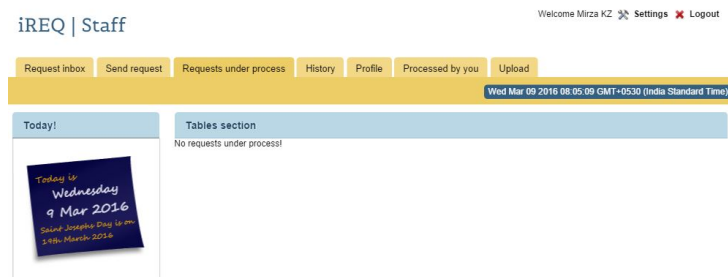


Fig. 4.7: Staff- Request under process

## Description

This page indicates the staff history page, when new request are passed history show the necessary details.

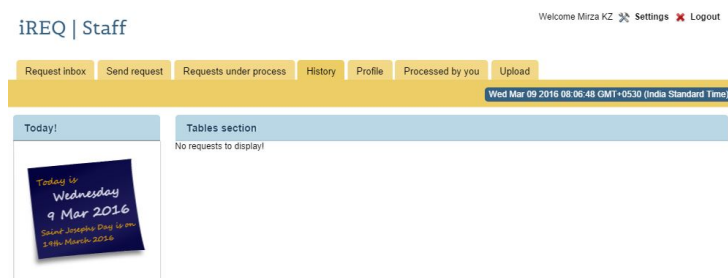


Fig. 4.8: Staff - History

## Description

This page show the user request inbox page. The request accepted or passed till now are shown.



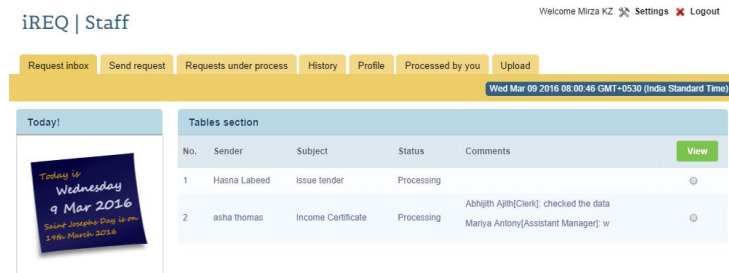


Fig. 4.9: Staff- Request inbox

## Description

This indicate the user history page which indicates the request under process.

### 4.2.5 End User

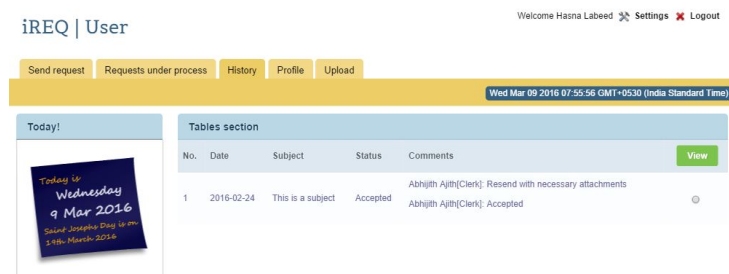


Fig. 4.10: User - history

## Description

This is a user request page which indicates the current request under process. The request can also be viewed by the officer.

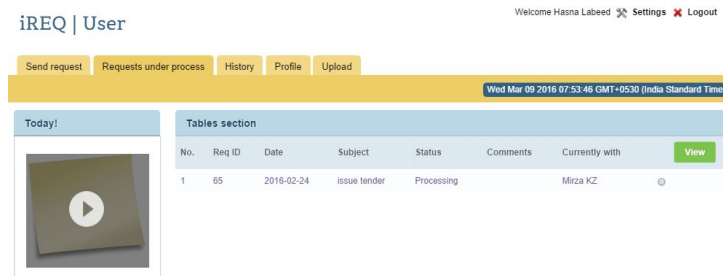


Fig. 4.11: User - Request under process

## Description

This page shows the requests sent by the user which is under process. The page also shows the status of the request, the name of the staff who currently has the request and an option to view the request, with the help of a radio button.

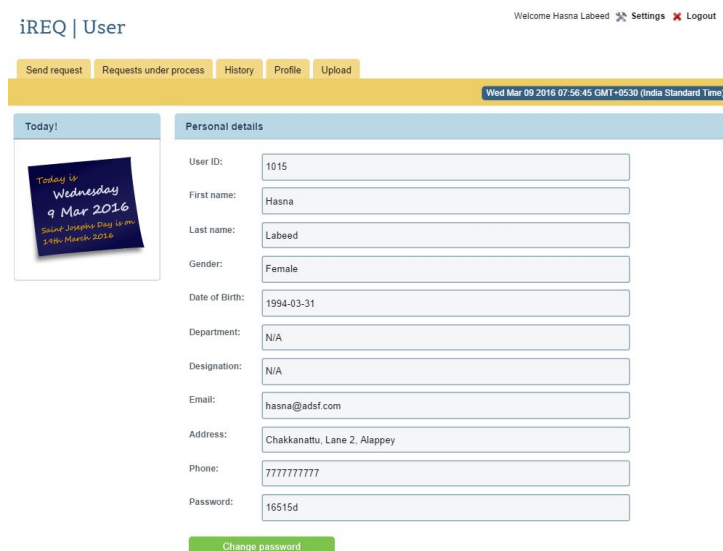


Fig. 4.12: User -Details

## Description

This is normal user history page which show the entered user details in the registration form ie Their first name, second name id, address etc

The screenshot shows the 'iREQ | User' interface. At the top, there's a navigation bar with 'Send request', 'Requests under process', 'History', 'Profile', and 'Upload'. A welcome message 'Welcome Hasna Labeed' and links for 'Settings' and 'Logout' are on the right. A timestamp 'Wed Mar 09 2016 07:52:34 GMT+0530 (India Standard Time)' is displayed. The main content area is divided into two sections. The left section, titled 'Today!', features a calendar snippet for Wednesday, 9 Mar 2016, noting 'Saint Joseph's Day is on 14th March 2016'. The right section, titled 'Request currently being processed', contains a form with fields for 'Subject:', 'To:' (a dropdown menu), and 'Message:'. A green 'Submit' button is located to the right of the message field.

Fig. 4.13: User - send request

## Description

This is a user request page where user can submit their request to corresponding officers.

The screenshot shows the 'iREQ | User' interface for the upload page. The navigation bar and header are identical to the previous figure. The main content area has a 'Personal details' section. On the left, there's a 'Today!' section with the same calendar snippet. The 'Personal details' section on the right contains a text input field with the placeholder 'Enter the associated request ID:' and a green 'Go' button below it.

Fig. 4.14: User upload page

## Description

This is a user upload page where the user can upload any request after entering their user id.

### 4.2.6 Request

Hasna Labeed  
Chakkannattu  
Lane 2  
Alappey  
2016-02-24

The Manager  
Computer Science  
Office

**Subject:** issue tender [Attachment included]

Sir/ Madam,

please issue the notice for the invitation of tenders to the maintenance of our office buildings immediately

Yours faithfully,  
Hasna Labeed

---

**Comments so far:**

---

View attachments:  
Attachment 1 ▾

---

**Action**

**Comments**

Fig. 4.15: Request page

### Description

This is a normal request page where the corresponding officer can take any action. The officer can accept or reject the request. The status will be notified to the users .

## **Chapter 5**

# **TESTING**

When a system is developed, it is hoped that it performs properly. In practice, however some errors always occur. The main purpose of testing an information system is to find the error and correct them. A successful test is one that finds an error. System testing is a critical aspect of Software Quality Assurance and represents the ultimate review of specification, design and coding. Testing is the process of executing a program with the intent of finding an as yet undiscovered error. Nothing is complete without testing. Testing is vital to the success of the system.

The main objectives of the system testing are:

- To ensure during operation the system will perform as per specification.
- To make sure that the system meets user requirements during operation.
- To verify that the controls incorporated in the system function as intended.

If the testing conducted successfully, it will uncover errors in the software. As a secondary benefit, testing demonstrates that the software functions appear to be working according to specification and that performance requirements appear to have been satisfied.

The system “Berry Terminal” is tested in such a way that almost all errors that may occur are found and corrected. The test process carried out in this system includes the following:

## **5.1 CODE TESTING**

In code testing the logic of the developed system is tested. For this every module of the program is executed to find any error. To perform specification test, the examination of the specifications stating what the program should do and how it should perform under various conditions. This testing was done side by side with coding. This examines the logic of the program. In java special test cases are used for testing the code. Every part of the program was tested in this phase.

## **5.2 PASSWORD TESTING**

This is done for user authentication. This process ensures security of the system thus avoiding unauthorized access to the system. When the user enters the username and password, checking it with the already registered usernames and passwords will validate him. If it matches then only the user is allowed to enter into the system. Otherwise he is denied access and there by strong security is provided. The password checking was done for the system. Only registered users were able to login.

## **5.3 UNIT TESTING**

Unit testing is undertaken after a module has been coded and reviewed. Before carrying this testing, the unit test cases have to be designed and the test environment for the unit under test has to be developed. The various test cases are driver and stub modules. The main objective is to determine the correct working of the individual modules. During the testing each module is isolated from other modules and individually unit tested. It involves a precise definition of the test cases, testing criteria, and management of test cases. The modules that are tested include Client Terminal, Network Management and Server Management module.

## 5.4 INTEGRATION TESTING

System testing does not test the software as a whole, but rather than integration of each module in the system. The primary concern is the compatibility of individual modules. One has to find areas where modules have been designed with different specifications of data lengths, type and data element name. Testing and validation are the most important steps after the implementation of the developed system. The system testing is performed to ensure that there are no errors in the implemented system. The software must be executed several times in order to find out the errors in the different modules of the system. Each of the modules were integrated together and subjected to testing.

## 5.5 VALIDATION TESTING

Validation refers to the process of using the new software for the developed system in a live environment i.e., new software inside the organization, in order to find out the errors. The validation phase reveals the failures and the bugs in the developed system. It will come to know of the practical difficulties the system faces when operated in the true environment. Validation test was performed in the Login section. By testing the code of the implemented software, the logic of the program can be examined. A specification test is performed to check whether the specifications stating the program are performing under various conditions. Apart from these tests, there are some special tests conducted which are given below:

- **Peak load test** This determines whether the new system will handle the volume of activities when the system is at the peak of its processing demand. The test has revealed that the new system is capable of handling the demands at the peak time.
- **Storage testing** This determines the capacity of the new system to store transaction data on a disk or on other files. The proposed software has the required storage space available.
- **Performance time testing** This test determines the length of the time used by the system to process transaction data.

## 5.6 SYSTEM TESTING

After all units of a program have been integrated together and tested system testing is taken up. It is same for both procedural and object oriented programming. System tests are designed to validate a fully developed system to assure that it meets its requirements. The system test cases can be classified into performance and functionality test cases. The functionality test cases are designed to check whether the software satisfies the functional requirements as documented in the SRS document. The performance tests on the other hand test the conformance of the system with non-functional requirements of the system.

## 5.7 OUTPUT TESTING

After the performance of unit testing, the next step is output testing. No system would be useful if it does not produce the required output in the specific format, thus output format on the screen is found to be correct when the format was designed in the system phase according to the user need.

The maintenance of software is the time period in which software product performs useful works. Maintenance activities involve making enhancement to software product, adapting product to new environment and correcting problems. It includes both the improvement of the system function.

It may involve the continuing involvement of a large proportion of computer department resources. The main task may be to adapt existing system in a changing environment. System should not be changed casually following informal requests. To avoid unauthorized amendments, all requests for change should be channelled to a person nominated by management. The nominated person has sufficient knowledge of the organization's computer based systems to be able to judge the relevance of each proposed change.

No annual costs for support or maintenance are required. Of course, the individual system components come with limited warranty from the manufacturers, i.e., the PC, camera, etc. There is no obligation to purchase or pay for any extended maintenance or support.



## **5.8 GOAL OF TESTING**

Many users may use our project. So the project designer must test all the modules of the project. The main goal of our project is, whenever user uses our project, it should run without any error.

## **5.9 PASS/FAIL CRITERIA**

The pass/fail criteria specifies a set of constraints whose satisfaction leads to approval or disapproval of the proper functioning of the system .

### **5.10 PASS CRITERIA:**

The system must meet all the functional and non-functional requirements. Pass all the test cases, get the expected response, and get acceptable performance to be tested pass.

- User can login and submit request.
- Officials can verify and process the request
- Periodic update of database

### **5.11 FAIL CRITERIA**

If one of the following situations happens, the system is considered to fail:

- User cannot login
- Database updation failure
- OTP sending failed
- Human error

## 5.12 TEST REPORTS

The detailed test reports prepared for each function. A sample test report is given below:

Table 5.1: Test Report

|   |   |
|---|---|
| Name                                    | iREQ  |
| Version                                 | 1.0   |
| Author                                  | Amritha K, Anjana B, Bibinraj B S, Harivishnu R, Vishnujith N   |
| Approved By                             | Self  |
| Date                                    | Wednesday, 9th March 2016   |
| Role                                    | User and admin can operate the system.  |
| Prerequisite                            | The user is logged into to system.  |
| User/Actor                              | System Response   |
| Login                                   | User can enter their username and password to login   |
| Run applications                        | After successful authentication, the server resources are made available on the client terminal. User can run those applications. |
| Handle user data                        | The administrator can handle various user data..  |
|   |   |
| The test project was successfully saved |   |

Test case preparation helps the user and the developer to find and fix the errors easily and in advance. Berry Terminal is well tested with the proper test cases and thus passed a better unit test. Elaborated test cases also prepared subjected the system for thorough testing. The test cases prepared are in the following format

### 5.13 BLACK BOX TESTING

Black-box testing is a method of software testing that examines the functionality of an application without peering into its internal structures or workings. Test case preparation helps the user and the developer to find and fix the errors easily and in advance. Disease-Symptom Analyser using MapReduce is well tested with the proper test cases and thus passed a better unit test.

The test cases prepared are in the following format:

Table 5.2: Test Cases

| Test | Use case step | Action performed / user input         | Expected result /system response                            | Actual result | Do expected and actual result correspond |
|------|---------------|---------------------------------------|---|---------------|--|
| 1    | Admin         | Create user                           | Successful creation of user                                 | As Expected   | Yes                                      |
| 2    | User          | Enter username and password           | View success page as the iREQ homepage                      | As Expected   | Yes                                      |
| 3    | User          | Select the application to be accessed | View success page with display of the requested application | As Expected   | Yes                                      |
| 4    | Admin         | Install/Update applications           | Successful installation                                     | As Expected   | Yes                                      |
| 5    | Admin         | Hardware upgrading                    | Successful upgrading  | As Expected   | Yes                                      |
| 6    | User          | Logout                                | View home page of the site                                  | As Expected   | Yes                                      |

## **Chapter 6**

### **FUTURE SCOPE**

In the proposed system all details about the system is designed into an application. It is designed as a web based application, which will work smoothly on any computer system with internet connection. For using/operating the application, there will be no need of additional staff. Any person with basic computer knowledge can use this application. The applications own operational staff is found to be sufficient and efficient to handle the system. Once the system is hosted on the web, the system will be automatically operated. Nowadays people are familiar with internet and e-mail, and hence, they can easily use the options from the home page of the new system. So the proposed system is operationally feasible. The document automation software is also supported by a powerful database where the documents are arranged, making updates and collaboration easy and fast. The system proposed here is meant for use in colleges and for other small scale applications. The basic steps in application processing is same everywhere, and hence the extended version of this system be used in advanced areas too. It can be used at private/ public firms. It can even be used at Government offices where a large number of requests for documents are submitted by the people on a daily basis. At such places, the delay in processing increases with the number of requests, because most of the work like the transfer of the files has to be done manually by the staff.

## **Chapter 7**

# **CONCLUSION**

In this project, an automated document processing system, iREQ has been proposed replacing the current manual document processing system with an attempt to rectify the faults encountered in the present one. To fulfil the requirements, a detailed study has been made on the proposed system in such a way that it is user friendly and easy to use. This particular system has been organized to implement in an attractive manner, so that even a user with minimum knowledge can be able to operate the system easily.

The system is proposed to develop with scalability in mind. Such that, all modules in this system can be tested separately and put together to form the system. Thus the system has fulfilled all the objectives identified and is able to replace the existing system. One of the widest application level advantage of this system is that, the package can be easily upgraded to incorporate new features and extensions fulfilling the requirements of the user. In future, the application will play a dominant role in the whole industry scenario, such that even the small office automation systems will be dependent on the proposed automated document processing system, iREQ.

## REFERENCES

- [1] <http://www.php-mysql-tutorial.com/wikis/mysql-tutorials/> Accessed on 03/01/2016
- [2] <http://howtoprogramwithjava.com/java-ui/> Accessed on 03/01/2016
- [3] <http://docs.oracle.com/javase/tutorial/uiswing/> Accessed on 03/01/2016
- [4] <http://www.cis.upenn.edu/~bcpierce/courses/629/papers/java-tutorial/ui> Accessed on 03/01/2016
- [5] MySQL Tutorial, Luke Welling, Laura Thomson
- [6] Thinking in Java, 3rd Edition, Bruce Eckel
- [7] Fundamentals of Database System(5th Edition), Ramez Elmasri, Shamkanth B Navathe

# GLOSSARY

|            |  |
|------------|--|
| COCOMO     | Constructive Cost Estimation Model         |
| DFD        | Data Flow Diagram                          |
| KLOC       | Kilo Lines of Code                         |
| LOC        | Lines of Code                              |
| UML        | Unified Modeling Language                  |
| HTML       | Hypertext Markup Language                  |
| GUI        | Graphical User Interface                   |
| OS         | Operating System                           |
| WBS        | Work Breakdown Structure                   |
| PC         | Personal Computer                          |
| LAN        | Local Area Network                         |
| SD card    | Secured Digital Card                       |
| IP address | Internet Protocol Address                  |
| SWIG       | Simplified Wrapper and Interface Generator |

## Index

### **B**

Blackbox testing,59

### **C**

COCOMO,10,11

Class diagram,20

Code Testing,54

Cost Estimation,10

Client termial module,15

### **D**

Data Flow Diagram,22

Design,14

### **E**

Economical Feasibility,8

Existing System,3

### **F**

Feasibility Study,6

Features of proposed system,5

Functional Requirements,13

Fail criteria,57

Future Scope,60

### **G**

Gantt Chart,9

Goal of Testing,57

### **H**

Hardware Requirements,12

### **I**

Integration Testing,67

### **J**

Java,45

### **L**

Life Cycle Model,6

Languages and platform used,28

### **M**

Modules,15

Module wise implementation,33

### **N**

Non Functional Requirements,14

Network Management Module,16



**O**

Operational Feasibility,8

Output Testing,69

**P**

Pass Criteria,70

Performance Requirements,14

Proposed System,4

Purpose,2,23

password Testing,54

Pass/Fail Testing,57

**R**

References,75

Regression Testing,69

**S**

Safety Requirements,13

Screen shots,34

Scope,2

Security,17

Security Requirements,15

Sequence Diagram,27

Social Feasibility,8

Software Requirements,12

System testing,67

Social Feasibility,8

System Design,14

Server Management Module,16

Sequence Diagram,18

**T**

Technical Feasibility,8

Test reports,71

**U**

UML diagrams, 23

Use case diagram,16

Unit Testing,67

**V**

Validation Testing, 55