

A
Minor Project Report
on
E-CARE SYSTEM
Submitted in Partial Fulfillment of
the Requirements for the Third Year
of
Bachelor of Engineering
in
Computer Engineering
to
North Maharashtra University, Jalgaon

Submitted by

Vinaya Patil
Shiwani Suryawanshi
Sneha Suryawanshi
Madhura Raverkar

Under the Guidance of

Mr. S.D. Rajput



DEPARTMENT OF COMPUTER ENGINEERING
SSBT's COLLEGE OF ENGINEERING AND TECHNOLOGY,
BAMBHORI, JALGAON - 425 001 (MS)
2015 - 2016

**SSBT's COLLEGE OF ENGINEERING AND TECHNOLOGY,
BAMBHORI, JALGAON - 425 001 (MS)
DEPARTMENT OF COMPUTER ENGINEERING**

CERTIFICATE

This is to certify that the minor project entitled *E-Care System*, submitted by

**Vinaya Patil
Shiwani Suryawanshi
Sneha Suryawanshi
Madhura Raverkar**

in partial fulfillment of the Third Year of *Bachelor of Engineering* in *Computer Engineering* has been satisfactorily carried out under my guidance as per the requirement of North Maharashtra University, Jalgaon.

Date: April 11, 2016

Place: Jalgaon

Mr. S.D. Rajput
Guide

Prof. Dr. Girish K. Patnaik
Head

Prof. Dr. K. S. Wani
Principal

Acknowledgements

It is indeed a great pleasure and sense of fulfillment for me to present a seminar report on “E-Care System ”.I acknowledge a deep sense of gratitude for their constant inspiration and helpful criticism. I also thank our Principal Dr. K.S. Wani, H.O.D. Dr. Girish K. Patnaik , Mr S.D. Rajput to work under the versatile and kind guidance and all staff members for providing necessary inputs at the right time. Last but not least, no words enough to express heartiest gratitude to our beloved parents for their assistance in building up our carrier. Thanking in anticipation all the readers.

Vinaya Patil

Shiwani Suryawanshi

Sneha Suryawanshi

Madhura Raverkar

Contents

Acknowledgements	ii
Abstract	1
1 Introduction	2
1.1 Background	2
1.2 Motivation	3
1.3 Problem Defination	3
1.4 Scope	5
1.5 Summary	5
2 System analysis	6
2.1 Literature survey	6
2.2 Proposed System	6
2.3 Feasibility Study	7
2.3.1 Economical feasibilty	7
2.3.2 Operational Feasilbilty	7
2.3.3 Technical feasibilty	7
2.4 Summary	8
3 System requirement specification	9
3.1 Hardware requirement	9
3.2 Software Requirement	9
3.3 Functional Requirements	9
3.4 Non-Functional Requirements	10
3.5 Summary	10
4 System Design	11
4.1 System architecture	11
4.2 E-R diagram	12
4.3 Database design	15
4.3.1 Schema	15

4.4	Data Flow diagram	18
4.5	UML Diagrams	18
4.5.1	Use case diagram	21
4.5.2	Class Diagram	22
4.5.3	Sequence Diagram	22
4.5.4	Component Diagram	22
4.5.5	Deployment Diagram	23
4.6	Summary	23
5	Implementation	31
5.1	Implementation deetails	31
5.2	Implementation environment	32
5.2.1	About Java	32
5.2.2	About JDBC	33
5.2.3	About Servlets	33
5.3	Summary	34
6	System Testing	35
6.1	How to implement testing	35
6.2	Test cases and test results	37
6.3	summary	37
7	Result and Analysis	38
7.1	Analysis	38
8	Conclusion and future scope	44
8.1	Conclusion	44
8.2	Future scope	44
	Bibliography	45

List of Tables

4.1	Login	15
4.2	HDLevels	15
4.3	registration	15
4.4	superuserprofile	16
4.5	HDEmployee	16
4.6	Problemrequest	16
4.7	HDcompanyinfo	17
4.8	Categoryies	17
4.9	Module	17
4.10	supportteam	17
4.11	schedules	17
4.12	forward	17
4.13	HDmessages	18
4.14	Problemhistory	18

List of Figures

4.1	E-R DIAGRAMS OF E-Care	12
4.2	Flow for Posting Query	13
4.3	Flow for forwarding the query to the levels	14
4.4	Zero Level	19
4.5	Context Level	19
4.6	first level	19
4.7	Second Level	20
4.8	Corporate User And Corporate Client Use Case Diagram	21
4.9	Super User Use Case Diagram	21
4.10	Class Diagram of E-care help desk system	22
4.11	Corporate Client Sequence Diagram	24
4.12	Corporate Client Sequence Diagram	25
4.13	Super user Sequence Diagram	26
4.14	Level One Administrator Sequence Diagram	27
4.15	Level Two Administrator Sequence Diagram	28
4.16	Level Three Administrator Sequence Diagram	29
4.17	Component Diagram of E-care	30
4.18	Deployment Diagram of E-care	30
7.1	Corporate User And Corporate Client Use Case Diagram	39
7.2	Corporate User And Corporate Client Use Case Diagram	40
7.3	Corporate User And Corporate Client Use Case Diagram	40
7.4	Corporate User And Corporate Client Use Case Diagram	41
7.5	Corporate User And Corporate Client Use Case Diagram	41
7.6	Corporate User And Corporate Client Use Case Diagram	42
7.7	Corporate User And Corporate Client Use Case Diagram	42
7.8	Corporate User And Corporate Client Use Case Diagram	43
7.9	Corporate User And Corporate Client Use Case Diagram	43

Abstract

E-care is customer care management system. E-care is called as E-desk or Help Desk System. It is so called as Help Desk System because it tries to solve all the HD (helpdesk) problems, which are coming from the users of E-care. E-care will take care of every request coming from their users and try to solve and produce the solution of the particular request. E-care also will store the responses for the future use. E-care contains six main members who play very important role in this Help Desk System. They include super user, level1 administrator, level2 administrator, level3 administrator, corporate client user and corporate user (or end-user). E-care is very useful for the corporate clients who want their business to extend world wide i.e., by keeping contact and good relation with the corporate users who buy their products. The corporate clients who buy E-care for their business can know what are the various problems of their product which are coming from their users so that they improve their products to the user needs. Any user who knows a little about of the computer can handle e-care.

Chapter 1

Introduction

In this chapter basic introduction about E-care help desk system along with its need and objectives has been described.

Section 1.1 describes the background of E-care help desk system. Section 1.2 and 1.3 describes the limitations in earlier system and the need of E-care help desk system. Scope of this project has been described in section 1.4.

1.1 Background

E-care is called as E-desk or Help Desk System. It is so called as Help Desk System because it tries to solve all the HD (helpdesk) problems, which are coming from the users of E-care. E-care will take care of every request coming from their users and try to solve and produce the solution of the particular request. E-care also will store the responses for the future use.

E-care contains six main members who play very important role in this Help Desk System. They include super user, level1 administrator, level2 administrator, level3 administrator, corporate client user and corporate user (or end-user).

Super user is the chief head of the organization that assigns first level, second level, and third level administrators. The super user is responsible for all the three level administrators including the corporate clients. The Super user is the sole person who is responsible for creating categories, modules and the support team.

Super user creates new categories only if HD problems dont belong to those categories present in E-care and similarly with the creation of modules. When the super user creates new categories and modules he will label these categories and modules as levels and appoint one administrator for each of them. When the super user gets the HD problem as request he can modify the request or can answer it by sending solution to the level3 or the support team

who is responsible for messaging the response or the solution to the end-user (corporate user).

Corporate client is the one who buys E-care product for his business purpose. He can create corporate users for his product. Corporate user is the one who has to register himself to the corporate client before he submits the HD problems to the super user and corporate user is the actual end-user of the product or service.

Corporate user will register to the corporate client and corporate client sends all the registration details to the super user, and after the registration the end-user will submit the HD problems to the super user. When the super user gets the HD problems he will check the details of the end-user and send the responses to the support team who will message the responses to the corporate users.

E-care is very useful for the corporate clients who want their business to extend world wide i.e., by keeping contact and good relation with the corporate users who buy their products. The corporate clients who buy E-care for their business can know what are the various problems of their product which are coming from their users so that they improve their products to the user needs. Any user who knows a little about of the computer can handle e-care.

1.2 Motivation

The interface is designed such a way that the end users find it easy to access. The system aim to provide a complete problem solving approach help desk.

The system aims to automate the functions performed by the super user. Operations such as registering the new corporate client, creating corporate users, posting the problems, creating support team, registering the employees in each level, resolving the problems etc. are included in this system. The main objective of E-care help desk system is to provide powerful functionalities to the user by making use of simple interface.

1.3 Problem Defination

This project mainly deals with automating the task of the E-care system that helps many end-users to know the solutions for their request within no time. E-care can be done manually but there may be problems that have to be faced so to overcome such problems We need to automate the Help Desk System.

if the system is maintaining the list of all the details of corporate users and corporate clients manually it will take a lot of time to retrieve even single information such as retrieving or deleting single information becomes very difficult task.

The updations for the list such as creating a new creating new end-user details or creating new corporate client details and including the details of the categories and the modules that are created newly by the super user becomes very difficult manually. Once all information is provided manually it become quit difficult for updating the records with new information manually, so we automate the system to overcome all such difficulties.

E-care, which is developed manually, is the existing system. This system generates and delivers the services to the end-users, which depends on the response provided by the support team who actually message the responses to the corporate users for their requests.

By manually the Help Desk System will face difficulties while updations and it is time consuming and leads more errors. E-care automates the tedious job of maintaining the corporate user details or corporate client details and manages the updations and see that the responses are sent to the corporate user as soon as possible. It includes main members like:

1. Super user
2. Corporate client
3. Corporate user
4. Level1 administrator
5. Level2 administrator
6. Level3 administrator or support team.

Super user will manage the entire organization that includes corporate clients, corporate users, and level administrators and also he can create the new categories and also the modules.

Proposed systems are those that are automated, so that it is easy to retrieve the responses from the system fastly and updating the details once the response or services are provided to the end-users upon their request without any difficulty and saves time.

1.4 Scope

Following are the scope of this project:

- E-care help desk system provides information about the organisation.
- The system can be accessed by number of users.
- Each user is provided with separate user id and password for secure access.
- Each user is assigned a different set of permission for each module of help desk system
- Fine session is under control of super user and level administrators.

1.5 Summary

This chapter deals with basic introduction of E-care help desk system,its scope and its problem defination.

Chapter 2

System analysis

System Analysis is first stage according to System Development Life Cycle model. This System Analysis is a process that starts with the analyst.

In this chapter details of earlier system and advantages of proposed system, has been described.

Section 2.1 and 2.2 describes the literature survey and earlier proposed systems and advantages of new help desk system over former one. Feasibility of this system has been described in section 2.3.

2.1 Literature survey

Analysis is a detailed study of the various operations performed by a system and their relationships within and outside the system. One aspect of analysis is defining the boundaries of the system and determining whether or not a candidate should consider other related systems. During analysis, data is collected from the available files, decision points, and transactions handled by the present system.

Logical system models and tools are used in analysis. Training, experience, and common sense are required for collection of the information

2.2 Proposed System

Proposed system are those that are automated, so that it is easy to retrieve the responses from the system fastly and updating the details once the response or services are provided to the end-users upon their request without any difficulty and saves time.

2.3 Feasibility Study

All projects are feasible, given unlimited resources and infinite time. But the development of software is plagued by the scarcity of resources and difficult delivery rates. It is both necessary and prudent to evaluate the feasibility of a project at the earliest possible time. Three key considerations are involved in the feasibility analysis.

2.3.1 Economical feasibility

This procedure is to determine the benefits and savings that are expected from a candidate system and compare them with costs. If benefits outweigh costs, then the decision is made to design and implement the system. Otherwise, further justification or alterations in proposed system will have to be made if it is to have a chance of being approved. This is an ongoing effort that improves in accuracy at each phase of the system life cycle.

2.3.2 Operational Feasibility

Operations done in the E-care are very fast and are able to succeed in sending the responses very fastly. Updations are done very quickly that the modified data is stored in the database in no time. E-care supports up to five hundred simultaneously users against the central database at any given time. E-care must be able to complete 80percent of all transactions within two minutes.

2.3.3 Technical feasibility

E-Care is the product that is designed mainly on Java. The main reason behind choosing the Java environment is that, as the users. In order to post the queries to related Organization always require Input screen which usually provides services all around the time. Such environment that always gives instant reply to the users should support input screen. These requirements are satisfied by Java, as it is multi platform environment supports webBrowser where as other languages lack these features.

In order user to continuously communicate he/she needs a reliable protocol i.e., is TCP/IP protocol. This protocol frequently transfers messages between client and user. Communication Standards such as HTTP is used .An HTTP is a client/server interaction consists of a single request/reply interchange, it user-friendly environment.

2.4 Summary

This chapter focuses on the previous help desk systems and new proposed E-care help desk system and also focuses on its feasibility study.

Chapter 3

System requirement specification

3.1 Hardware requirement

Hard ware Specification:

- Processor: Intel P-III based system
- Processor Speed :250 MHz to 833MHz
- RAM:64MB to 256MB
- Hard Disk:2GB to 30GB

3.2 Software Requirement

- Language:JDK 1.2
- Database:SQL
- Operating System:WindowsNT/95/98/2000

3.3 Functional Requirements

1. **Register new corporate client** This feature can be performed by the new help desk user who wants to take facilities provided by the E-care. The client must have to register himself if he is not the user of E-care help desk system. Following are the **Functional Requirements**:

- (a) User must have to fill the required information.
- (b) System must be able to verify the provided information.
- (c) System must be able to unauthorise the user if it finds something wrong in user.

2. **Register level administrators** This feature is performed by the super user. Super user creates three administration levels viz. level1, level2, level3 etc. **Functional Requirements** are as follows:

- Super user adds employees to various levels according to the modules.
- Each employee is provided with separate user id and password to solve the posted problems of corporate clients and users according to their corresponding levels.
- Super user must be able to remove any employee at any moment.

3. **Posting the problem** This feature is performed by corporate clients and users. The posted problems are received by super user. Super user then distributes each problem to its corresponding level employees.

3.4 Non-Functional Requirements

- **Efficiency Requirements** When E-care help desk system is implemented users and employees can easily communicate with each other through problems posted by users.
- **Reliability Requirements** The system should accurately perform user registration, employee registration, employee validation, problem posting, problem solving etc.
- **Usability Requirements** The system is designed for a user friendly environment so that organisation and its users can perform various tasks easily and in an effective way.
- **Implementation Requirements** In implementing whole system it uses HTML in front end with java as a server side scripting language which will be used for database connectivity and the backend i.e. the database part is developed in oracle.
- **Delivery Requirements** The whole system is expected to deliver in two months with a weekly evolution by the project guide.

3.5 Summary

This chapter deals with the hardware and software requirements and also functional and non functional requirements of the system.

Chapter 4

System Design

This chapter deals with the design of E-care system. In section 4.1 system's architecture is explained, in section 4.2 E-R diagrams are given, in section 4.3 database design is given, section 4.4 covers Dataflow diagrams while section 4.5 covers UML diagrams.

4.1 System architecture

E-Care [help desk system] is a customer care management system. The following are the modules related to E-Care [help desk system] are as follows: -

1. User module
2. Super User module
3. Administration module

1. User Module: In this module user can register themselves to the e-Care if they are new users, otherwise they can login directly and can then post their queries to e-Care or e-desk. First user has to register with corporate client. In order to send request the end-user has to login and the login details are stored in the database maintained by e-Care system. Database checks the details of the user, if he is valid then allows the user to access the help desk system.

1. Super User module: In this module Super user is the chief head of the Organization who assigns 1st level and 3rd level administrators. The super user is responsible for all the three levels administrators including the corporate clients. The super user is the sole person who is responsible for creating categories, modules and support team. Super User will take all the registration details of the Corporate Users so as to check with the queries which in turn consist of the query and also the userid. Super user has the right to remove any level or add any level and also responsible for creating modules and categories .

4.2 E-R diagram

E-R (Entity-Relationship) Diagram is used to represents the relationship between entities in the table.

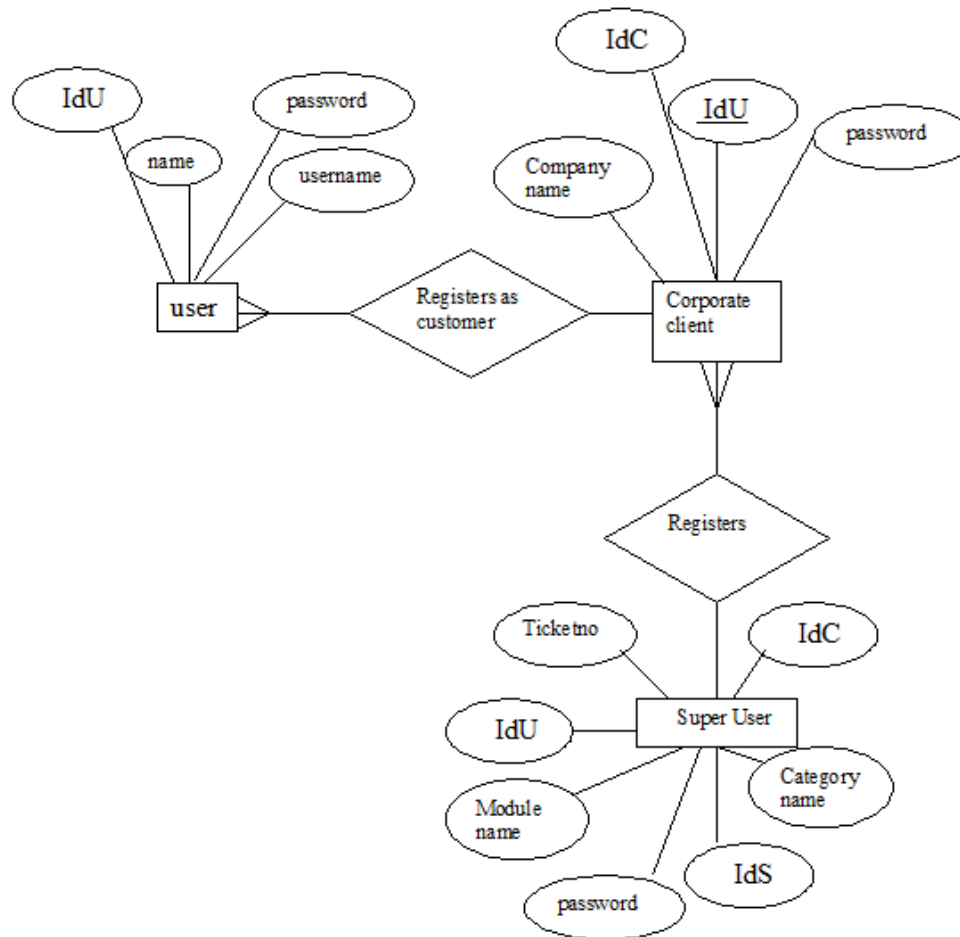


Figure 4.1: E-R DIAGRAMS OF E-Care

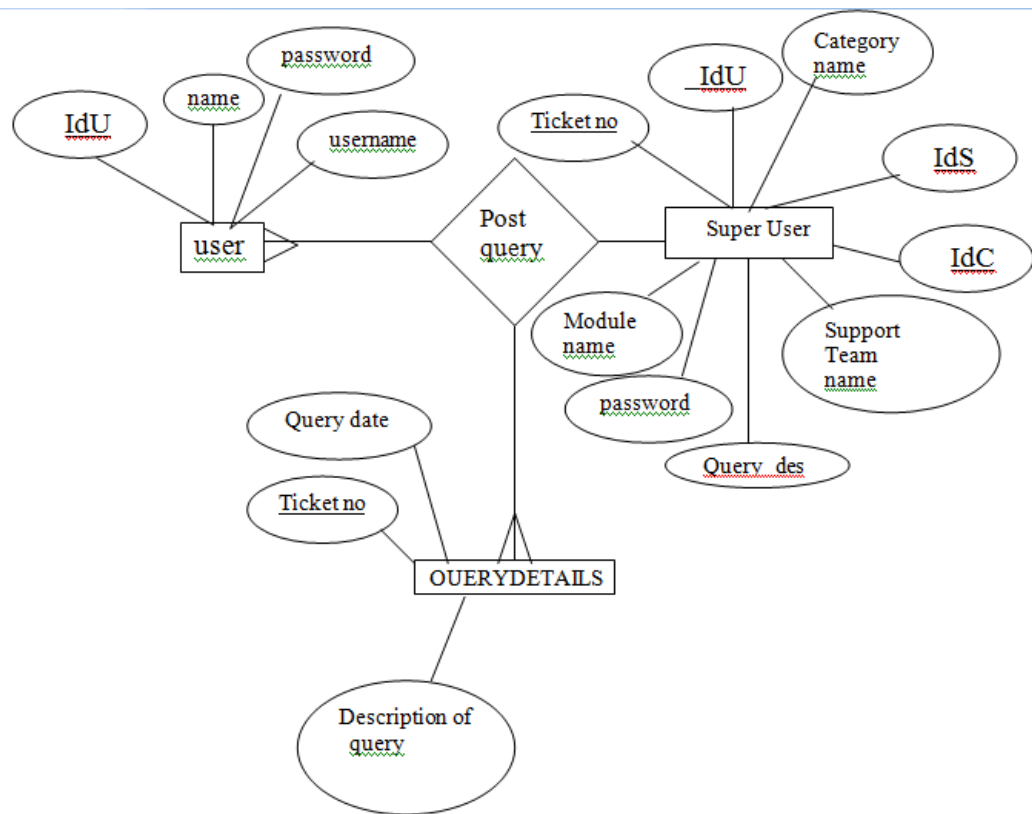


Figure 4.2: Flow for Posting Query

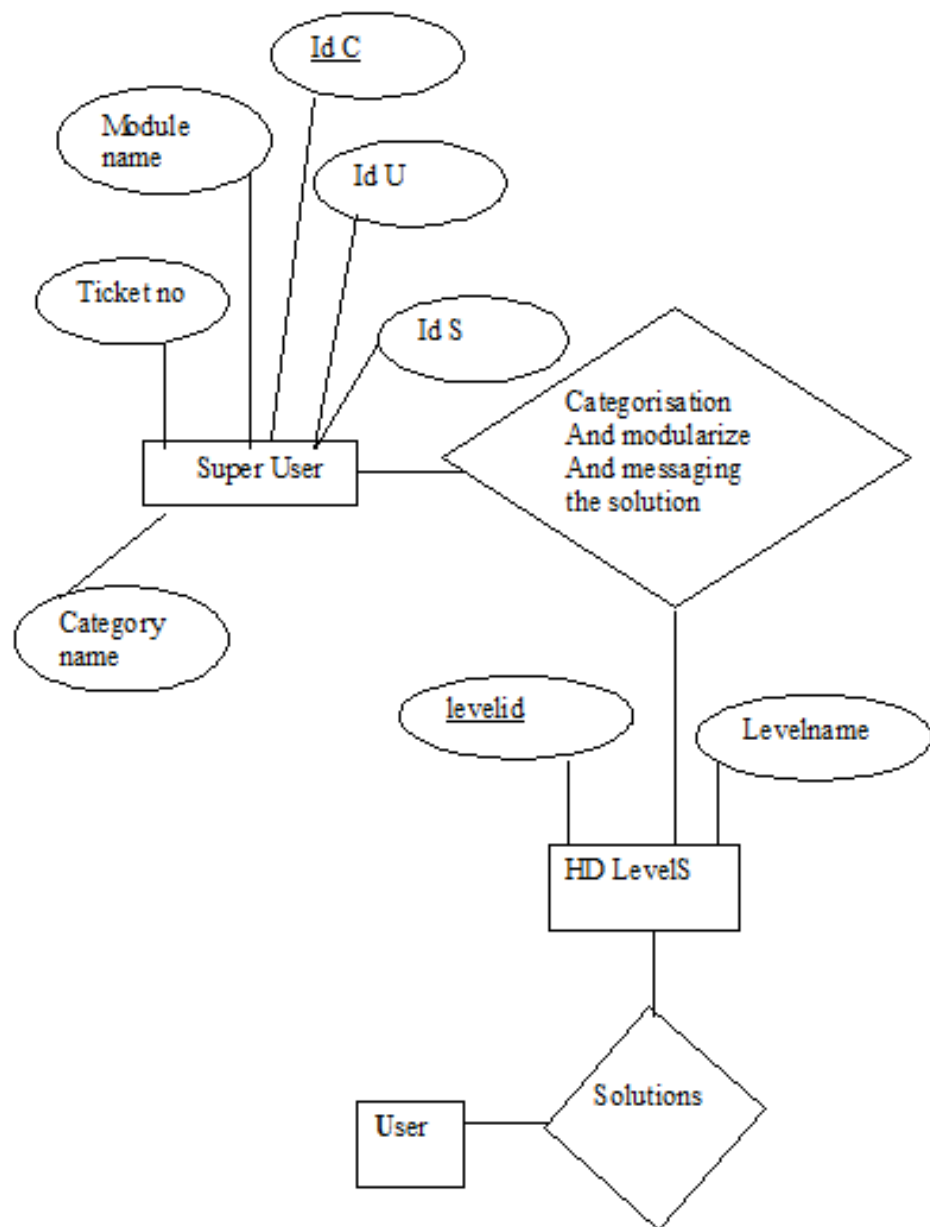


Figure 4.3: Flow for forwarding the query to the levels

4.3 Database design

4.3.1 Schema

Table 4.1: Login

Fieldname	Type
username	varchar2(30)
password	varchar2(30)

Table 4.2: HDLevels

Fieldname	Type
levelcode	varchar2(30)
levelname	varchar2(30)

Table 4.3: registration

Fieldname	Type
firstname	varchar2(30)
lastname	varchar2(30)
address	varchar2(30)
phoneno	number
city	varchar2(30)
state	varchar2(30)
country	varchar2(30)
gender	varchar2(30)
zipcode	number
emailid	varchar2(30)
ctype	varchar2(30)
userid	varchar2(30)
password	varchar2(30)
confirmation password	number
hintquestion	Type
hintsolution	varchar2(30)

Table 4.4: superuserprofile

Fieldname	Type
firstname	varchar2(30)
lastname	varchar2(30)
address	varchar2(30)
phoneno	number
city	varchar2(30)
state	varchar2(30)
country	varchar2(30)
gender	varchar2(30)
zipcode	number
emailid	varchar2(30)
ctype	varchar2(30)
userid	varchar2(30)
password	varchar2(30)
confirmation password	number
hintquestion	Type
hintsolution	varchar2(30)

Table 4.5: HDEmployee

Fieldname	Type
userid	varchar2(30)
employeenname	varchar2(30)
address	varchar2(30)
phoneno	number
city	varchar2(30)
state	varchar2(30)
country	varchar2(30)
gender	varchar2(30)
zipcode	number
emailid	varchar2(30)

Table 4.6: Problemrequest

Fieldname	Type
ticket	number
categoryid	varchar2(30)
moduleid	varchar2(30)
problemdescription	varchar2(30)
lastsolution	varchar2(30)
userid	varchar2(30)
problemreportdate	varchar2(30)
forward	varchar2(30)
reply	number

Table 4.7: HDcompanyinfo

Fieldname	Type
locationid	varchar2(30)
locationname	varchar2(30)
companyname	varchar2(30)
userid	number
address	varchar2(30)
state	varchar2(30)
country	varchar2(30)
gender	varchar2(30)
zipcode	number

Table 4.8: Categoryies

Fieldname	Type
categoryid	varchar2(30)
categoryname	varchar2(30)
status	varchar2(30)

Table 4.9: Module

Fieldname	Type
moduleid	varchar2(30)
categoryid	varchar2(30)
modulename	varchar2(30)
status	varchar2(30)

Table 4.10: supportteam

Fieldname	Type
supportid	varchar2(30)
moduleid	varchar2(30)
categoryid	varchar2(30)
supportname	varchar2(30)

Table 4.11: schedules

Fieldname	Type
ticket	number
scheduledate	varchar2(30)

Table 4.12: forward

Fieldname	Type
ticket	number
fromname	varchar2(30)
toname	varchar2(30)
status	varchar2(30)

Table 4.13: HDmessages

Fieldname	Type
toname	varchar2(30)
fromname	varchar2(30)
subject	varchar2(30)
message	number
sentdate	varchr2(30)

Table 4.14: Problemhistory

Fieldname	Type
ticket	number
problemdate	varchar2(30)
problemdescription	varchar2(30)
solution	number
solutionby	varchr2(30)
solutiondate	varchar2(30)
status	varchar2(30)

4.4 Data Flow diagram

The data flow diagram is used for classifying system requirements to major transformation that will become programs in system design. This is starting point of the design phase that functionally decomposes the required specifications down to the lower level of details. It consists of a series of bubbles joined together by lines. Bubbles: Represent the data transformations. Lines: Represents the logic flow of data.

Data can trigger events and can be processed to useful information. System analysis recognizes the central goal of data in organizations. This dataflow analysis tells a great deal about organization objectives are accomplished.

Dataflow analysis studies the use of data in each activity. It documents this finding in DFDs. Dataflow analysis give the activities of a system from the viewpoint of data where it originates how they are used or hanged or where they go, including the stops along the way from their destination. The components of dataflow strategy span both requirements determination and systems design. The first part is called dataflow analysis.

As the name suggests, we didnt use the dataflow analysis tools exclusively for the analysis stage but also in the designing phase with documentation.

4.5 UML Diagrams

The Unified Language allows software engineer t express an analysis model using the modelling notation that is govern by the set of syntatic.semantic and pragmatic rules.

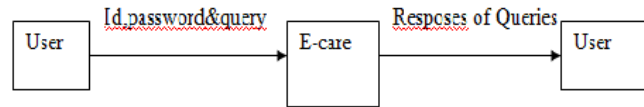


Figure 4.4: Zero Level

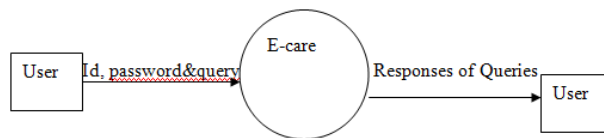


Figure 4.5: Context Level

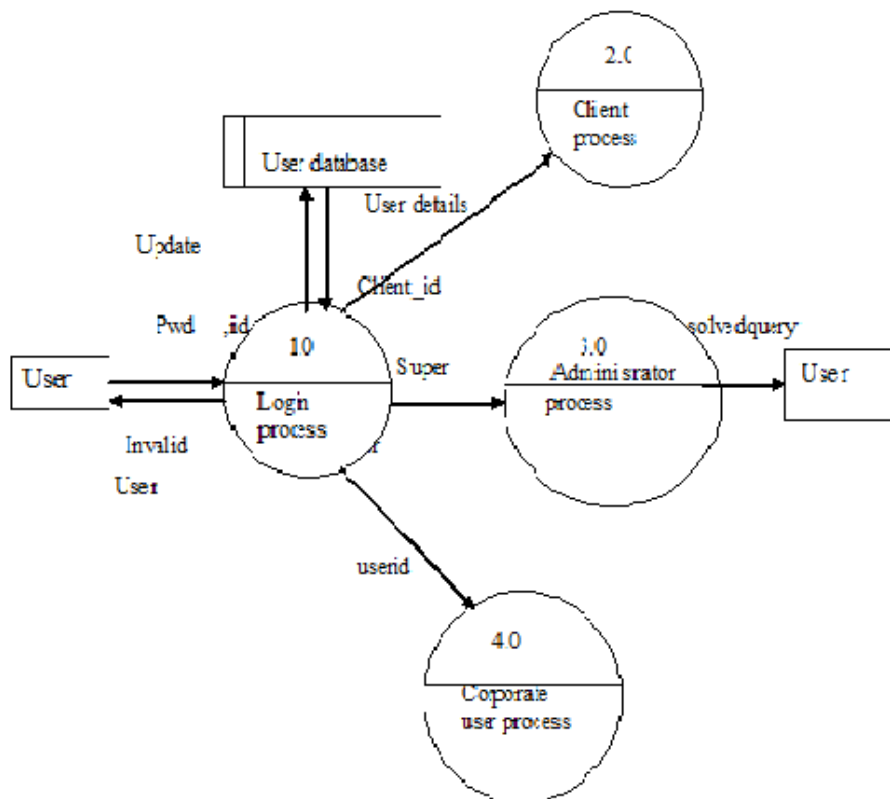


Figure 4.6: first level

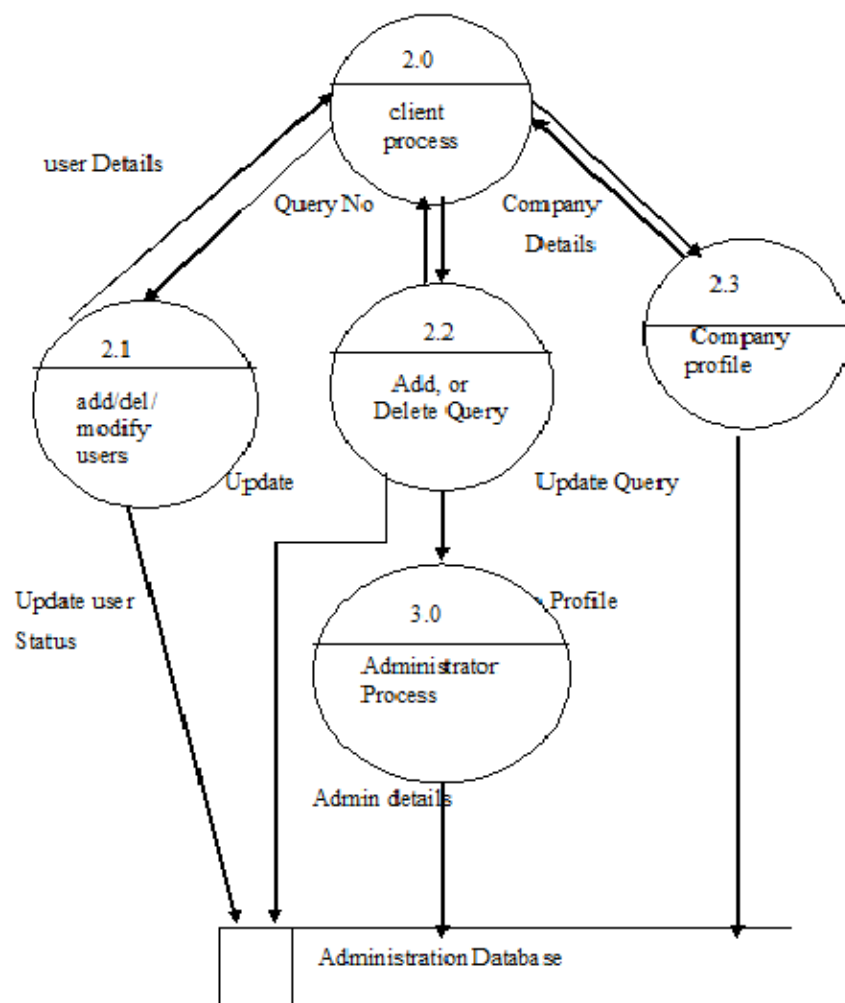


Figure 4.7: Second Level

4.5.1 Use case diagram

In software and systems engineering, a use case is a list of actions or event steps, typically defining the interactions between a role (known in the Unified Modeling Language as an actor) and a system, to achieve a goal. The actor can be a human, an external system, or time refer fig 7.5 and fig 7.6

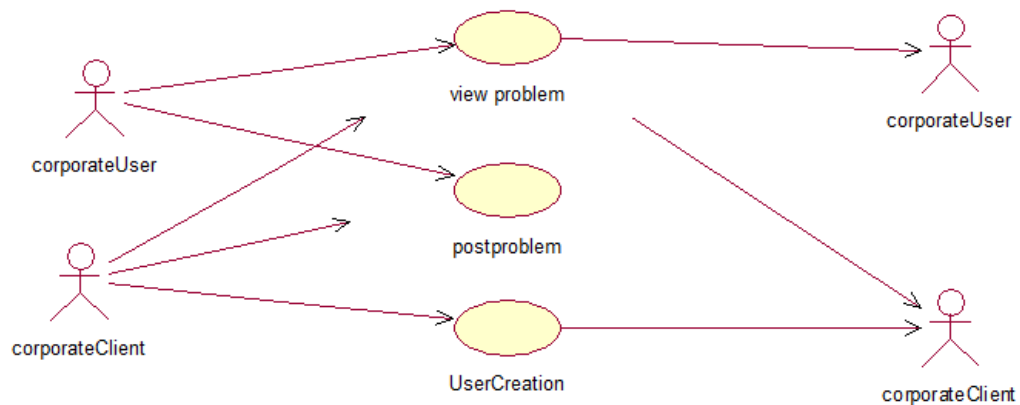


Figure 4.8: Corporate User And Corporate Client Use Case Diagram

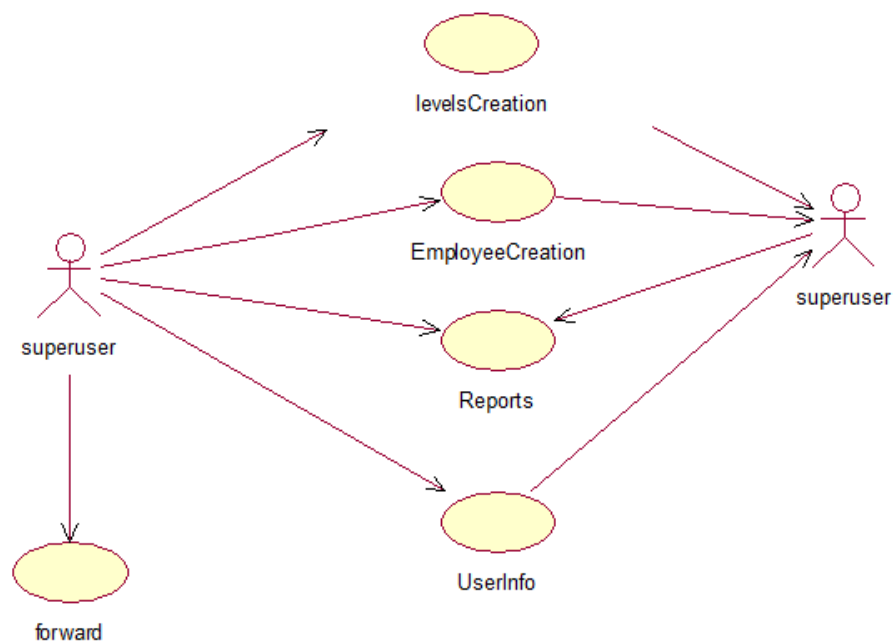


Figure 4.9: Super User Use Case Diagram

4.5.2 Class Diagram

In software engineering, a class diagram in the Unified Modeling 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 refer fig 7.7

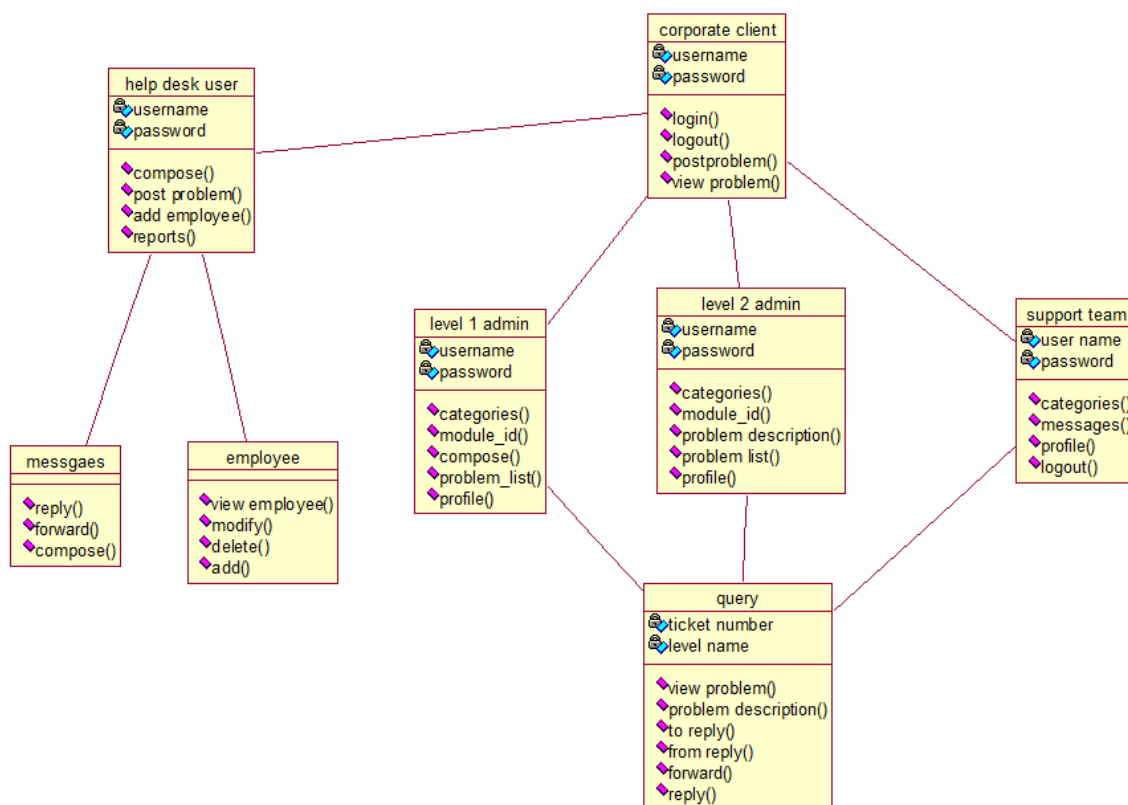


Figure 4.10: Class Diagram of E-care help desk system

4.5.3 Sequence Diagram

A Sequence diagram is an interaction diagram that shows how processes operate with one another and in what order. It is a construct of a Message Sequence Chart. A sequence diagram shows object interactions arranged in time sequence refer fig 7.8, fig 7.9,fig 4.13,fig 4.14 and fig 4.15

4.5.4 Component Diagram

Component diagram is a special kind of diagram in UML. The purpose is also different from all other diagrams discussed so far. It does not describe the functionality of the system but it describes the components used to make those functionalities refer fig 4.17

4.5.5 Deployment Diagram

Deployment diagram shows execution architecture of systems that represent the assignment (deployment) of software artifacts to deployment targets (usually nodes) refer fig 4.18

4.6 Summary

In this way chapter 4 covers all the designing aspects of E-care Help Desk system.

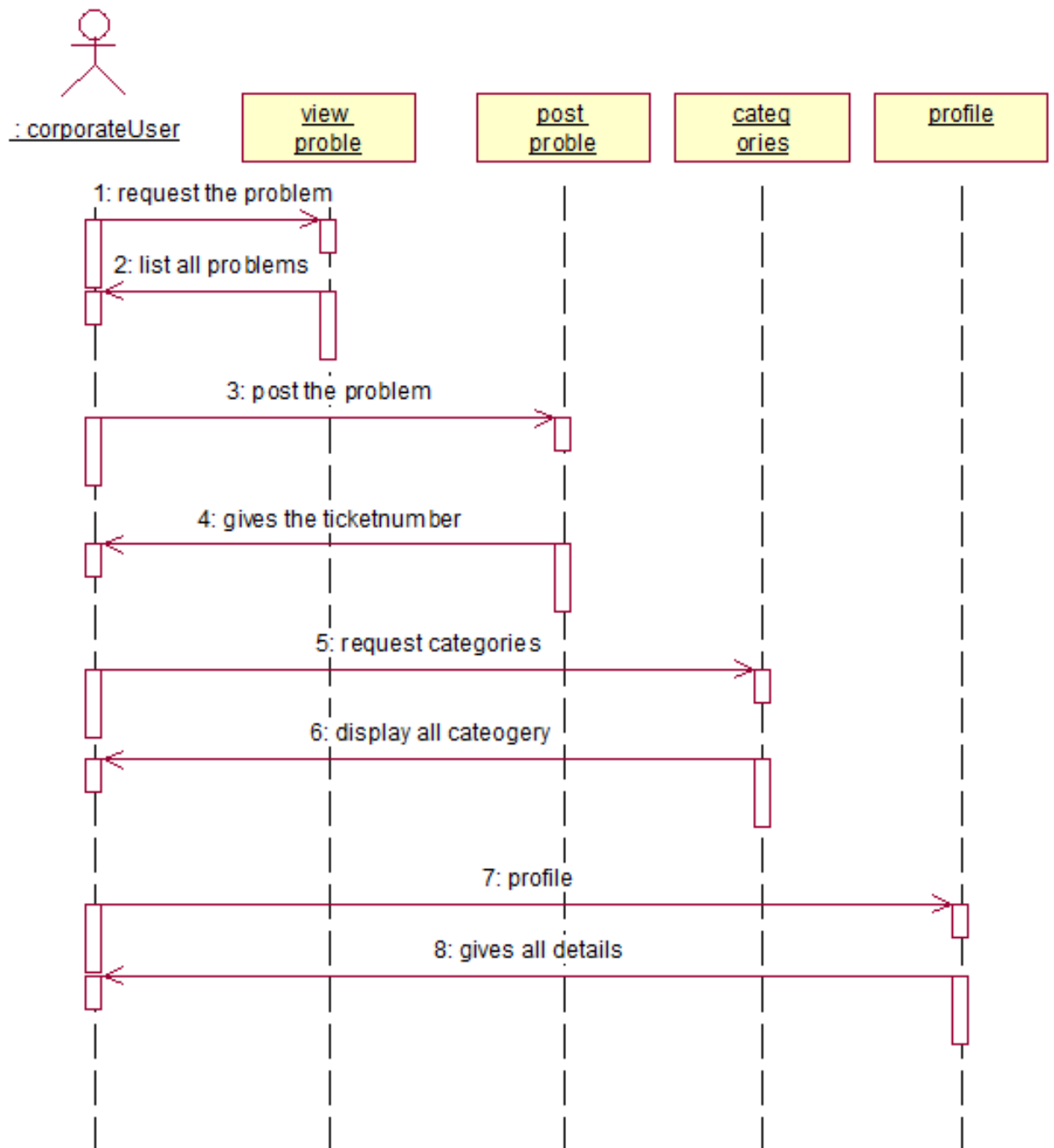


Figure 4.11: Corporate Client Sequence Diagram

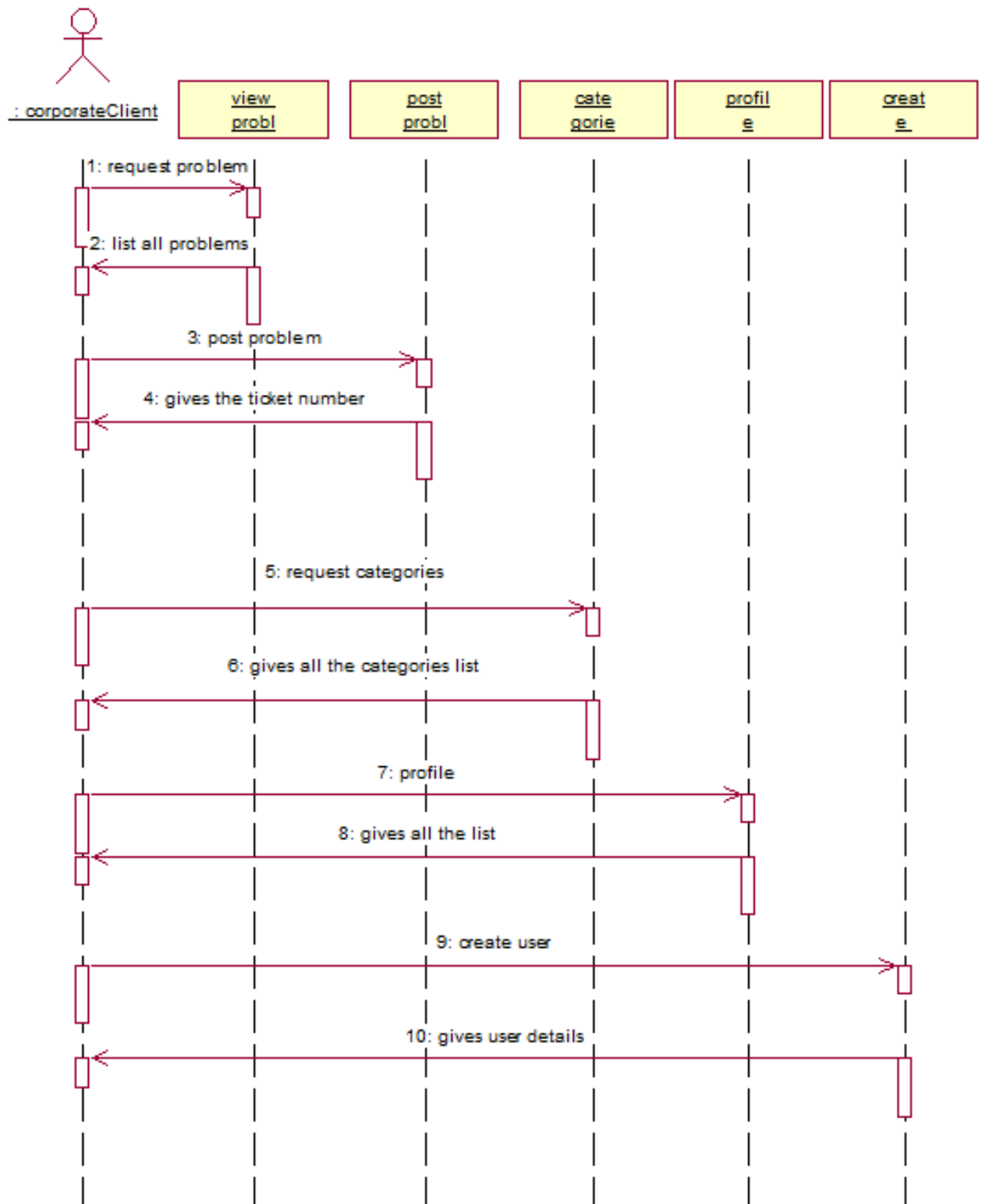


Figure 4.12: Corporate Client Sequence Diagram

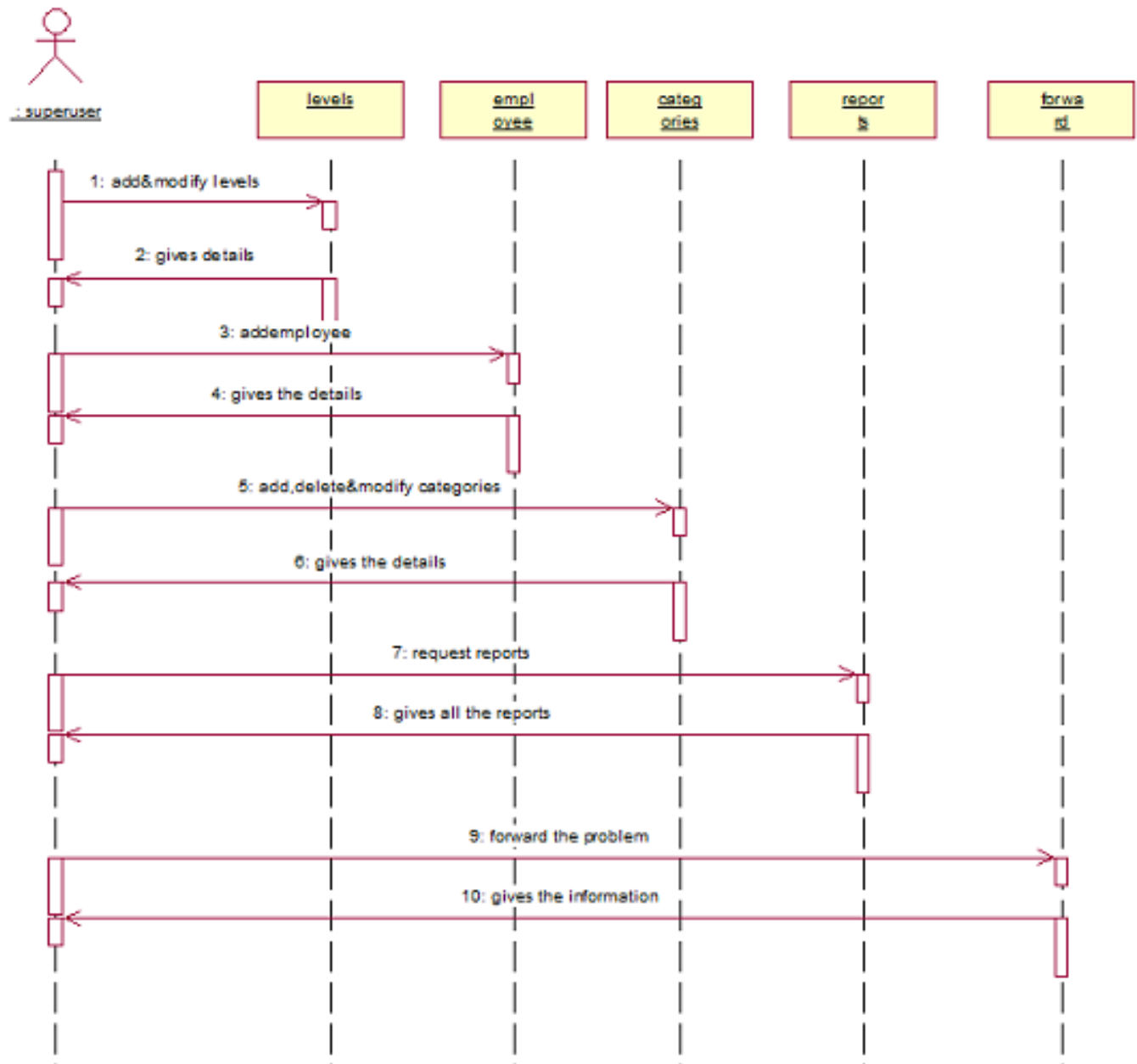


Figure 4.13: Super user Sequence Diagram

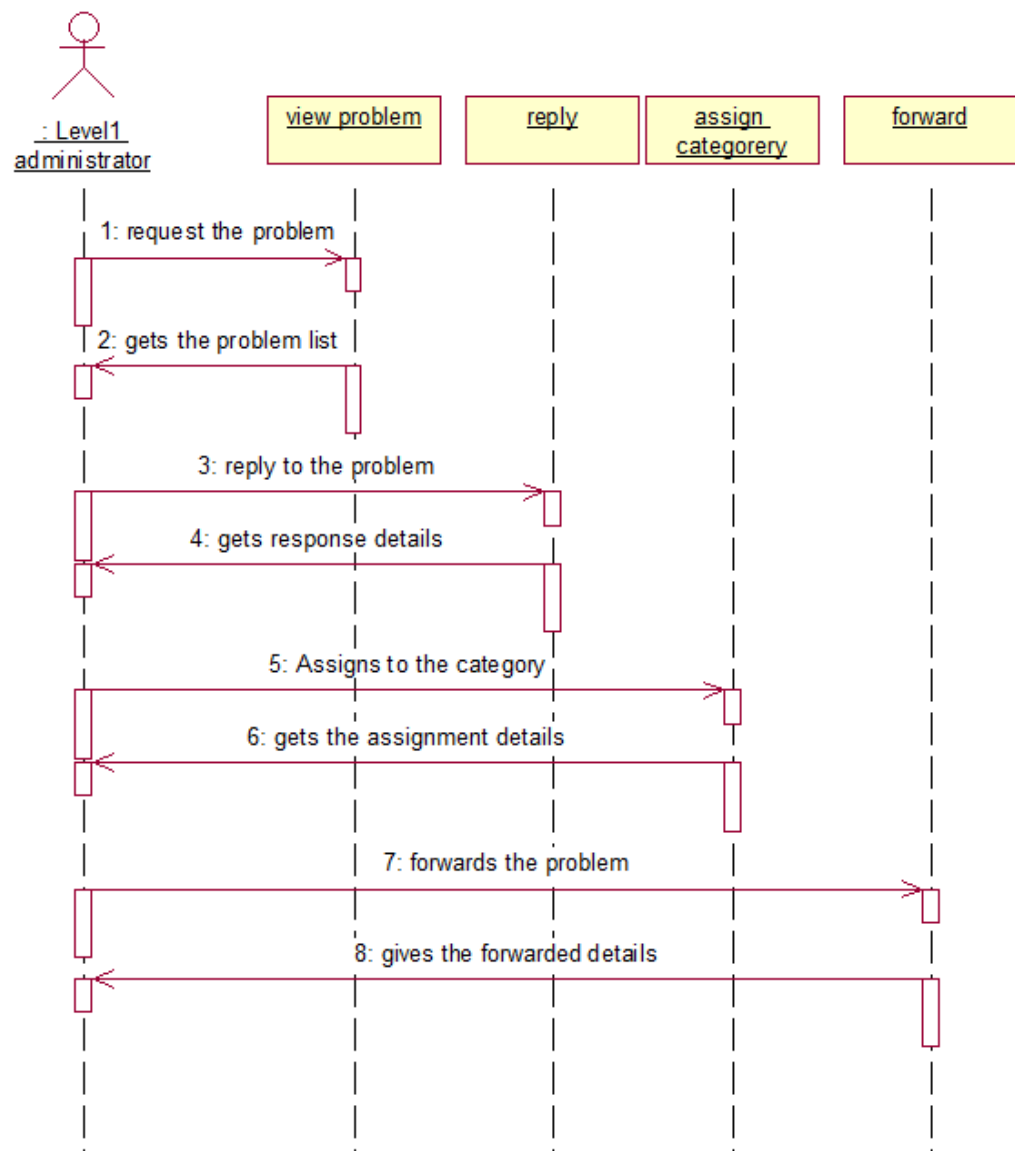


Figure 4.14: Level One Administrator Sequence Diagram

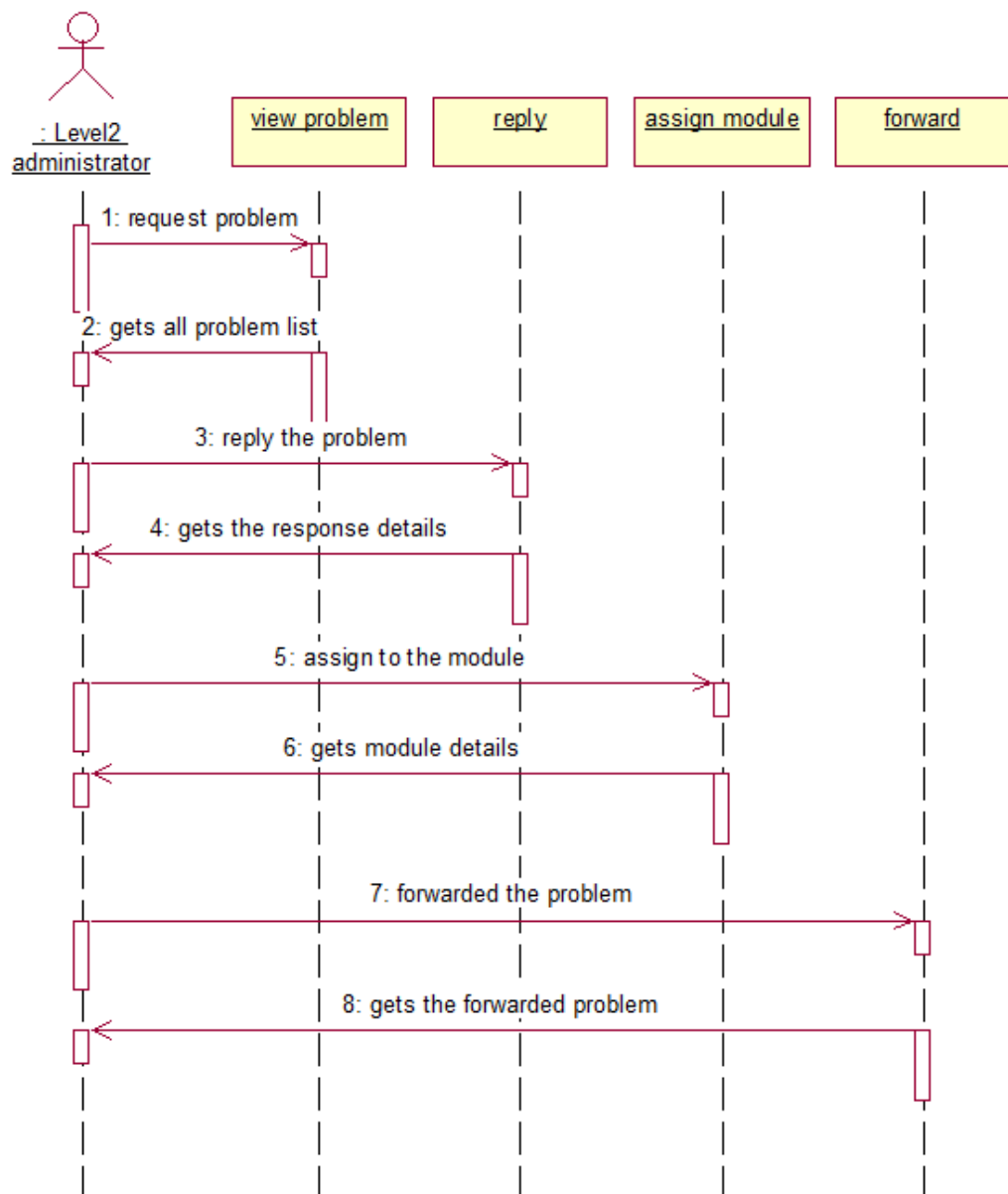


Figure 4.15: Level Two Administrator Sequence Diagram

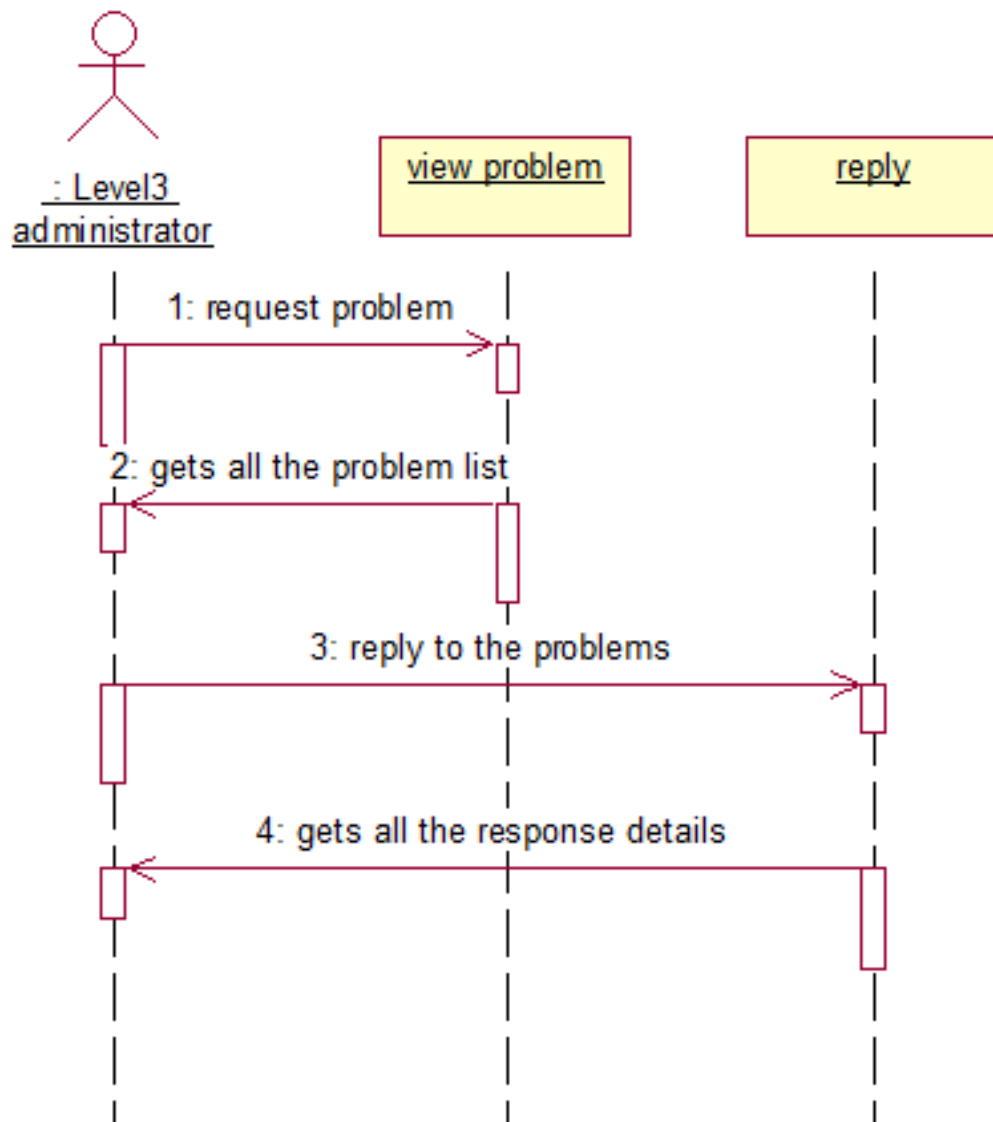


Figure 4.16: Level Three Administrator Sequence Diagram

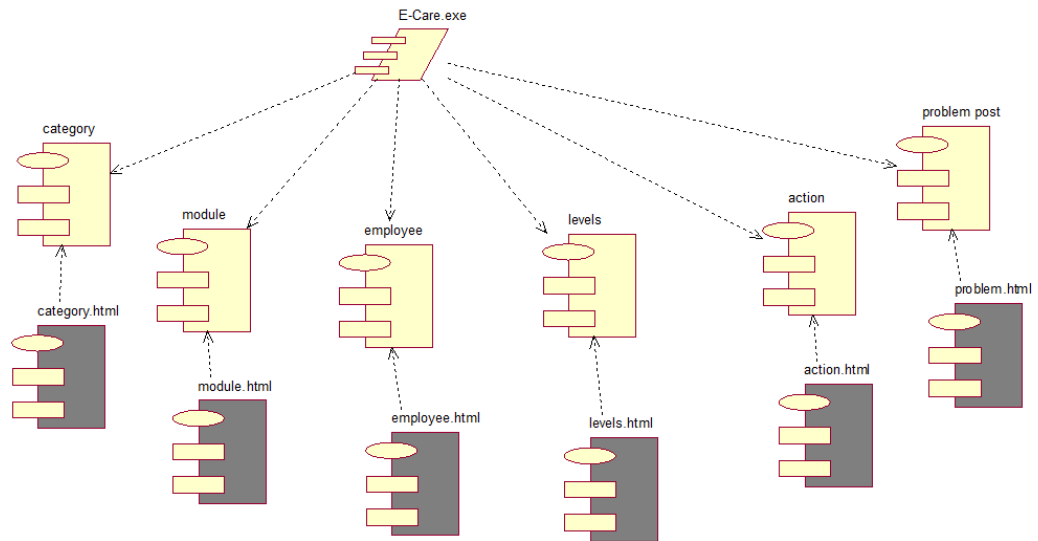


Figure 4.17: Component Diagram of E-care

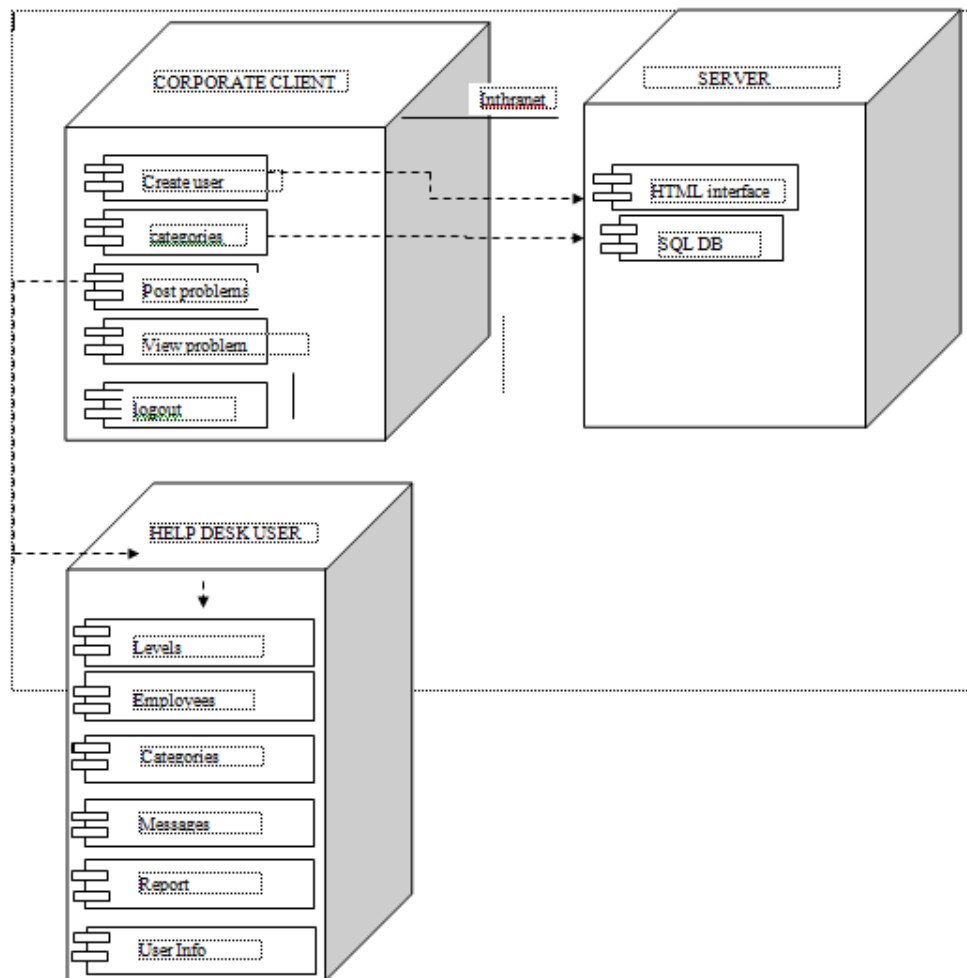


Figure 4.18: Deployment Diagram of E-care

Chapter 5

Implementation

This chapter mainly deals with the implementation schemes of E-care system. Section 5.1 covers implementation details, section 5.2 discusses about various implementation environment.

5.1 Implementation details

This project mainly deals with automating the task of the E-care system that helps many end-users to know the solutions for their request within no time. E-care can be done manually but there may be problems that have to be faced so to overcome such problems we need to automate the Help Desk System.

If the system is maintaining the list of all the details of corporate users and corporate clients manually it will take a lot of time to retrieve even single information such as retrieving or deleting single information becomes very difficult task.

The updations for the list such as creating a new creating new end-user details or creating new corporate client details and including the details of the categories and the modules that are created newly by the super user becomes very difficult manually. Once all information is provided manually it become quit difficult for updating the records with new information manually, so we automate the system to overcome all such difficulties. E-care, which is developed manually, is the existing system. This system generates and delivers the services to the end-users, which depends on the response provided by the support team who actually message the responses to the corporate users for their requests.

By manually the Help Desk System will face difficulties while updations and it is time consuming and leads more errors. E-care automates the tedious job of maintaining the corporate user details or corporate client details and manages the updations and see that the responses are sent to the corporate user as soon as possible. It includes main members like:

1. Super user
2. Corporate User
3. Corporate client
4. Level1 Administrator

5.2 Implementation environment

5.2.1 About Java

The Internet helped catapult Java to the forefront of programming, and Java, in turn, has had a profound effect on the Internet. The reason for this is quite simple: Java expands the universe of objects that can move about freely in the cyberspace. In a network, two very broad categories of objects are transmitted between the server and our computer passive information and dynamic, active programs.

As desirable as dynamic, networked programs are, they also present serious problems in the areas of security and portability. Java achieves this security or protection by confining a Java program to the Java execution environment and not allowing it to access to other parts of the computer. Many types of the computers and operating systems are in use throughout the world and many are connected to the Internet. For programs to be dynamically download to all the various types of platforms connected to the Internet, some means of generating portable executable code is needed.

Java was designed to be easy for the professional programmers to learn and use efficiently. The object model of Java is simple and easy to extend, while simple types, such as integers are kept as high-performance non-objects.

E-care is implemented using the Java language so as to make the programmers (which includes super user, and Levels) and different corporate clients and corporate users to handle e-Care easily without any difficulty.

The multiplatformed environment of the web places extraordinarily demands on a program, because the program must execute reliably in a variety of systems. Thus, the ability to create robust programs was given a high priority in the design of Java. At the same time, Java frees us from having to worry about many of the most common cause of programming errors. Because Java is strictly typed language, it checks our code at the compile time and also at run-time. Java was designed to meet the real world requirements of creating interactive,

Networked programs. Java supports multithreaded programming, which allows us to write programs that do many things simultaneously. A central issue for the Java designers

was that of code longevity and portability. Their goal was write once; run anywhere, any time , forever .

Java enables the creation of the cross-platform programs by compiling into intermediate representation called Javabyte code. This code can be interpreted on any system that provides a java virtual machine. Java is designed for the distributed environment of the Internet, because it handles TCP/IP protocols. In fact, accessing a resource using a URL is not much different from accessing a file. The original version of Java (OAK) included features for intra-address-space messaging. Java programs carry with them substantial amounts of the run-time type information that is used to verify and resolve accesses to objects at run-time. This makes it possible to dynamically link code in a safe and expedient manner.

5.2.2 About JDBC

It is a java data base connectivity having data base as a back-end of java as front end. Database is a collectivity of related data. A data base management system (DBMS) controls the storage or retrieval of data in the database. The main use of this JDBC is the database connectivity.

Using drivers we can get connection to the back-end:

- which drive we are going to connect back-end.
- create a data source name (dsn).
- create a statement for connection.

The package we use here is `import java. Sql. *` Interfaces of jdbc include driver, connection, and statement; prepare statement, callable statement, Result Set, Result Set Meta data. Classes of JDBC are driver manages, driver property information, date, time, and timestamp, type. The driver defined by the class `Jdbc odbc driver` in package `sun.jdbc.odbc`. Class `sun.jdbc.odbc.jdbc odbc driver` represents the jdbc to odbc bridge driver.

5.2.3 About Servlets

Servlet is server side applet. It contains several advantages. They are:

- Performance is significantly better. Servlets execute within the address space of the web server. Creating a separate process to handle each client request isnt necessary.
- Servlets are platform-independent, because they are written in Java. Several web servers, from vendors such as sun, Netscape, and Microsoft, offer the servlet API Programs developed for this API can be moved to any of these environments without recompilation.

- The java security manager on the server enforces a set of restrictions to protect the resources on a server machine.
- The full functionality of the Java class libraries is available to a servlet. It can communicate with applets, databases, or other software via the sockets and RMI mechanisms.

The Lifecycle of a servlet: Three methods are central to the lifecycle of a servlet: `init()`, `service()`, and `destroy()`. First, assume that a user enters a Uniform Resource Locator (URL) to the web browser. The web browser then generates an HTTP request for this URL and sends it to the appropriate server. Second the web server receives this HTTP request. The server maps this request to particular servlet. The servlet is dynamically retrieved and loaded into the address space of the server. Third, the server invokes the `init()` method of the servlet. This method is invoked only when the servlet is first loaded into the memory. Fourth, the server invokes the servlet `service()` method, which is called to process the HTTP request. The `service()` method is called for each HTTP request. Two packages are required to build the servlet they are `javax.servlet` and `javax.srvlet.http`. They constitute of servlet API. `ServletRequest` interface is used to read the data from a client request and `ServletResponse` is used to write data to the client response. The `javax.srvlet.http` package include interfaces like `HttpServletRequest` which enables servlets to read data from HTTP request and `HttpServletResponse` which enables servlets to write data from HTTP response.

5.3 Summary

In this chapter main focus is on the various technologies used in E-care help desk system.

Chapter 6

System Testing

Software Testing is a critical element of software quality assurance and represents the ultimate review of specification, design and coding, Testing presents an interesting anomaly for the software engineer.

Testing Objectives include:

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

In section 6.1 various testing implementation schemes are introduced and in section 6.2 test cases and their results are given.

6.1 How to implement testing

A Strategy for software testing integrates software test cases into a series of well planned steps that result in the successful construction of software. Software testing is a broader topic for what is referred to as Verification and Validation. Verification refers to the set of activities that ensure that the software correctly implements a specific function. Validation refers he set of activities that ensure that the software that has been built is traceable to customers requirements

Unit Testing:

Unit testing focuses verification effort on the smallest unit of software design that is the module. Using procedural design description as a guide, important control paths are tested to uncover errors within the boundaries of the module. The unit test is normally white box

testing oriented and the step can be conducted in parallel for multiple modules.

Integration Testing:

Integration testing is a systematic technique for constructing the program structure, while conducting test to uncover errors associated with the interface. The objective is to take unit tested methods and build a program structure that has been dictated by design.

Top-down Integration:

Top down integrations is an incremental approach for construction of program structure. Modules are integrated by moving downward through the control hierarchy, beginning with the main control program. Modules subordinate to the main program are incorporated in the structure either in the breath-first or depth-first manner.

Bottom-up Integration:

This method as the name suggests, begins construction and testing with atomic modules i.e., modules at the lowest level. Because the modules are integrated in the bottom up manner the processing required for the modules subordinate to a given level is always available and the need for stubs is eliminated.

Validation Testing:

At the end of integration testing software is completely assembled as a package. Validation testing is the next stage, which can be defined as successful when the software functions in the manner reasonably expected by the customer. Reasonable expectations are those defined in the software requirements specifications. Information contained in those sections form a basis for validation testing approach.

System Testing:

System testing is actually a series of different tests whose primary purpose is to fully exercise the computer-based system. Although each test has a different purpose, all work to verify that all system elements have been properly integrated to perform allocated functions.

Security Testing:

Attempts to verify the protection mechanisms built into the system.

Performance Testing:

This method is designed to test runtime performance of software within the context of an integrated system.

6.2 Test cases and test results

Sr.no.	Test condition	Expected Result	Actual Output	Test result
1	Client Registration	Client Registered	Client Registered successfully	pass
2	Creating User	User Created	User created successfully	pass
3	Posting Problems	problem posted	problem posted with ticket number	pass
4	message sending	message sent	message has been sent	pass
5	problem solution	problem is finished	problem solution has been sent	pass

6.3 summary

In this chapter various testing schemes and based upon that various test cases and their results are covered.

Chapter 7

Result and Analysis

7.1 Analysis



Figure 7.1: Corporate User And Corporate Client Use Case Diagram



Figure 7.2: Corporate User And Corporate Client Use Case Diagram



Figure 7.3: Corporate User And Corporate Client Use Case Diagram



Figure 7.4: Corporate User And Corporate Client Use Case Diagram

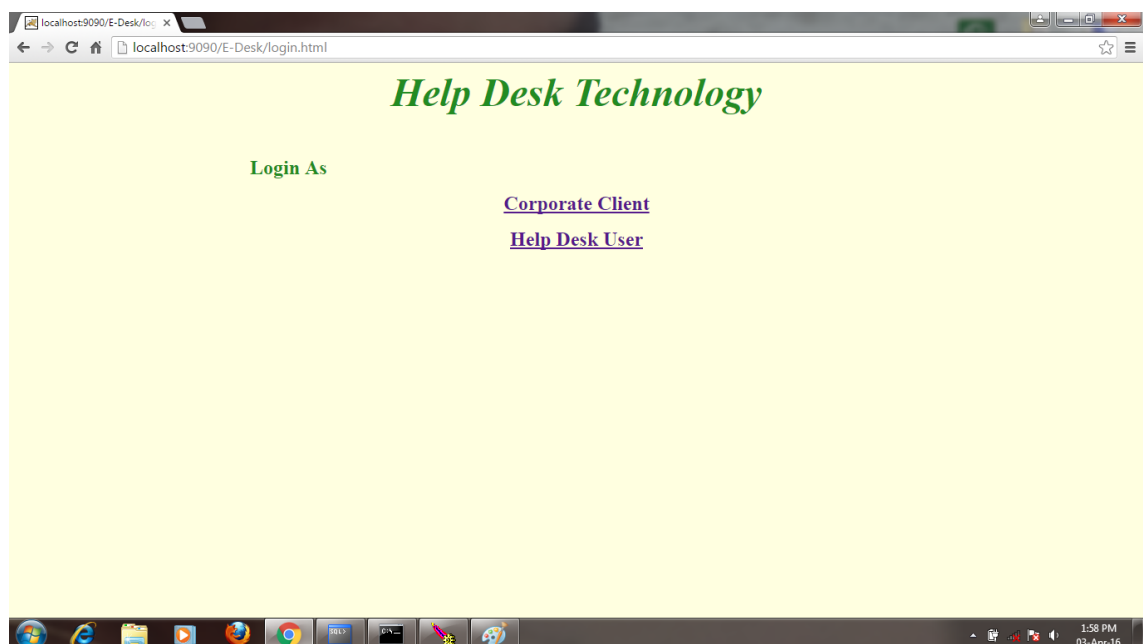


Figure 7.5: Corporate User And Corporate Client Use Case Diagram

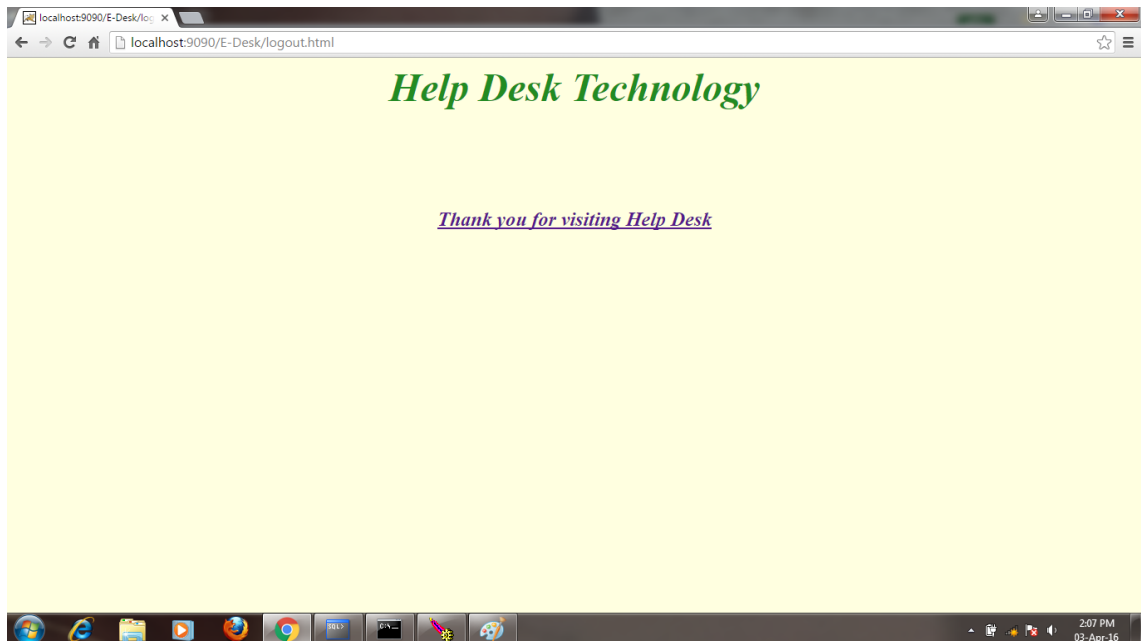


Figure 7.6: Corporate User And Corporate Client Use Case Diagram

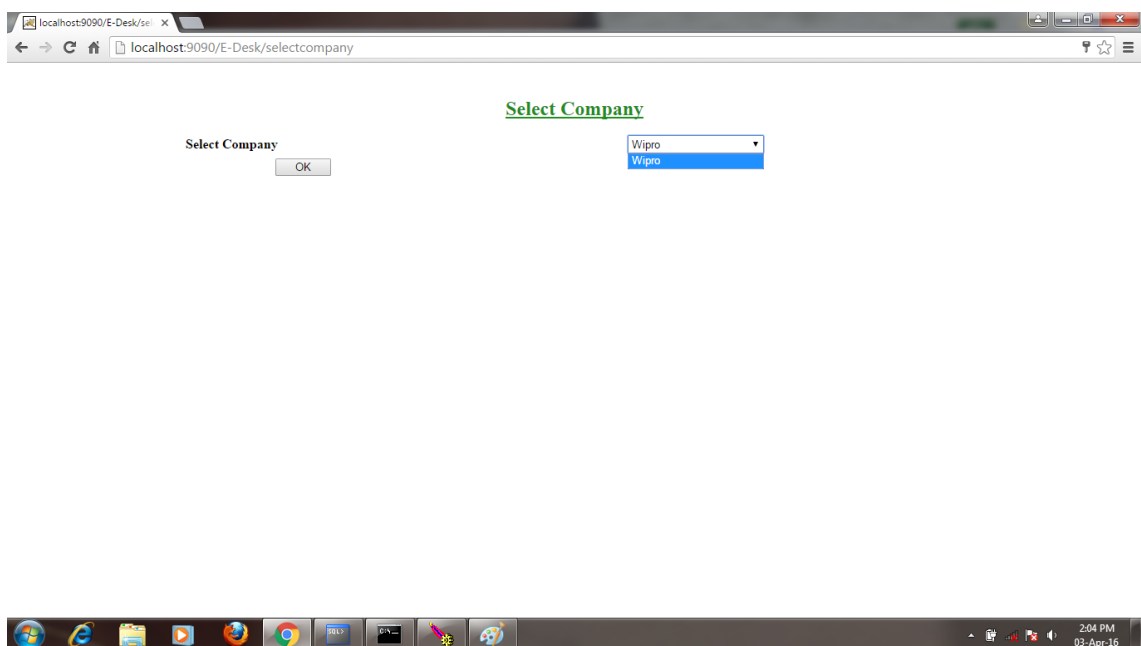


Figure 7.7: Corporate User And Corporate Client Use Case Diagram

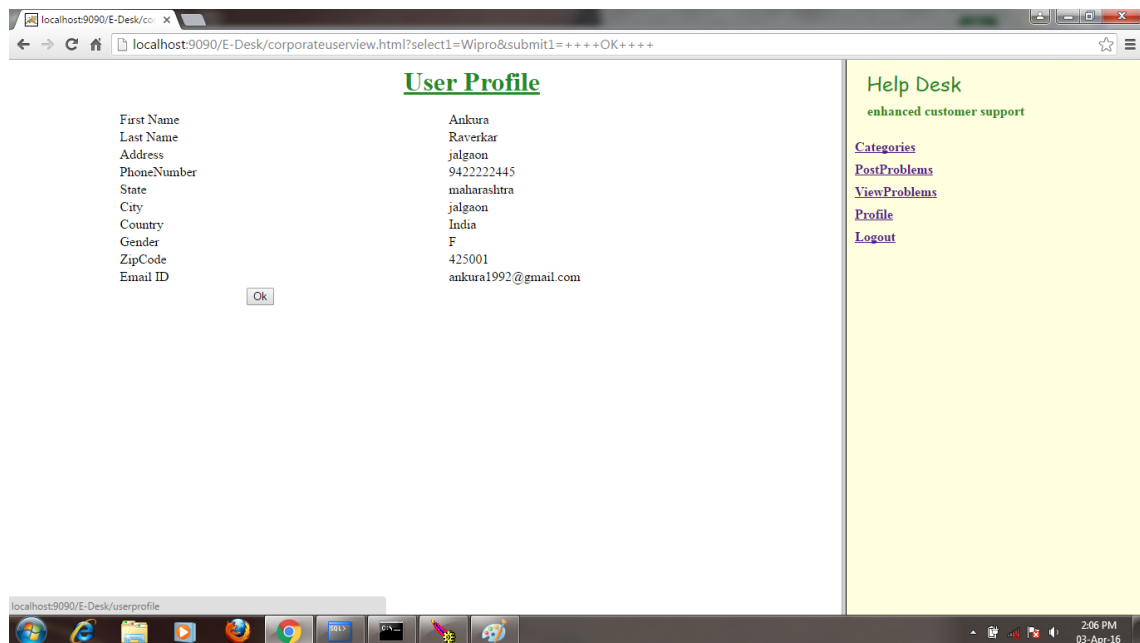


Figure 7.8: Corporate User And Corporate Client Use Case Diagram

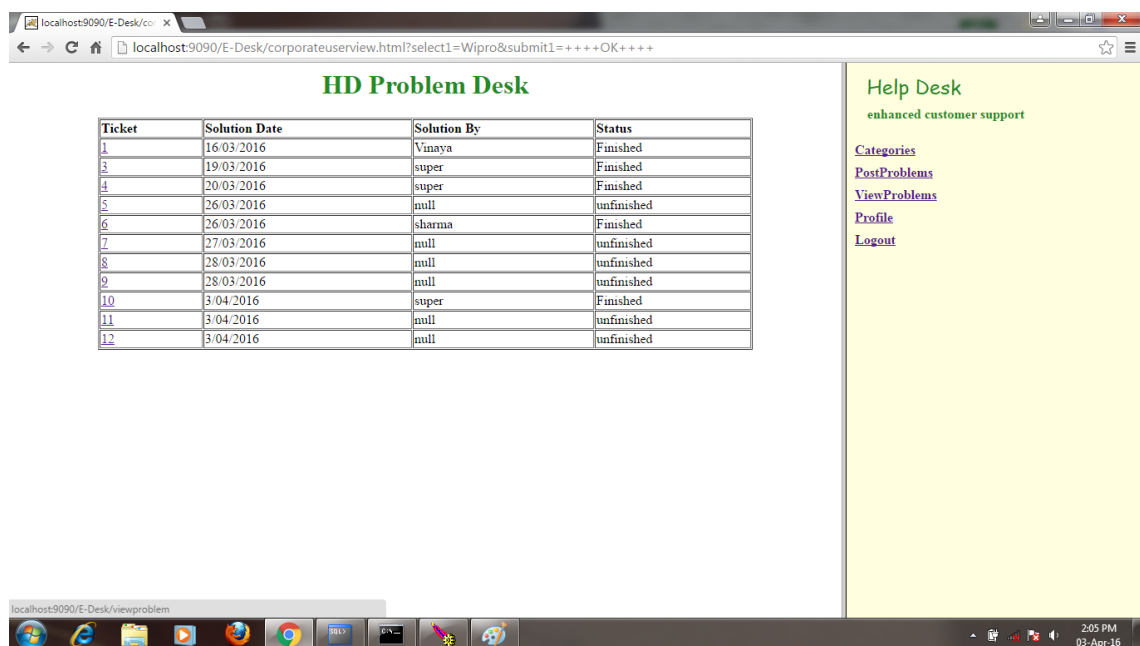


Figure 7.9: Corporate User And Corporate Client Use Case Diagram

Chapter 8

Conclusion and future scope

8.1 Conclusion

The result of the E-care is that it supports many corporate clients and the individual users or the end-users with unique interface, through which the HD problems are solved within given time and it shows efficiency in the internal messaging system and results in accurate task scheduling system and this E-care can be run with the minimal administration.

8.2 Future scope

Bibliography

- [1] Hempel, Jessi “Restart: Microsoft in the age of Satya Nadella”. Wired. Retrieved 22 January 2015.
- [2] Hempel, Jessi (21 January 2015). “Restart: Microsoft in the age of Satya Nadella ”. Wired. Retrieved 22 January 2015.
- [3] Davies, Chris “HoloLens hands-on: Building for Windows Holographic”. SlashGear. Retrieved 1 May 2015.
- [4] Robertson, Adi “Microsoft’s HoloLens is new, improved, and still has big problems ”. The Verge. Retrieved 2 May 2015.
- [5] Metz, Rachel “Why NASA Wants Microsofts HoloLens in Space ”. MIT Technology Review. Retrieved 12 September 2015.