The Great Mind Challenge'12

Nurturing Great Minds

HIGHLY CONFIDENTIAL SECURITY SYSTEM

Software Requirement Specification

TEAM NAME

TheFallen

TEAM MEMBERS

Dharmendra Hingu

Meenali Kaprani

Harnisha Gevaria

PROJECT GUIDE

Prof. Sandeep Udmale

Asst. Professor

Department Of Computer and Information Technology

VEERMATA JIJABAI TECHNOLOGICAL INSTITUTE, MUMBAI 400019

TABLE OF CONTENTS

1. INTRODUCTION

1.1 METHODOLOGY

The Rational Unified Process (RUP) brings together elements from all of the generic process models, supports iteration and illustrates good practice in specification and design. The RUP is normally described from three perspectives:

A ***dynamic perspective*** that shows the phases of the model over time

A ***static perspective*** that shows the process activities that are enacted

A ***practice perspective*** that suggests good practices to be used during the process

The different phases in RUP are

Inception

The goal of the inception phase is to establish a business case for the system. Identifying all external entities that will interact with the system and defining these interactions. This information is used to assess the contribution of system to business.

Elaboration

The goals of the elaboration phase are to develop an understanding of the problem domain, establish an architectural framework, develop project plan and identify key project risks.

Construction

This phase is concerned with system design, programming and testing. Parts of the system are developed in parallel and integrated during this phase.

Transition

This is the final phase of RUP and is concerned with moving the system from the development community to the user community and making it work in real environment.

1.2 PURPOSE

The purpose of the project is to create a Highly Confidential Security System as a web application. The confidential data of the user can be uploaded on the application and can also share the same. Due to busy life style we can't remember all confidential data like all mail id, password, all bank account, insurance policy no., PAN no., Driving license no., Passport no., ATM PIN no., all education certificate numbers, some highly value scan copy, some confidential photo and music, video.

We can develop highly secured web application, so that we can store all confidential data in single credentials.

1.3 SCOPE

This web application will incorporate all requirements needed for protecting, all the required confidential information in organized fashion. This will basically help the people to protect their information from getting hacked and can be accessed anywhere.

This integrates other services like online access of this application and the following:

- Security System (Software and hardware): Protection to the uploaded files
- Email id and password locker: Separate locker for storing email id's and passwords
- Bank account information locker: Separate locker for storing bank account information
- Videos locker: Separate locker for storing videos and animations
- Images locker: Separate locker for storing images and snapshots
- Music locker: Separate locker for storing audio files

1.4 DEFINITIONS, ACRONYMS AND ABBREVIATIONS

Highly confidential Security System (HCSS)

It's a web application that provides online security system where all the user data if safe from other world.

Administrator (admin)

Administrator has the authority to add/delete users, grant permission to moderator and Users to share a particular file and its grant hierarchy.

Moderator (mod)

Moderator is the one who handles all the files in the database and see to it that the system is working properly. If there is any complaints/reviews from the user, mod is supposed to resolve it and if needed involve the admin.

JSP

Java Server Pages is used to create dynamic web content.

DB2 (IBM Database 2)

DB2 is a database management system that provides a flexible and efficient database platform to raise a strong "on demand" business applications.

WASCE

Web Sphere Application Server Community Edition is an application server that runs, supports J2EE and web service applications.

XML

Extensible Mark-up Language is a text based format that let developers describe, deliver and exchange structured data between a range of applications to client for display and manipulation.

UML

Unified Modelling Language is a standard language for writing software blueprints. The UML may be used to visualize, specify, construct and document

AJAX

Asynchronous JavaScript and XML is a technique used in java script to create dynamic web pages.

1.5 TOOLS USED

Application architecture – JAVA, J2EE

JAVA

Java is an object-oriented programming language developed by Sun Microsystems a company best known for its high end UNIX workstations. Java language was designed to be small, simple, and portable across platforms, operating systems, both at the source and at the binary level, which means that Java programs (applet and application) can run on any machine that has the Java virtual machine (JVM) installed.

J2EE

Java Platform, Enterprise Edition or **Java EE** is a widely used platform for server programming in the Java programming language. The Java platform (Enterprise Edition) differs from the Java Standard Edition Platform (Java SE) in that it adds libraries which provide functionality to deploy fault-tolerant, distributed, multi-tier Java software, based largely on modular components running on an application server.

Web server – WASCE

Web Sphere Application Server Community Edition (from now on WASCE) is a free, certified Java EE 5 server for building and managing Java applications. It is IBM's supported distribution of Apache Geronimo that uses Tomcat for served container and Axis 2 for web services. Over 15WASCE developers are committers in the Apache Geronimo project.

Database platform – DB2

DB2 Database is the database management system that delivers a flexible and cost effective database platform to build robust on demand business applications and supports the J2EE and web services standards.

Design tool – Rational Software Modeler

IBM Rational Software Modeler, (RSM) made by IBM's Rational Software division, is a Unified Modeling Language UML 2.0-based visual modeling and design tool. Rational Software Modelers built on the Eclipse open-source software framework and includes capabilities focused on visual modeling and model-driven development (MDD) with the UML for creating resilient, thought-out applications and web services.

1.6 TECHNOLOGIES TO BE USED

- ✓ JSP, AJAX and HTML
- ✓ DB2: Relational Database Management System
- ✓ WASCE: Web sphere Application Server Community Edition
- ✓ Rational Rose/Rational Software Architecture

1.7 OVERVIEW

- ✓ Existing Systems:
  - ✓ Registration for the users
  - ✓ Able to upload files

- ✓ Drawbacks:
  - ✓ Less secure
  - ✓ No provision for sharing some of the information to authenticated users

- ✓ Proposed System:
  - ✓ Registration for the users
  - ✓ Moderators to help the user as in provides the sharing capabilities
  - ✓ Provision for uploading and downloading the user's as well as shared files
  - ✓ More secured

- ✓ Our Plan:
  - ✓ Registration for user
  - ✓ Online maintenance of the user's confidential data Support to share some data to authenticated users (friends, officemates etc.)
  - ✓ User review for the system enhancement

- ✓ This SRS will include two sections which are as follows,
  - ✓ Overall Description will describe the major components of the system, interconnection and external interfaces.
  - ✓ Specific Requirement will describe the functions of actors, their roles in the system and constraints.

2. OVERALL DESCRIPTION

2.1. Product Perspective

- ✓ The web pages provide the UI on the client side. Communication between client and server is through HTTP.
- ✓ The client side software consists of the web browser and the operating system itself.
- ✓ The server side consists of the web server (for the web application) and the database server (for storing data for the web application).

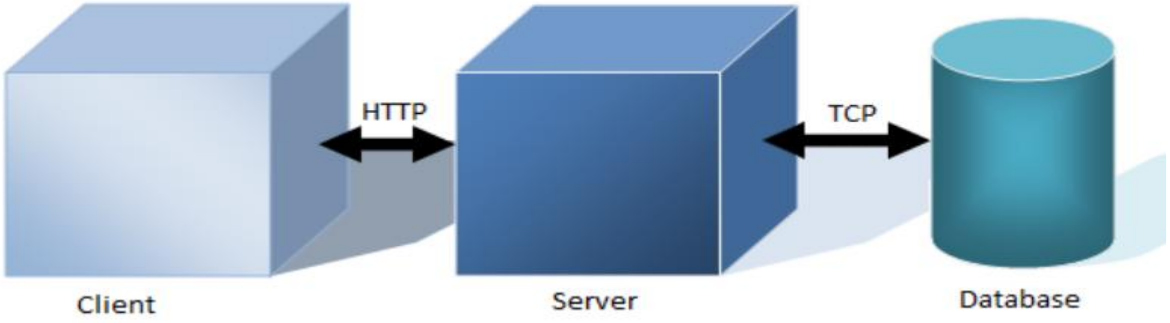

Fig 2.1 Model

- ✓ The database may reside on the same server on which the web server resides, or on a different server.

2.2. Software Interface

- ✓ *Client on Internet* - Web browser, any operating system
- ✓ *Client on Intranet* - Web browser, any operating system
- ✓ *Web Server* - WASCE, any operating system
- ✓ *Database Server – DB2*, any operating system
- ✓ *Development End* –RAD(Java, JSP, HTML, XML, Servlets), Web Sphere

2.3. Hardware Interfaces

Client Side	Processor	RAM	Free disk space
Web browser (any)	Pentium II at 500 MHz	256MB	100MB
Server Side	Processor	RAM	Free disk space
Web Sphere Application Server(Com. Edition)	Pentium III at 1GHz	256MB	120MB
DB2 Express C V9.7	Pentium III at 1GHz	512MB	512MB

Table 2.1 Hardware Requirements

2.4. COMMUNICATION INTERFACES

- ✓ Client (System user, Customer) on internet will be using HTTP/HTTPS protocol

2.5. CONSTRAINTS

- ✓ GUI is only in English
- ✓ Login and password is used for the identification of users
- ✓ Only registered users and moderators will be authorized to use the services
- ✓ Limited to HTTP/HTTPS
- ✓ The user should have basic knowledge of computer
- ✓ This system is working for single server

2.6 E-R DIAGRAM

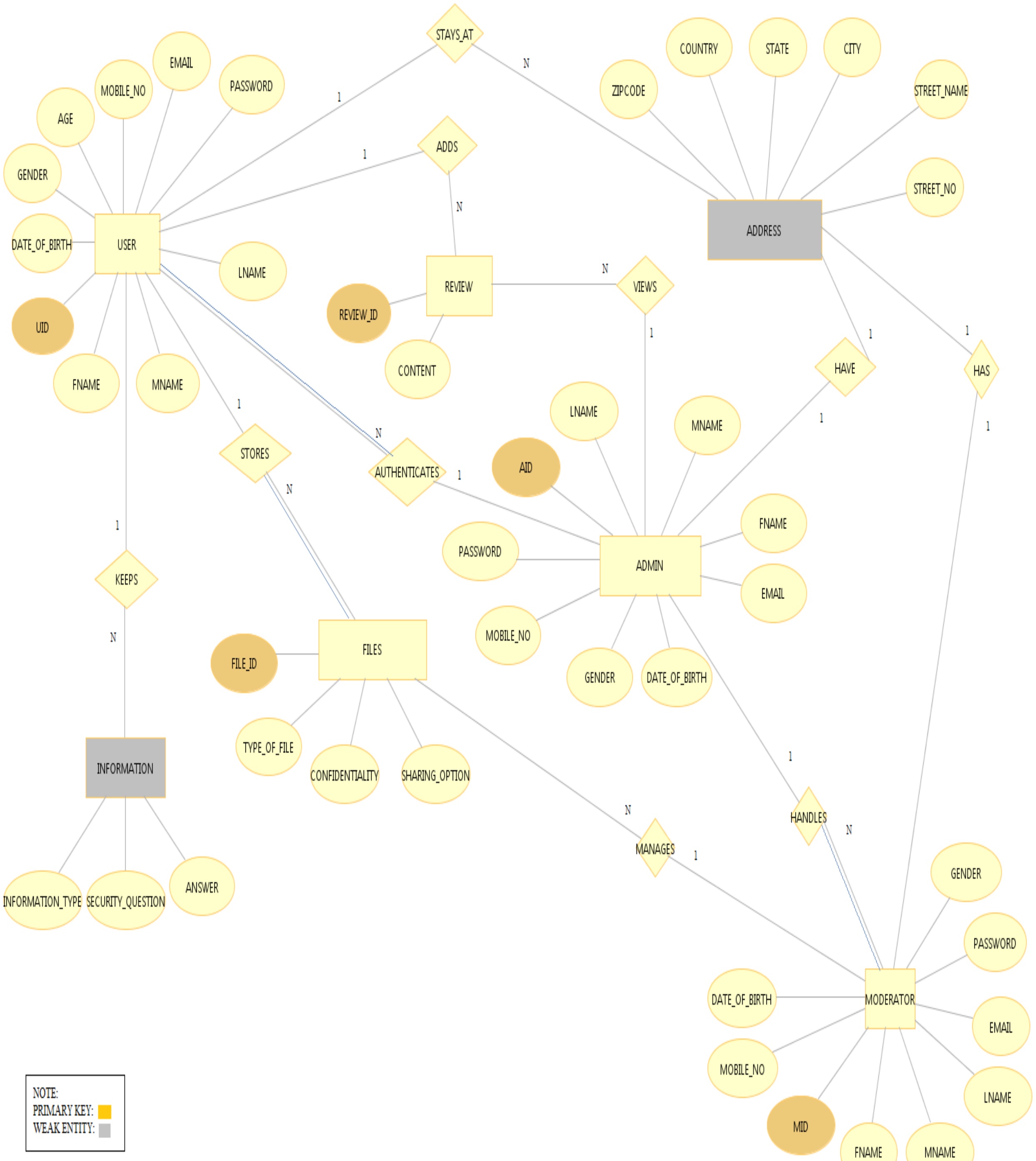

Fig. 2.2 Entity Relationship Diagram

2.7 USE CASE MODEL SURVEY

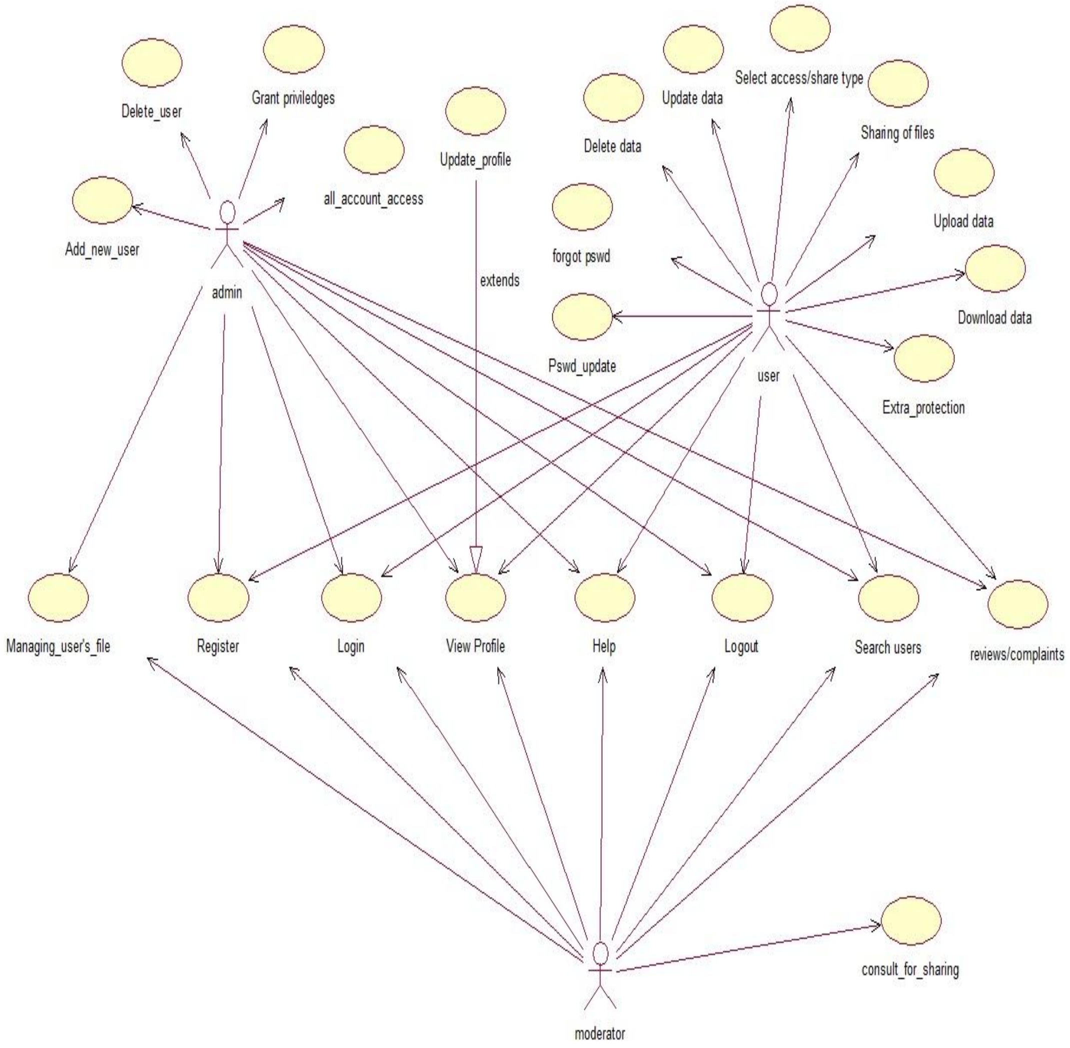

Fig 2.3 Use Case Model

User:

Users can view their files they have stored, other users on the system and can also share file with a particular user if he wants. User can also register complaint on any drawback of the system.

Moderator:

Moderator views and manages the files, he is only allowed to see the file name the user has stored .Moderator does not have access to the content of the users file. He also sends reports to admin.

Admin:

Admin has the authority to add/delete users, grant permission to users and moderator, to generate and view files store them safely. He responds the complaints of user and takes necessary actions.

2.8 ARCHITECTURE DESIGN

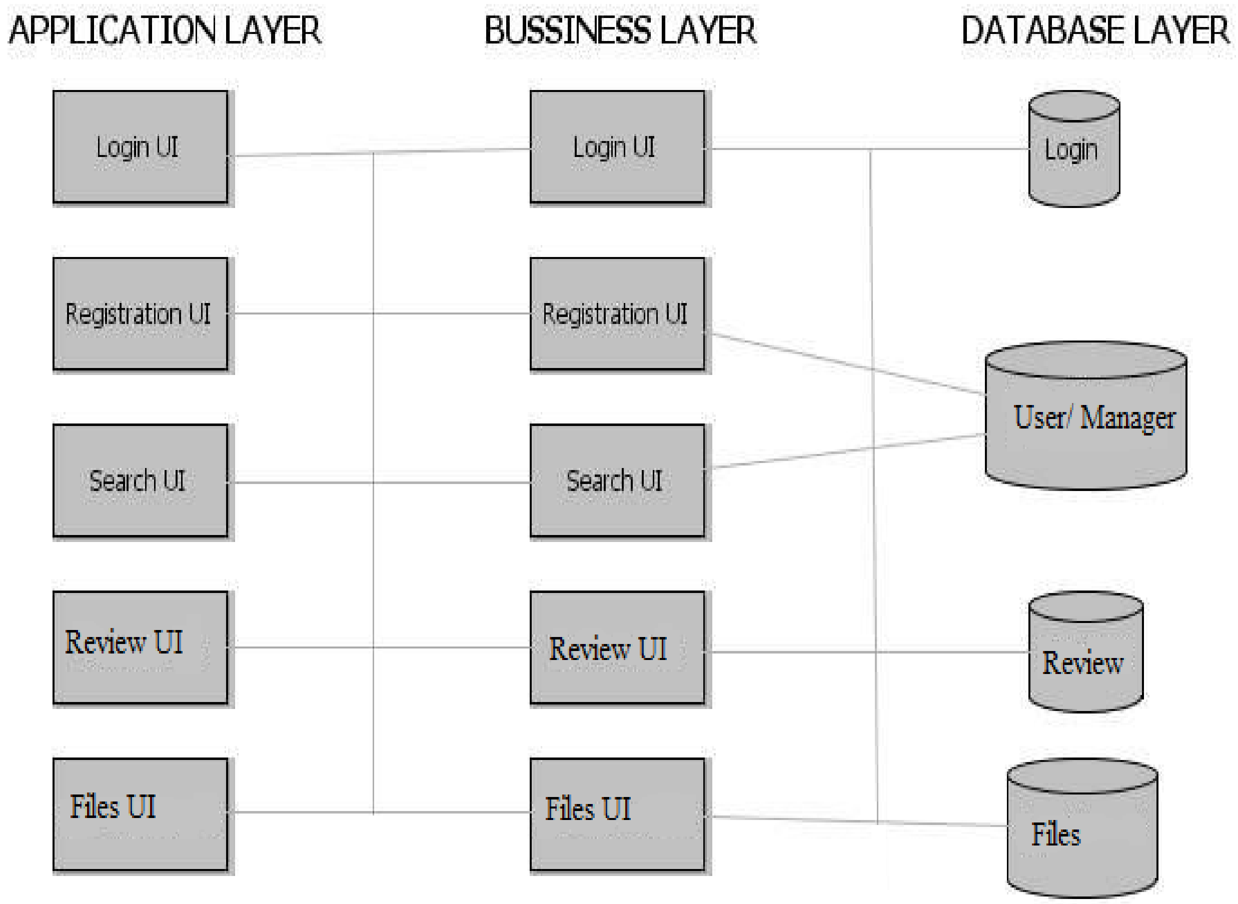

Fig 2.4 Architecture diagram

2.9 DATABASE SCHEMA DIAGRAM

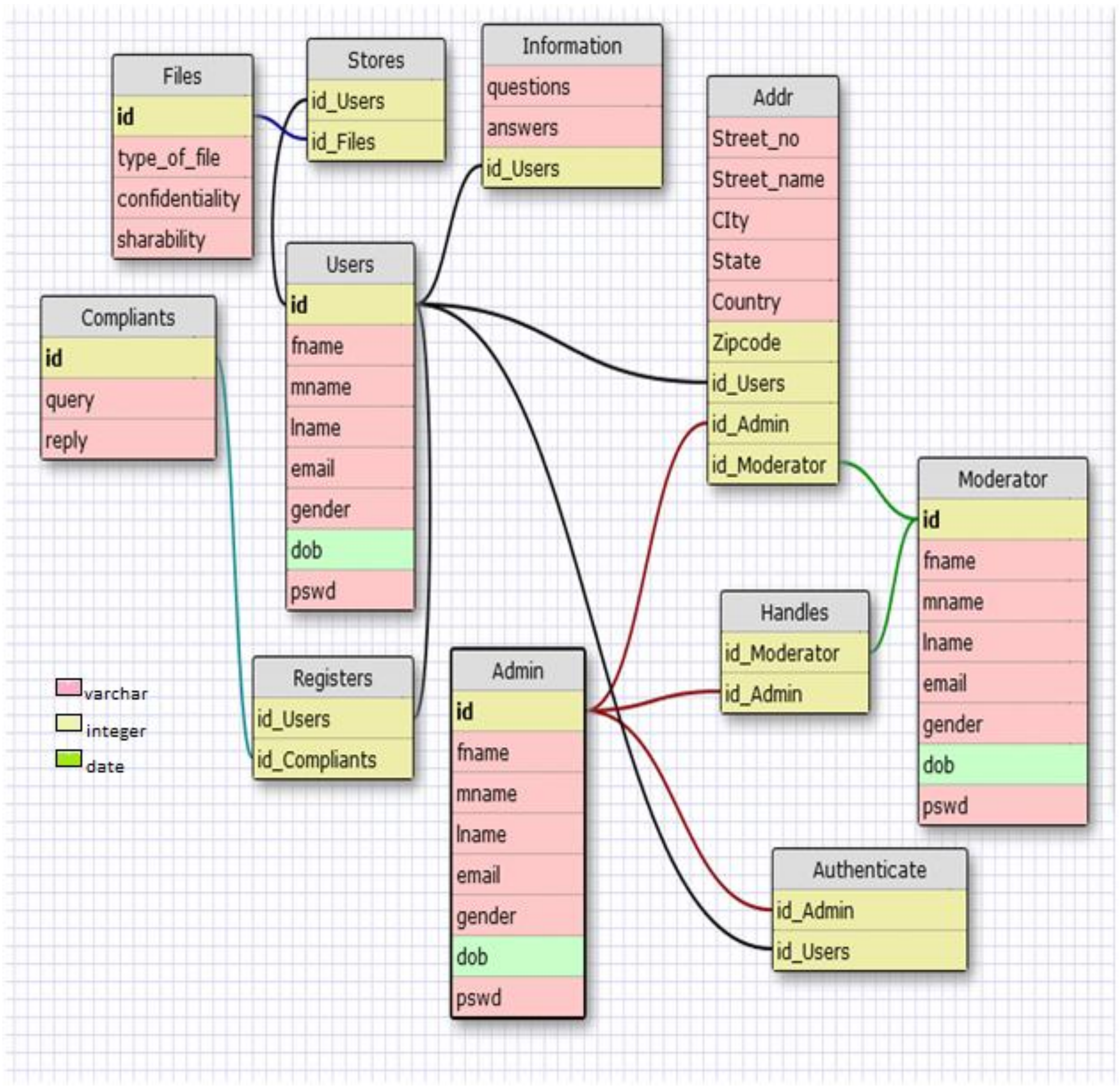

Fig 2.5 Database schema design

2.10 CLASS DIAGRAM

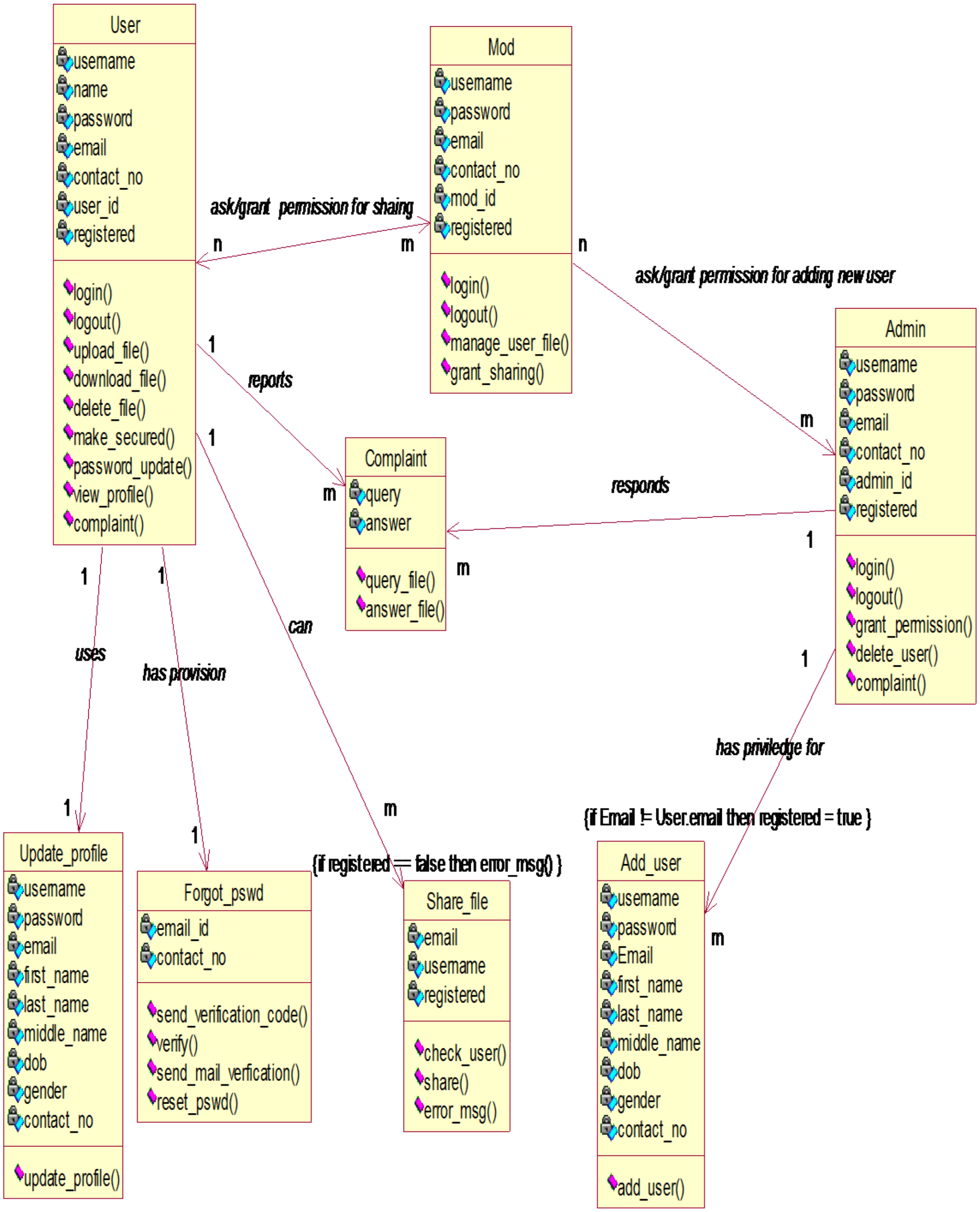

Fig 2.6 Class Diagram

3. SPECIFIC REQUIREMENT

3.1 USE CASE REPORTS

3.1.1 Administrator use-case report

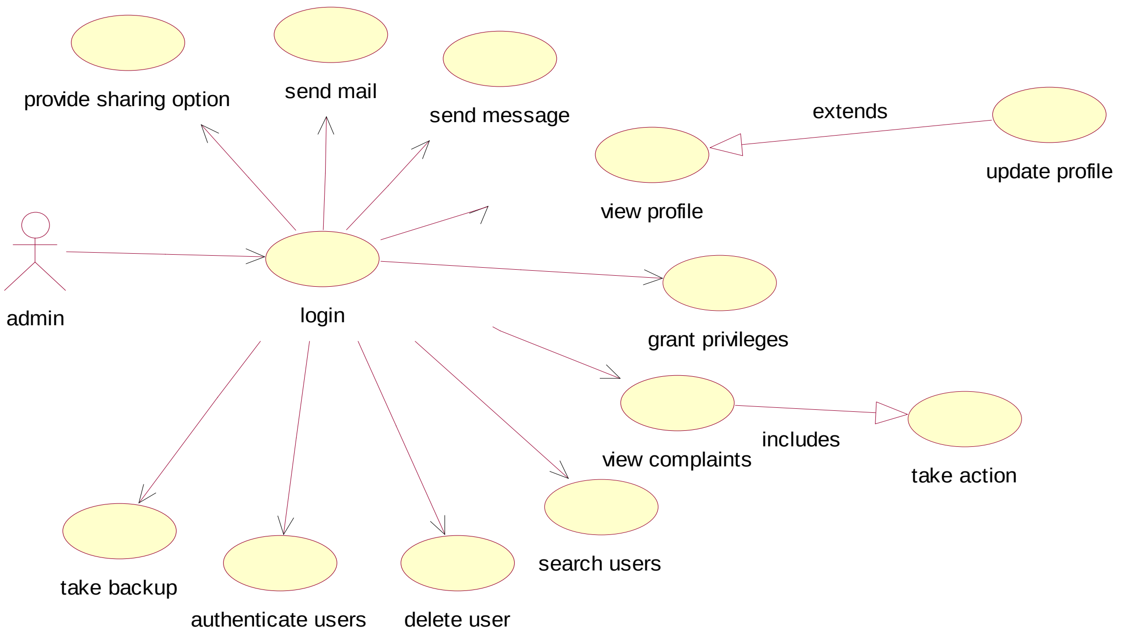

Fig 3.1 Admin use case

USE CASE	DESCRIPTION
Login	The admin has to login in order to do the following activities
provide sharing option	The admin provides options doe sharing the file of the user like read-only, write-only, share to others
send mail	Sends mail for user query request
send message	Can send message for user query request
view profile	The admin can view the profile of all the users
grant privileges	The admin can grant privileges to the user
view complaints	The admin views the review/complaints given by the users and takes action accordingly
search user	The admin can search the user in the system
delete user	The admin can delete a user from the system
authenticate user	The admin has the right to authenticate the new user
take backup	The admin backups all the files of different users

Table 3.1 Admin use cases

3.1.2 Moderator use-case report

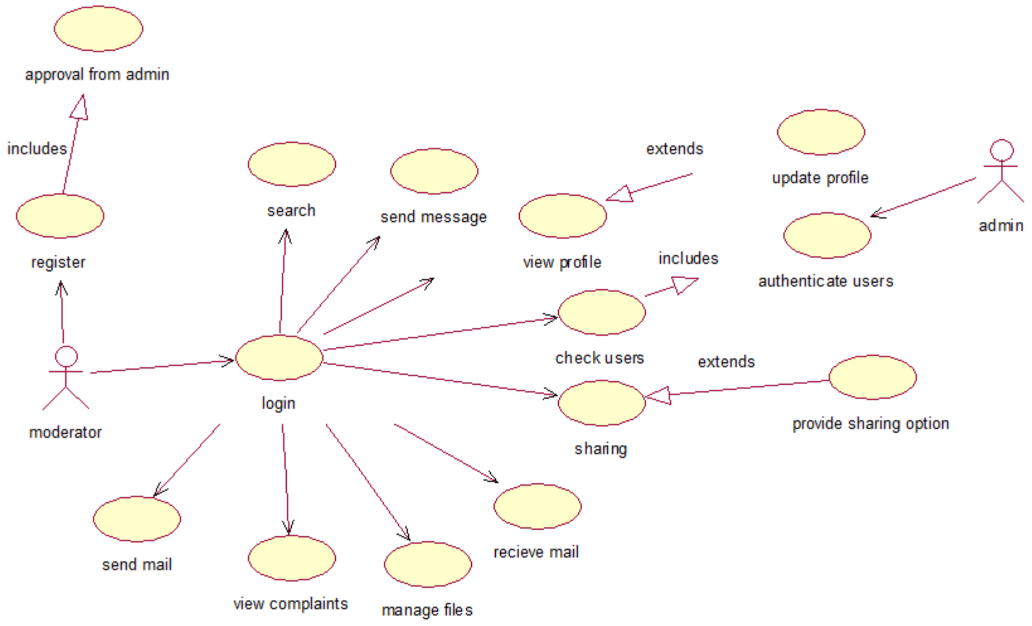

Fig 3.2 Mod use case

USE CASE	DESCRIPTION
login	The moderator has to login in order to do the following activities
search	The moderator can search users in the system
view profile	The moderator can view the profile of all the users in the system
check users	The moderator can check the users in order to authenticate new user
sharing	The moderator sends sharing requests of files from user to admin
receive mail	
manage files	The moderator manages all the files of the users
view complaints	The manger views the complaints or review from the user
send mail	Sends mail for the user query

Table 3.2 Mod use cases

3.1.3 User use-case report

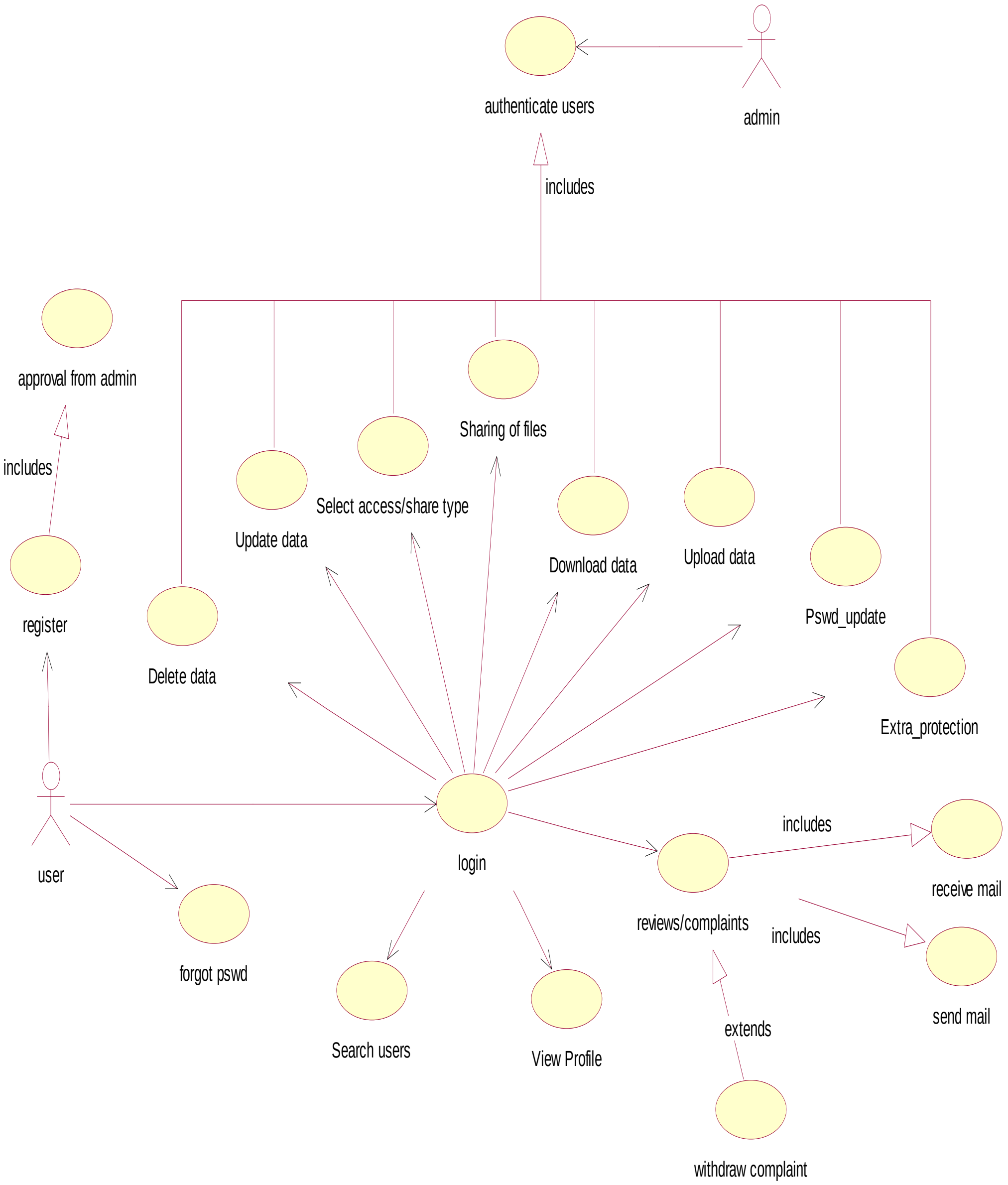

Fig 3.3 User use case

USE CASE	DESCRIPTION
register	User can register for the system
login	User has to login in order to do the following activities
search	The user can search the other users using the system
view profile	The user can see profiles of other users using the system
withdraw complaints	The user
select access/share type	User can select sharing type like read only, write , update etc.
receive mail	User receives reply to their request query
pswd update	User can update his/her password after authentication
view complaints	The manger views the complaints or review from the user
send mail	Can send complaints or query via mail
delete data	User can delete data after authentication from server
update data	User can update new versions of data/file after authentication from server
download data	User can download his/her own files/data or shared files/data after authentication
upload data	User can upload his/her own files/data or shared files/data after authentication
forgot pswd	User can retrieve his/her password after following procedure described
extra protection	User can provide an extra protection to his/her files/data after authentication

Table 3.3 User use cases

3.2 ACTIVITY DIAGRAM

3.2.1 User registration activity

The registration activity starts when the user enters all the details and then the username is checked if it is available. If the username is valid, then the user is asked for password and the password is verified. Then all the required fields are checked, if they are filled. If all the details are filled, the registration of the user is completed and the user homepage is displayed. If all the details are not filled, then the user is requested to fill all the necessary fields.

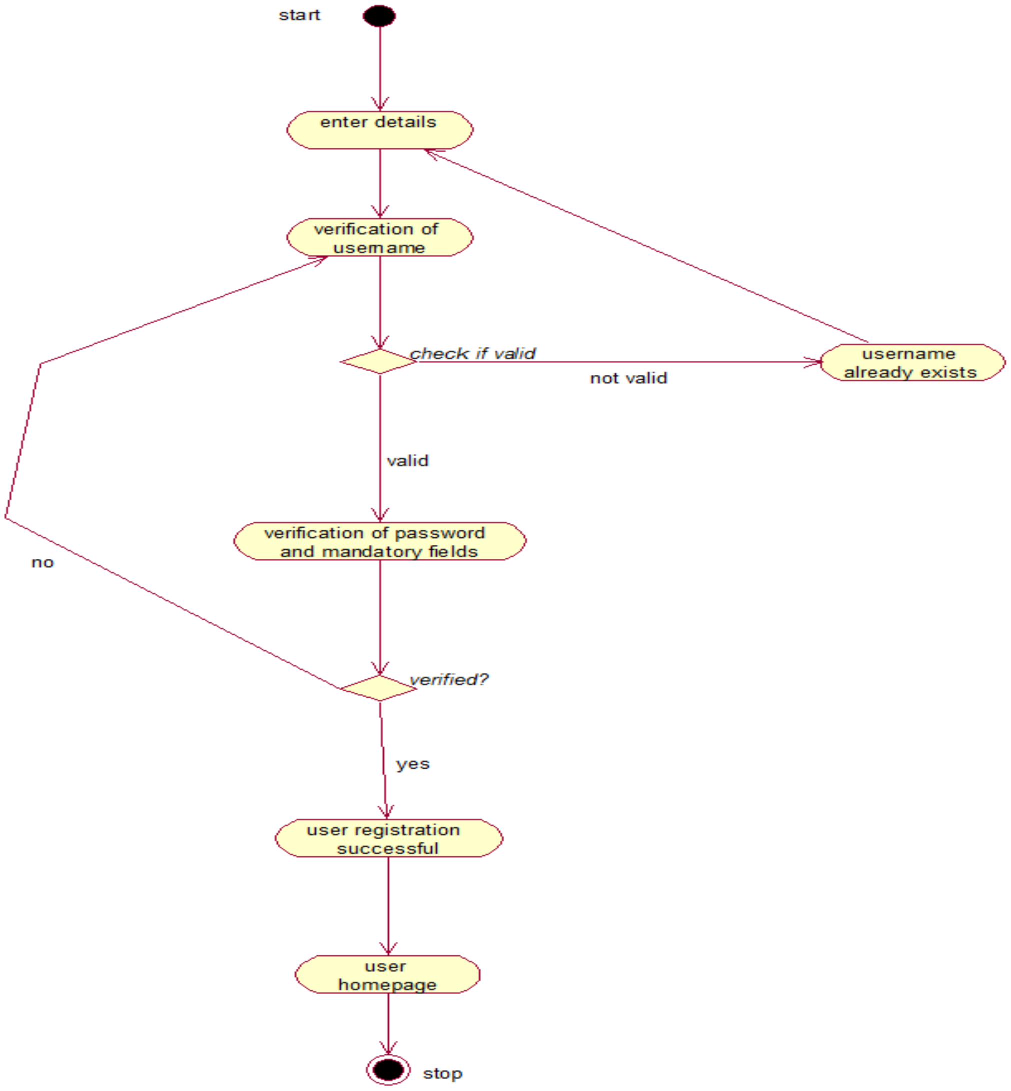

Fig 3.4 User Registration

3.2.2 User login activity

The user logs in by providing his/her username and password. If both are verified to be correct, then the user homepage gets displayed. If the username and/or the password are incorrect the system prompts an error and the user is requested to enter the correct username and/or password

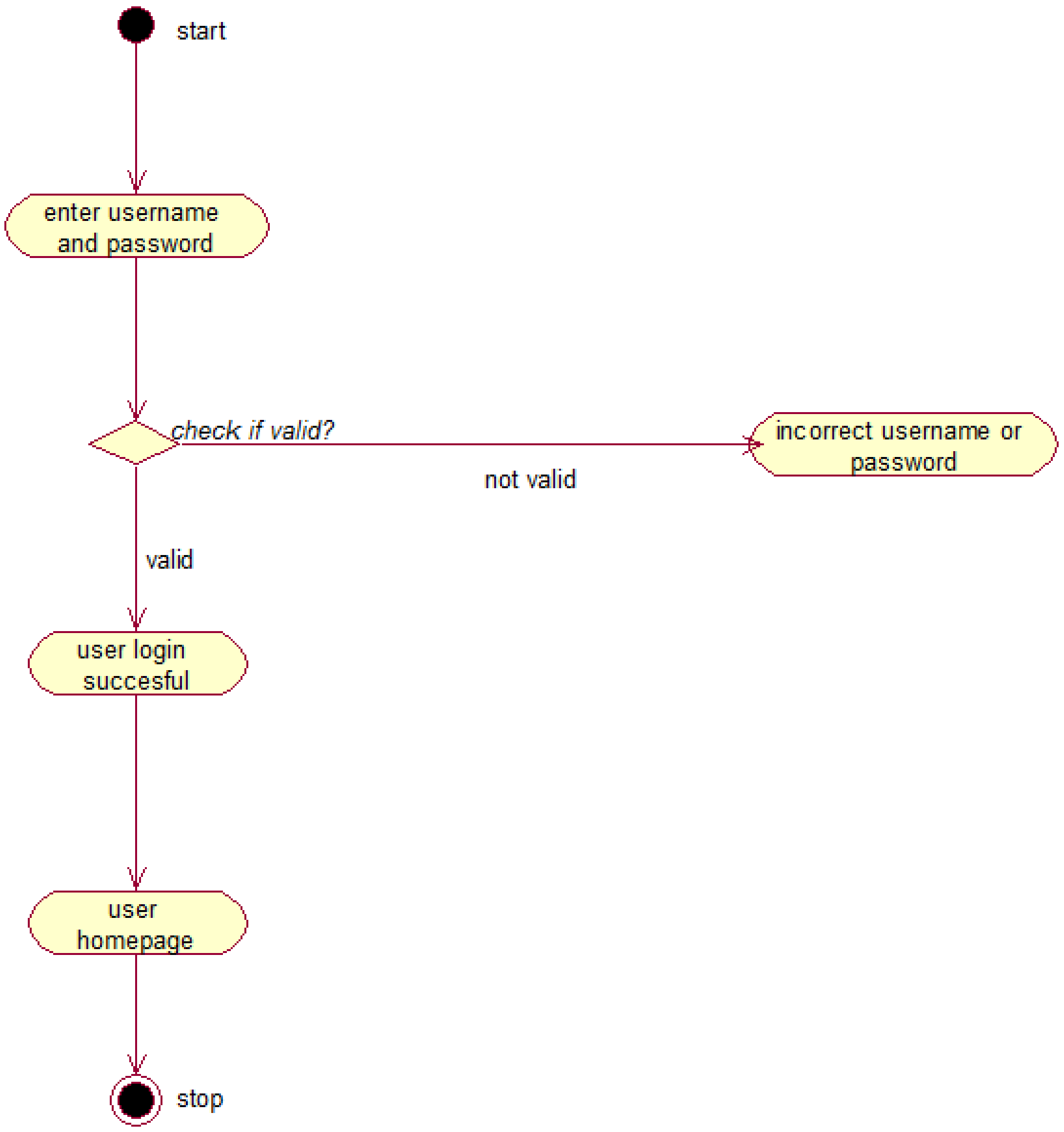

Fig 3.5 User Login

3.2.3 Add review activity

The user is allowed to add any review regarding the system, any suggestions or complaints. The user enters the text and then the review is sent to the admin, where the necessary actions are performed by the admin and a review id is provided to the user.

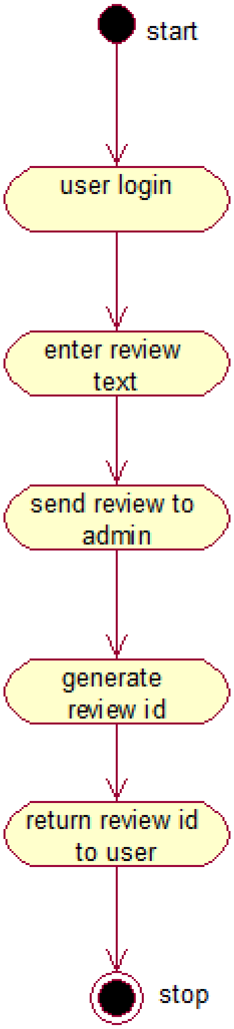

Fig 3.6 Add review

3.2.4 Add file activity

The user when logs in and wants to upload or update a file, then the system asks for sharing option. If the user wants to share files with other users, then the email id of those users are to be entered, and after verification the user is asked if the file needs extra protection. If yes, then the use is asked to enter mobile no and a code is sent on his mobile and after validating the code, the file is provided extra protection. Finally the file is uploaded.

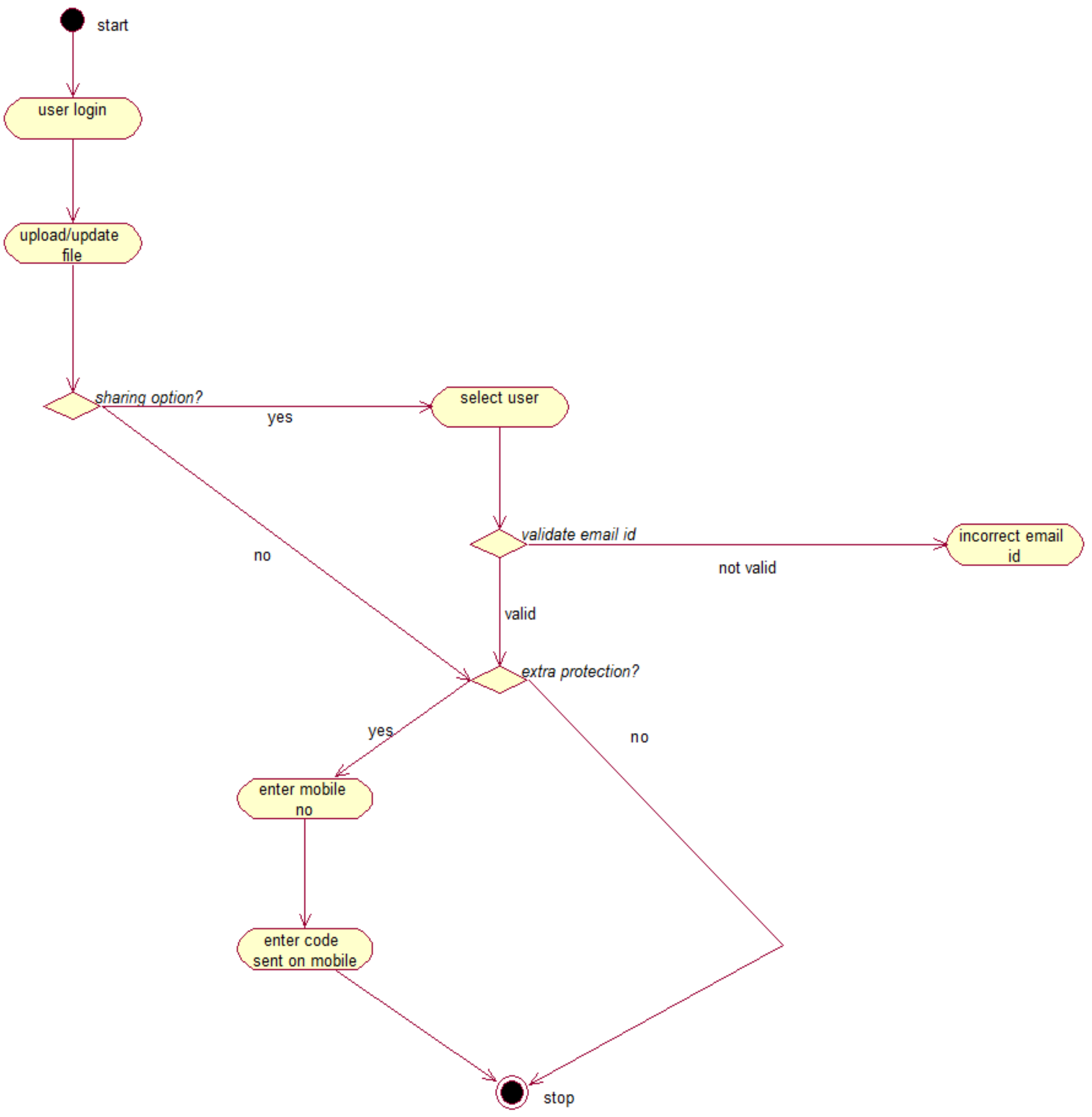

Fig 3.7 Add file

3.2.5 Forgot password activity

If the user forgets his username or password, then the user enters his mobile no. After which a code is sent on his mobile phone. The user enters the code and is validated. If the code is correct, then the password reset link is sent to his mail account. After clicking on the link, the user answers security question which is asked to the user at the time of registration. After verifying all the answer, the system asks the user to enter his new password and the user is logged out of the system.

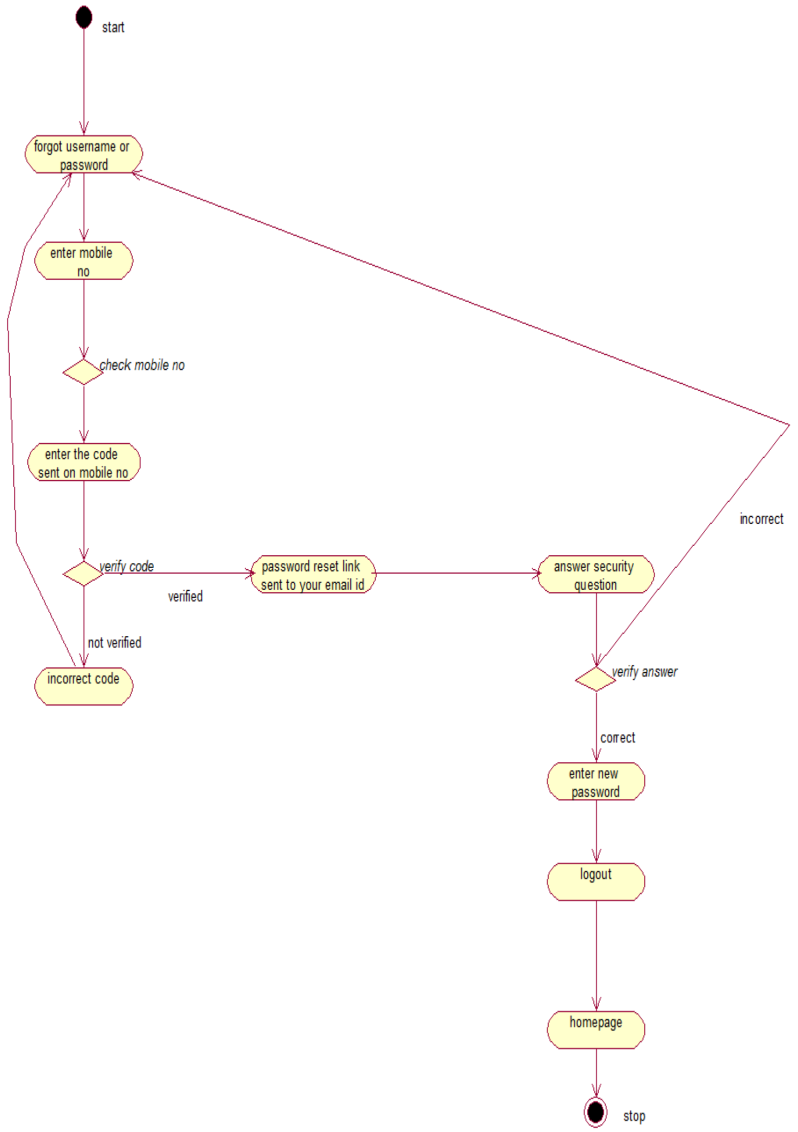

Fig 3.8 Forgot password

3.2.6 Update password activity

If the user wants to update his password, then after logging in, the user enters his previous password and answers the security question, which is verified. After this the user enters his new password.

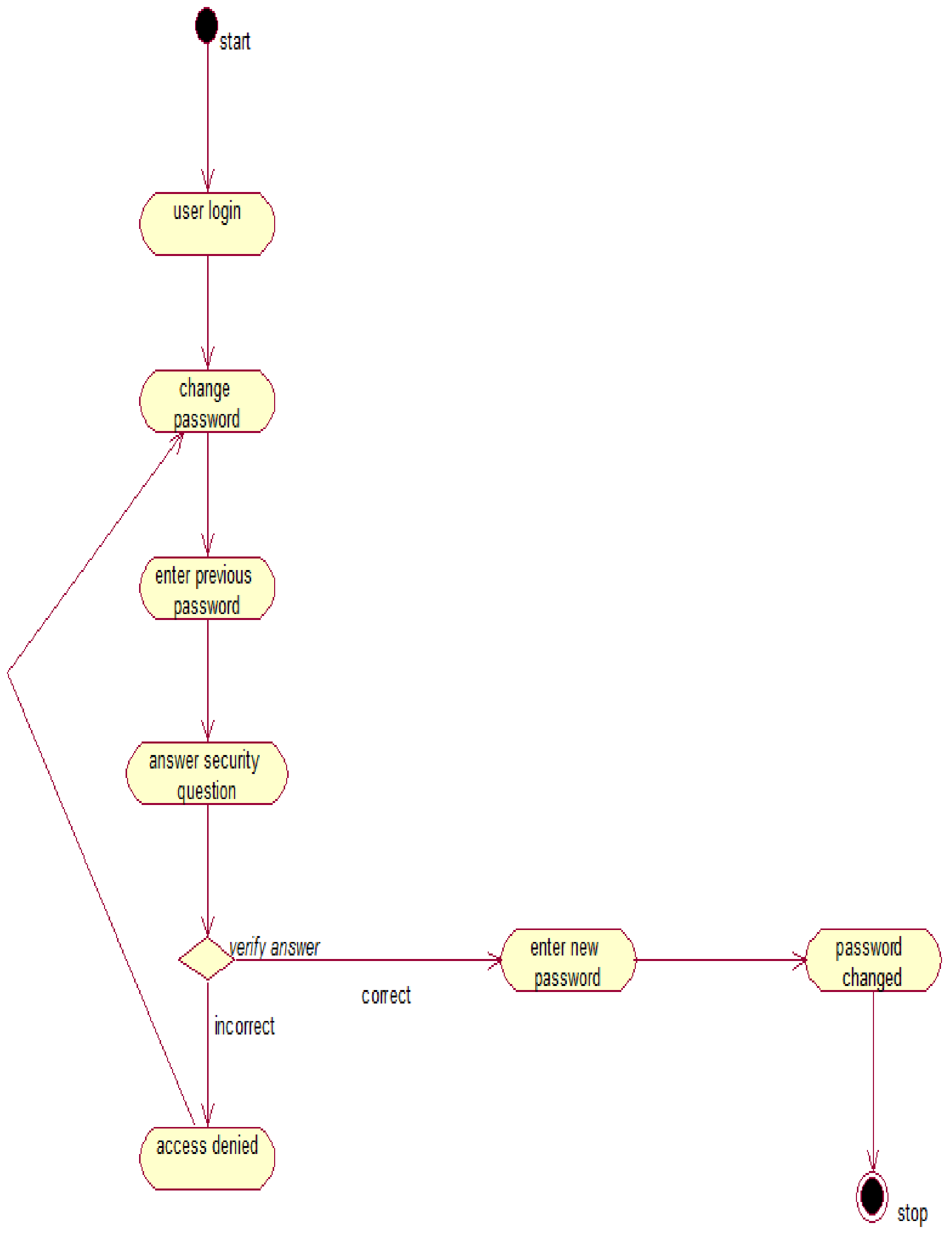

Fig 3.9 Update password

3.2.7 Download/view files activity

If the user wants to download or view a file, he enters the file name. If the file is present, the user is asked if he wants to view or download the file and accordingly the activity is performed.

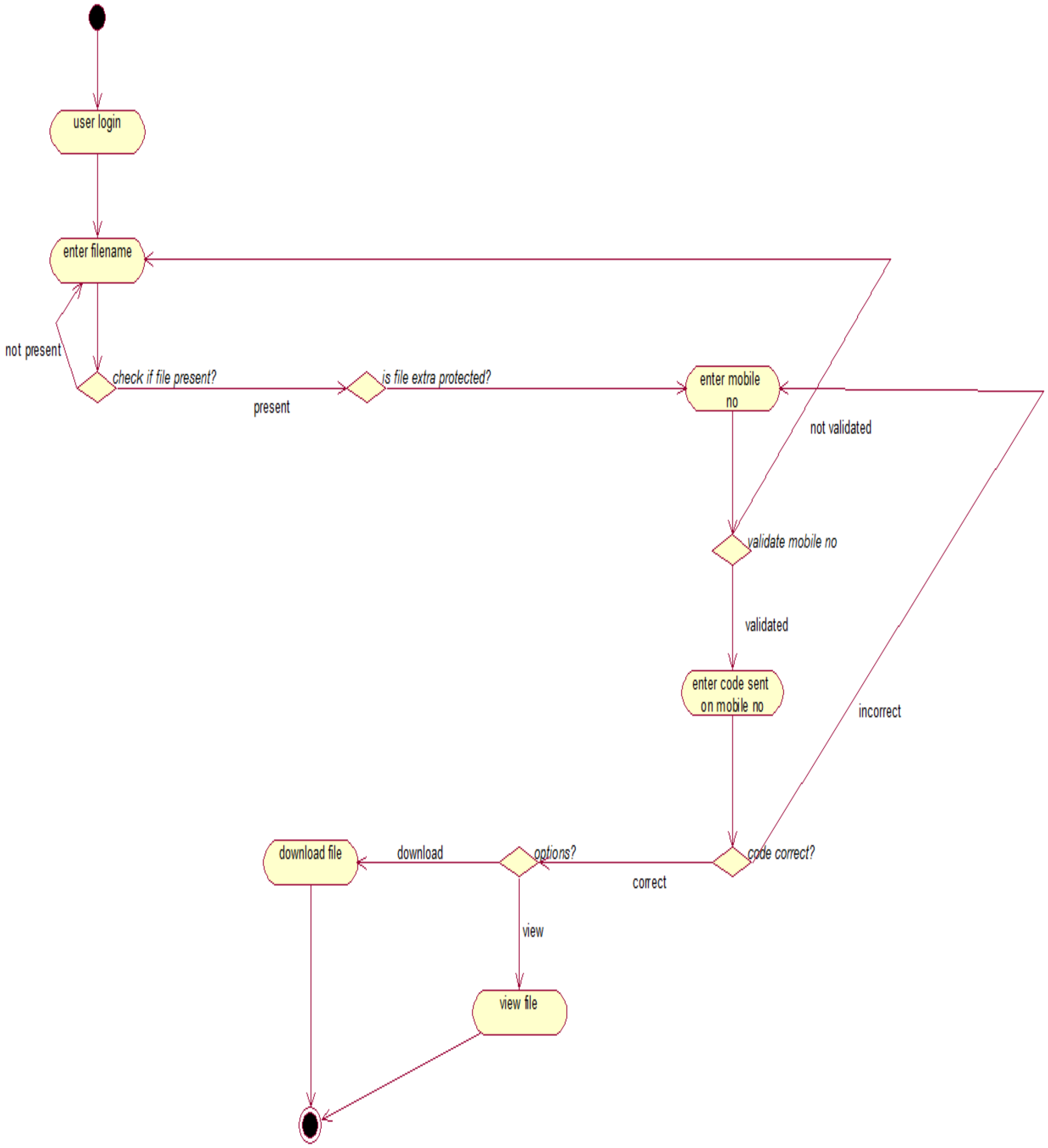

Fig 3.10 Download and view files

3.2.8 Admin activity

The admin logs into the system and authenticates new users of the system. The admin also grants/access privileges to the users.

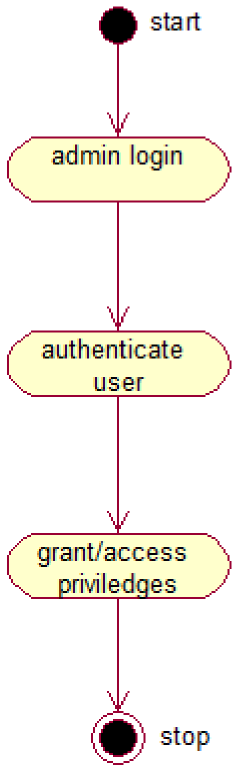

Fig 3.11 Admin activity

3.3 SEQUENCE DIAGRAM

3.3.1.1 **User Login**: Sequence Diagram

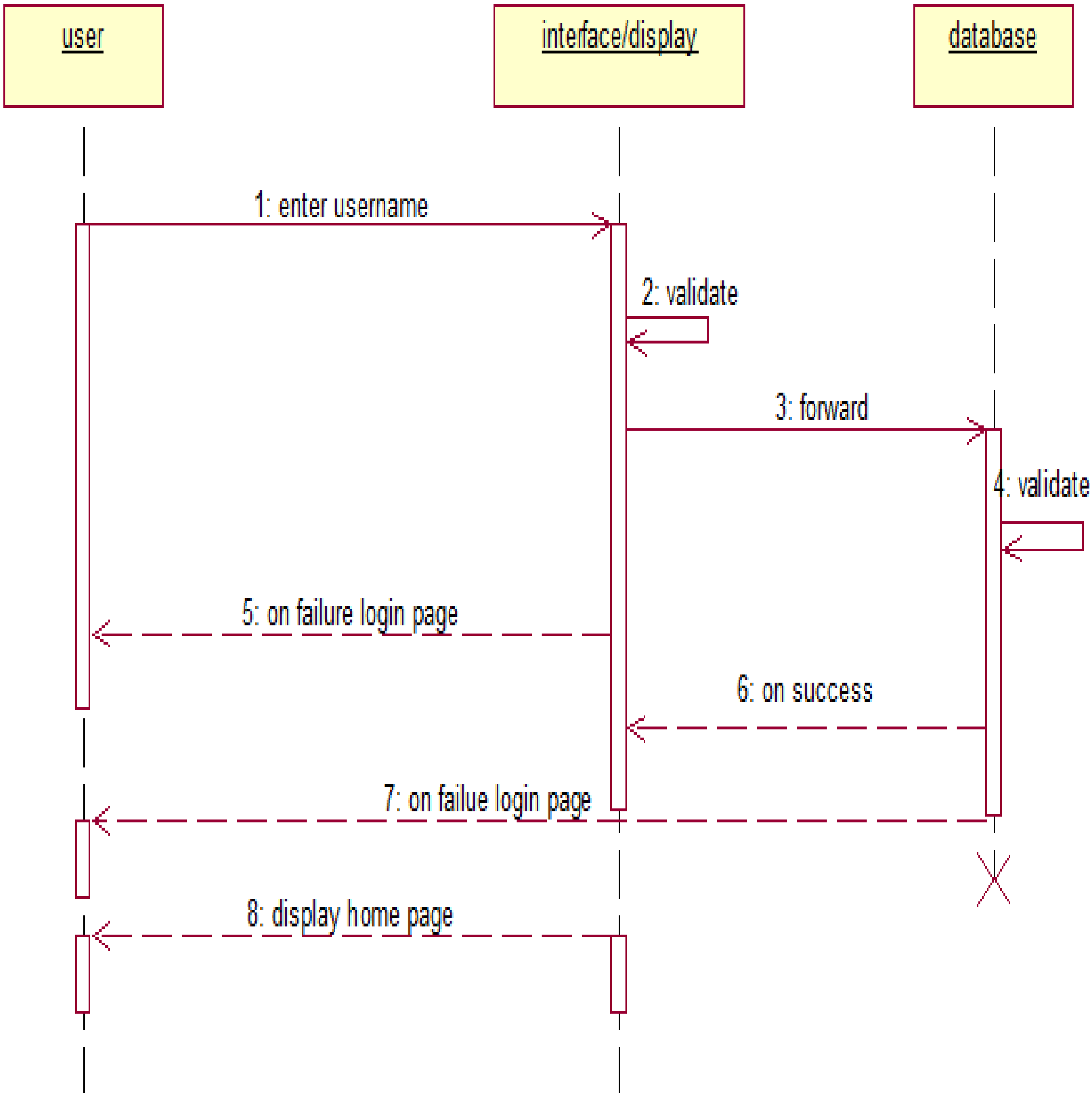

Fig 3.12.1 User Login

3.3.1.2 **User Login**: Deployment Diagram

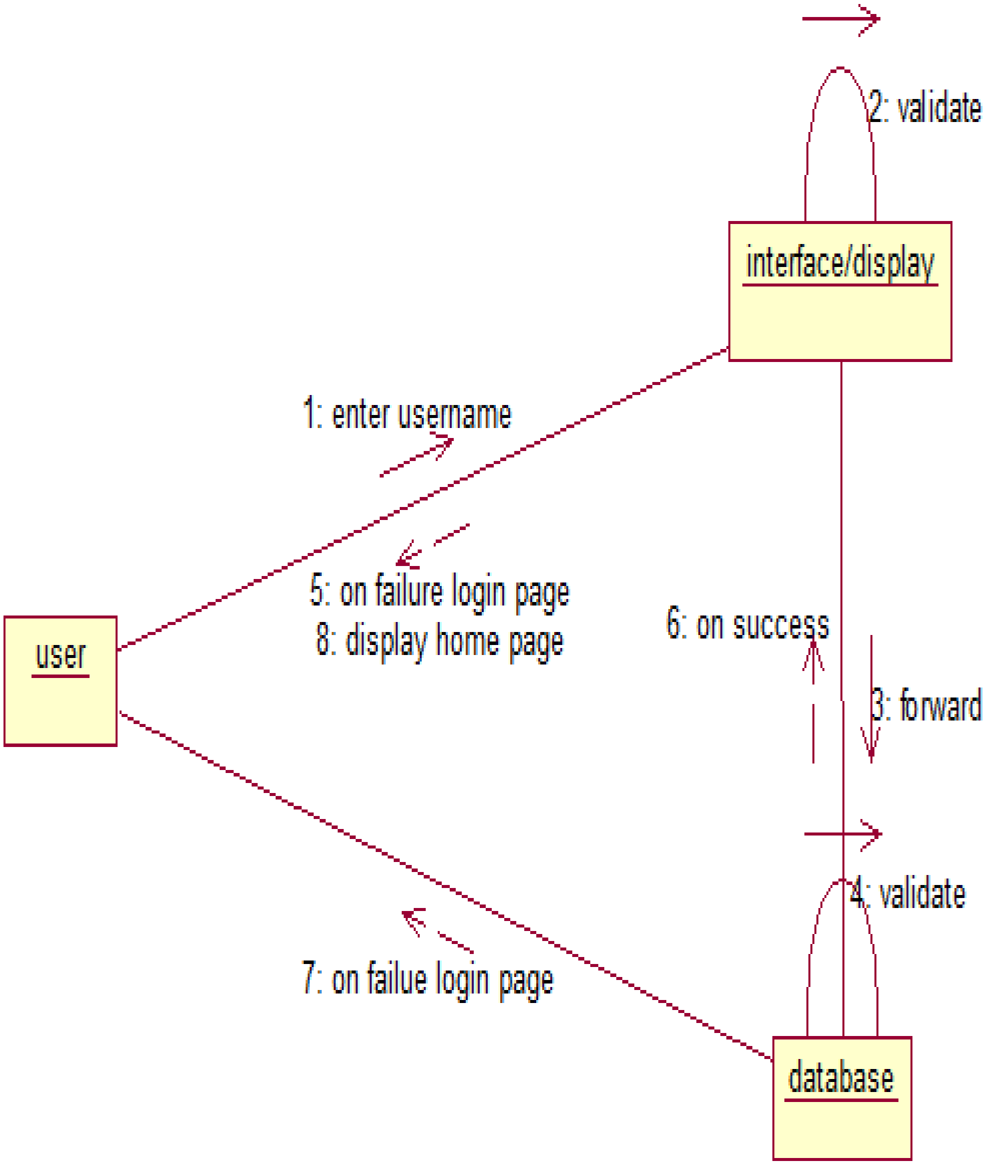

Fig 3.12.2 User Login

3.3.2.1 **Add files**: Sequence Diagram

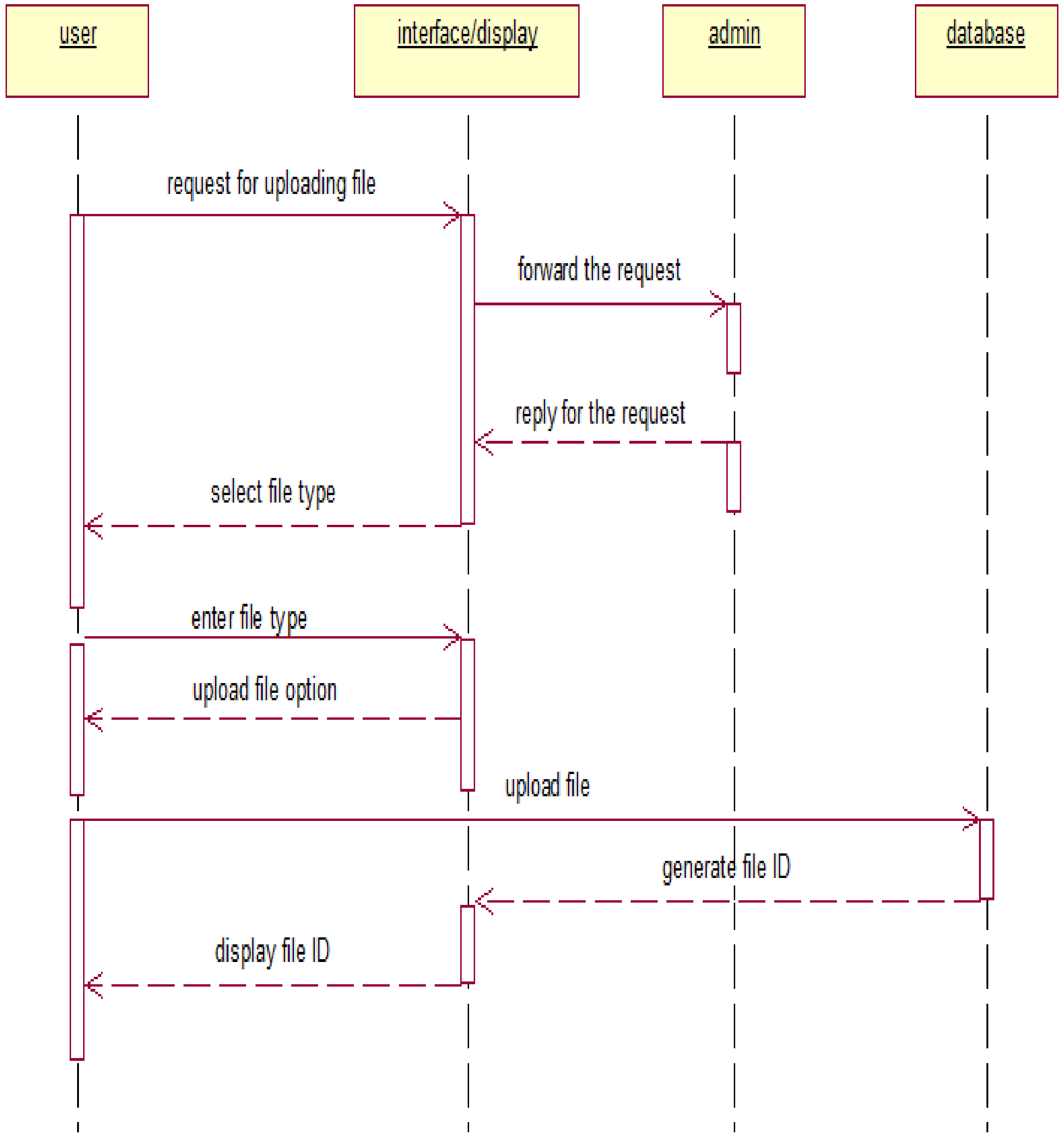

Fig 3.13.1 Add Files

3.3.2.2 **Add Files**: Deployment Diagram

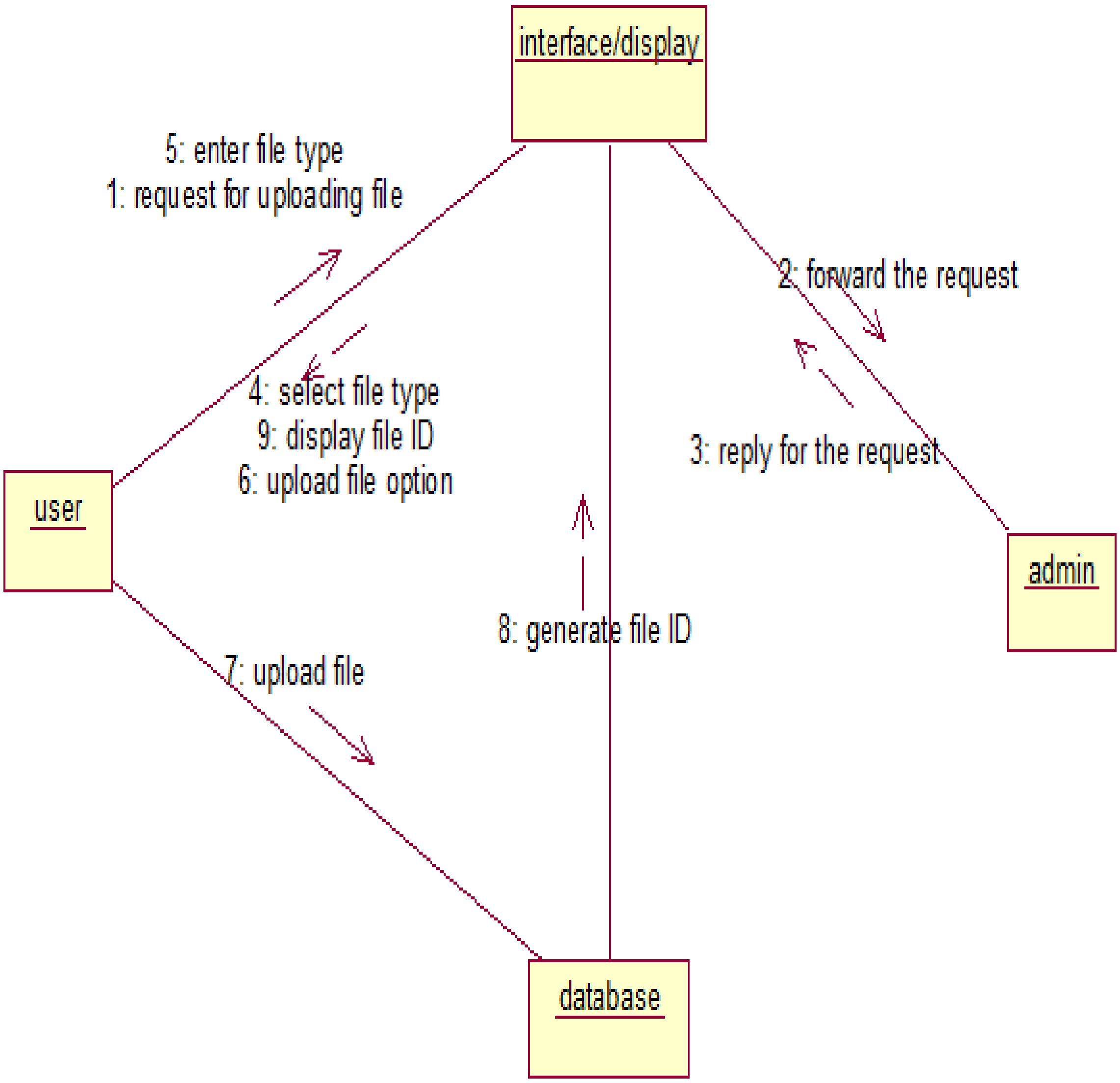

Fig 3.13.2 Add Files

3.3.3.1 **Add Review:** Sequence Diagram

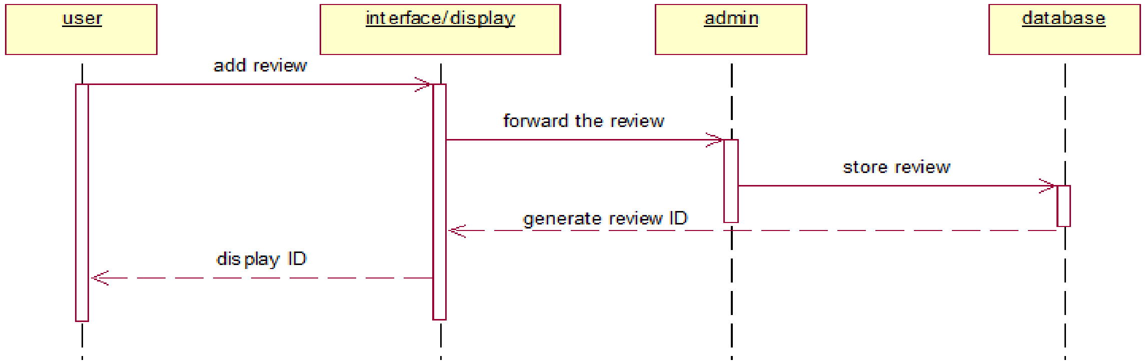

Fig 3.14.1 Add Review

3.3.3.2 **Add Review:** Deployment Diagram

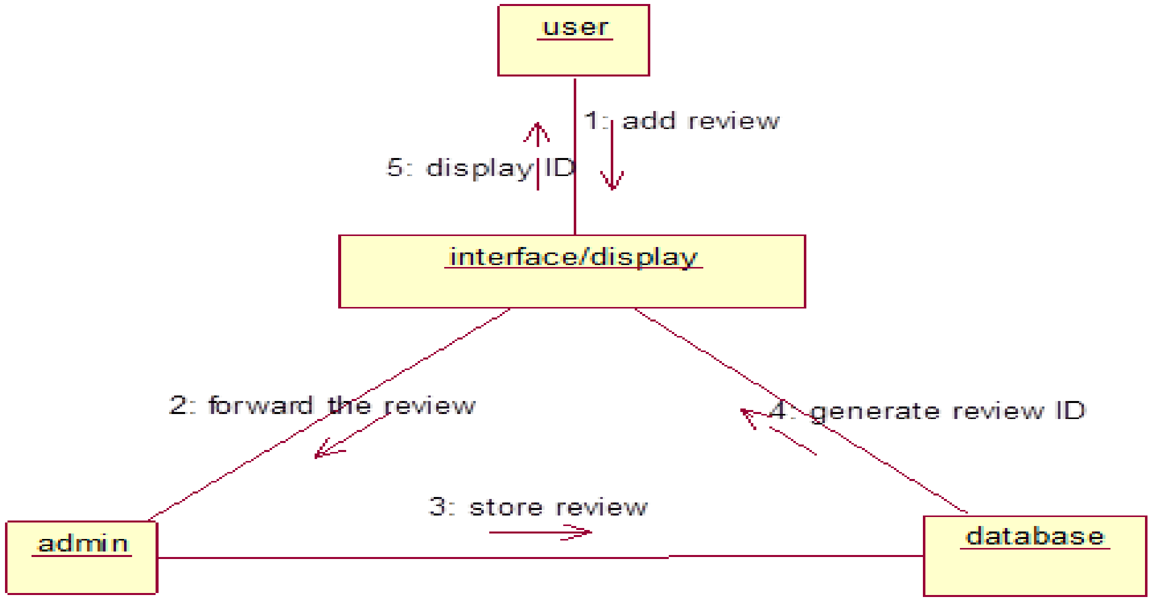

Fig 3.14.2 Add Review

3.3.4.1 **Extra Protection**: Sequence Diagram

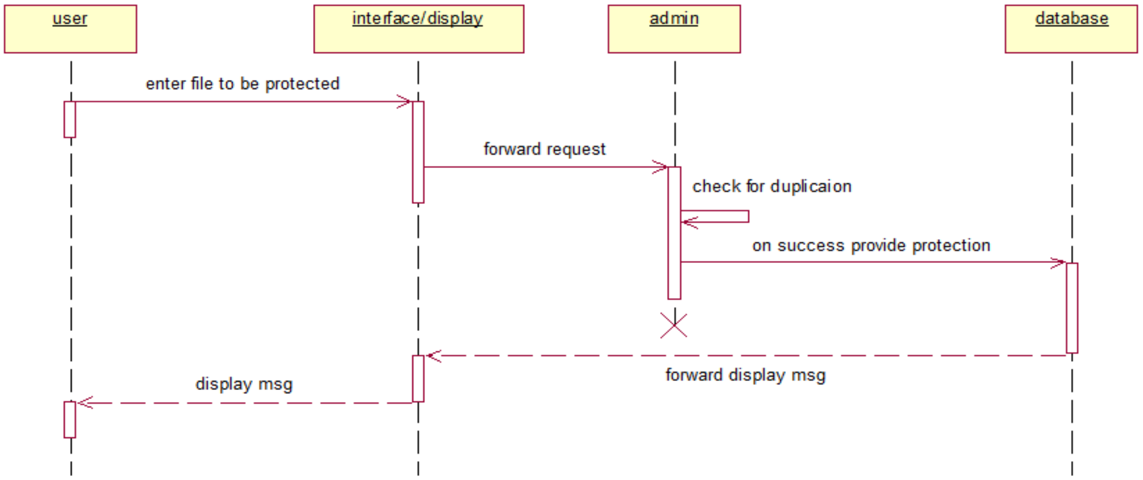

Fig 3.15.1 Extra Protection

3.3.4.2 **Extra Protection**: Deployment Diagram

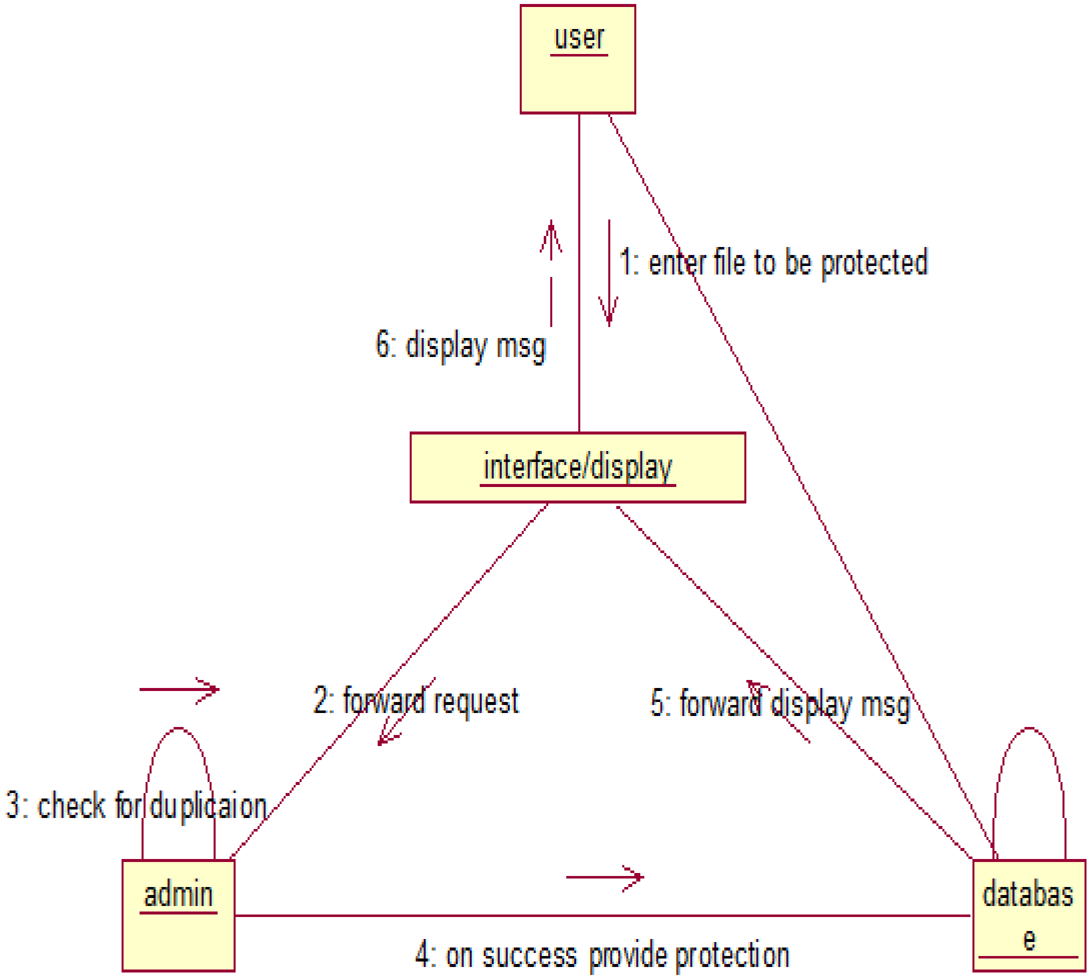

Fig 3.15.2 Extra Protection

3.3.5.1 **Forgot Password**: Sequence Diagram

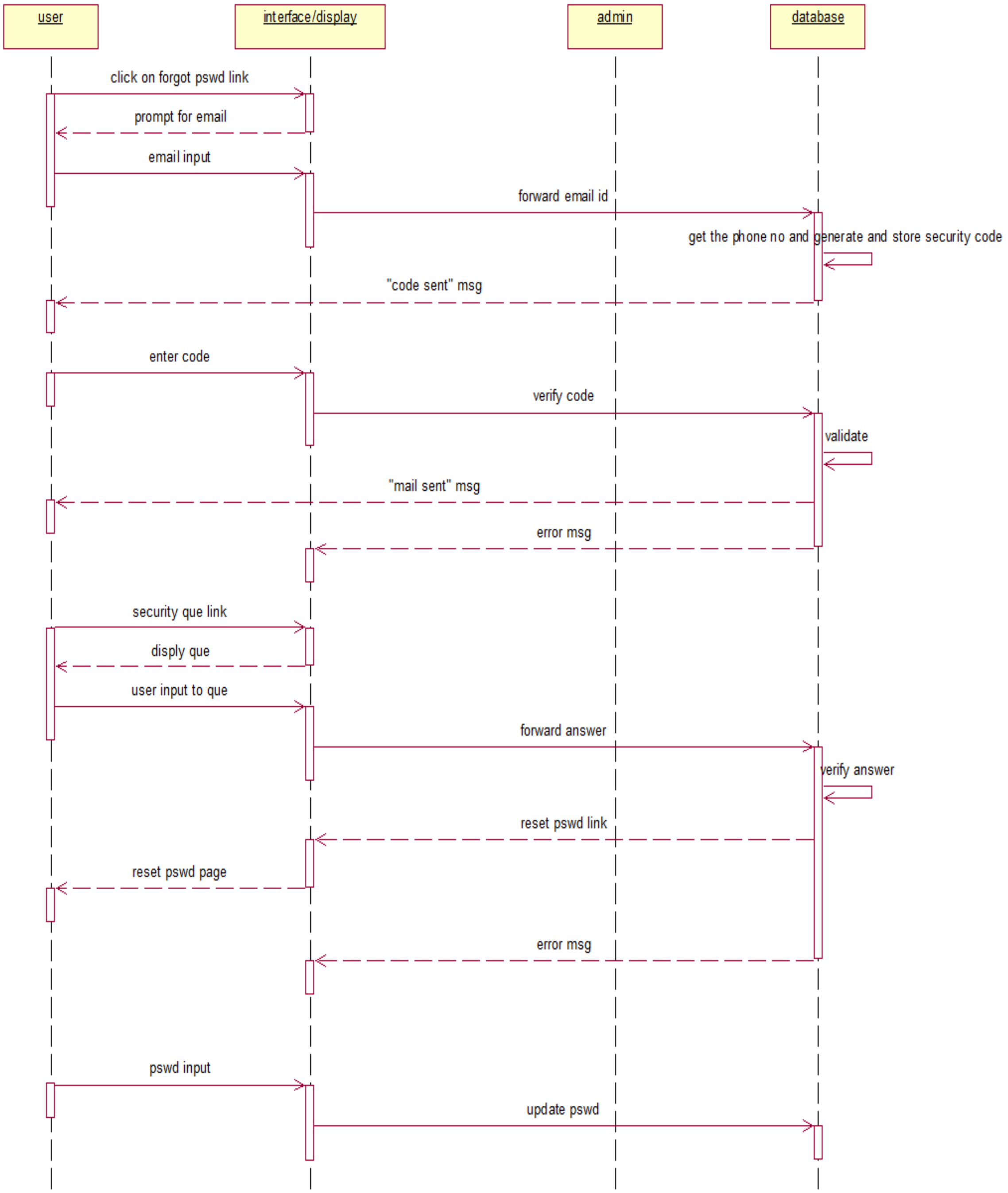

Fig 3.16.1 Forgot password

3.3.5.2 **Forgot Password**: Deployment Diagram

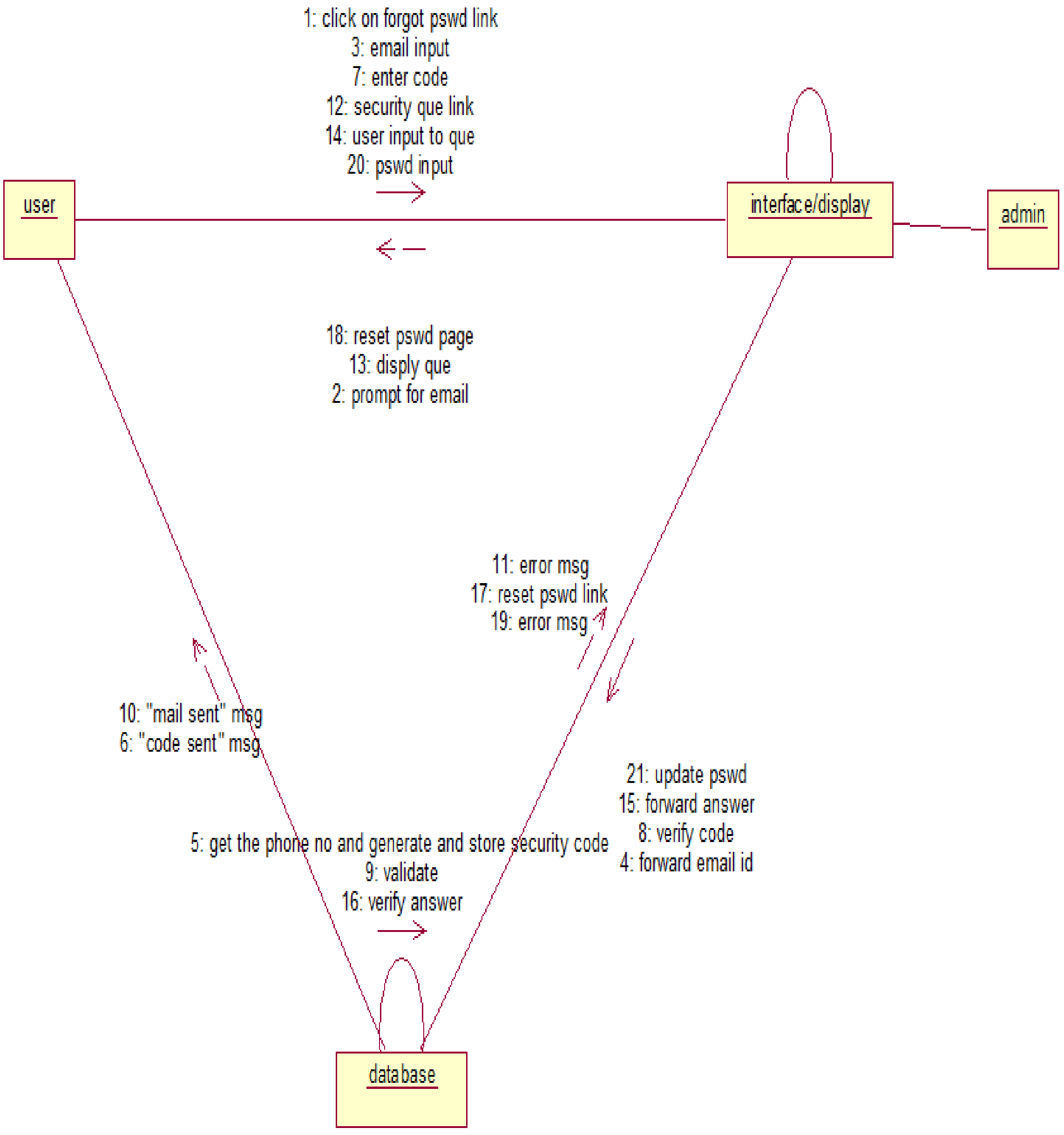

Fig 3.16.2 Forgot password

3.3.6.1 **Share File**: Sequence Diagram

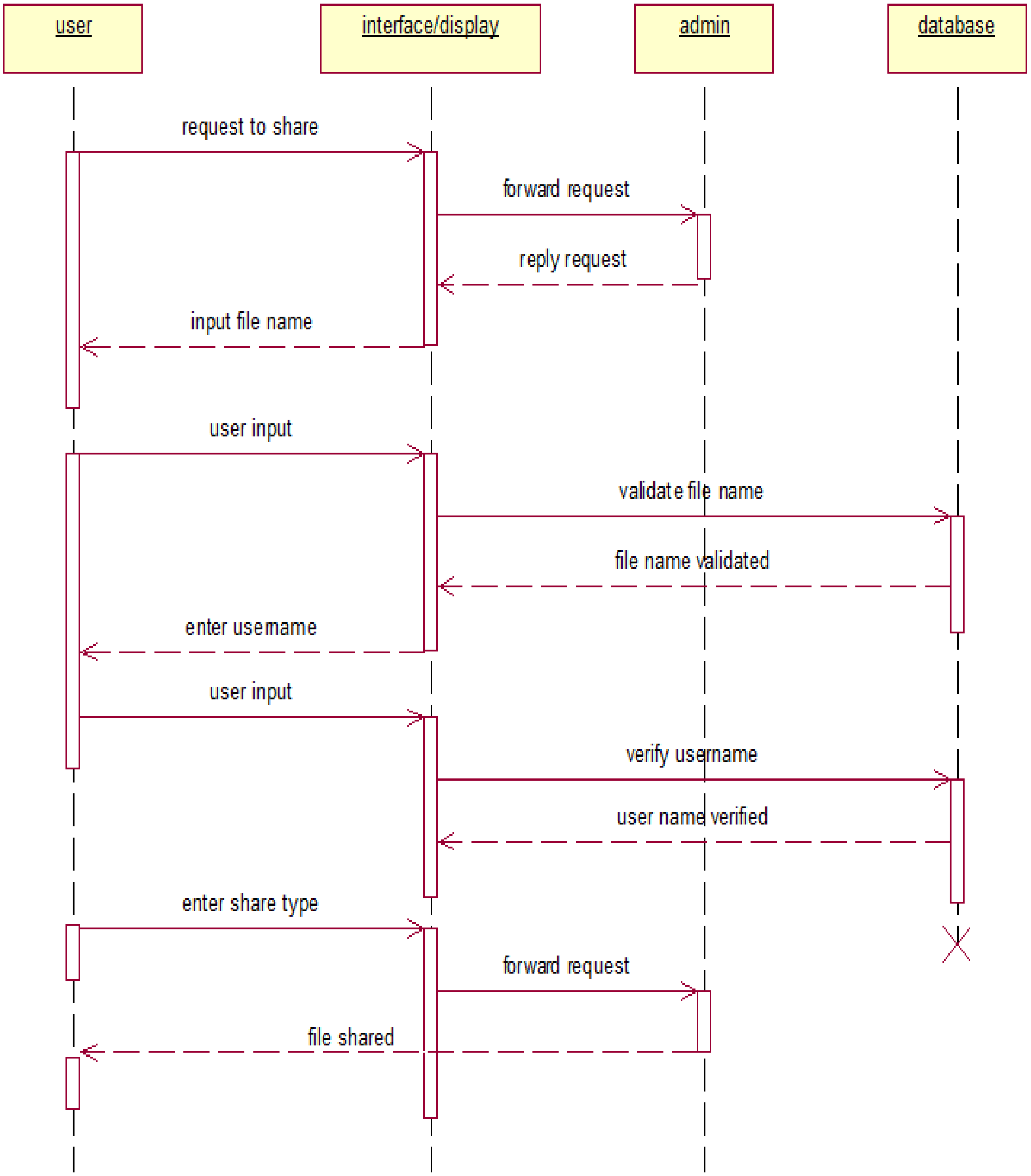

Fig 3.17.1 Share File

3.3.6.2 **Share File**: Deployment Diagram

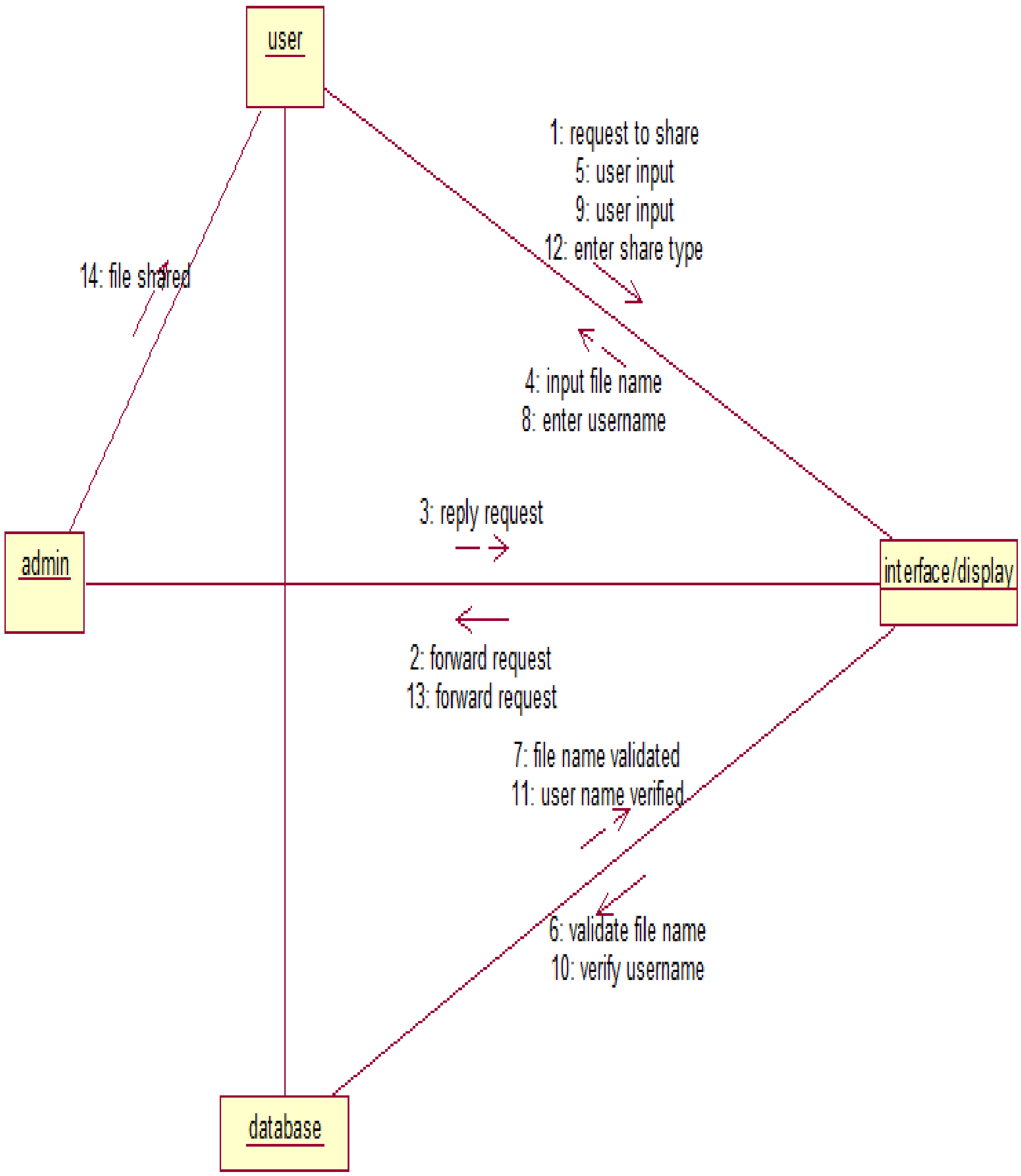

Fig 3.17.2 Share File

4. REFERENCES

- ✓ Michael Blaha and James Rambaugh, *Object Oriented Modelling and Design with UML*
- ✓ Roger Pressman, *Software Engineering*, Sixth Edition
- ✓ Navathe, *Fundamentals of Database Systems*, Fifth Edition
- ✓ Henry Korth, *Database System Concept*, Fifth Edition
- ✓ Project Specification provided by IBM
- ✓ www.ibm.in/developerworks
- ✓ *www.wikipedia.com*