

HNTechno Follow Up

A PROJECT REPORT

Submitted By

Ayushi Patel (130770107085)

Kosha Patel (130770107097)

In fulfillment for the award of the degree

of

BACHELOR OF ENGINEERING

in

Computer Engineering



**SILVER OAK COLLEGE OF ENGINEERING AND TECHNOLOGY,
AHMEDABAD**

Gujarat Technological University, Ahmedabad

May, 2017



GUJARAT TECHNOLOGICAL UNIVERSITY

CERTIFICATE FOR COMPLETION OF ALL ACTIVITIES AT ONLINE PROJECT PORTAL

B.E. SEMESTER VIII, ACADEMIC YEAR 2016-2017

Date of certificate generation : 09 April 2017 (17:52:04)

This is to certify that, *Patel Ayushi Pankajbhai* (Enrolment Number - 130770107085) working on project entitled with *Hntechno Followup* from *Computer Engineering* department of *Silver Oak College Of Engineering & Technology, Ahmedabad* had submitted following details at online project portal.

| | |
|---------------------------------|-----------|
| Periodic Progress Reports (PPR) | Completed |
| Business Model Canvas (Image) | Completed |
| Business Model Canvas (Report) | Completed |
| Patent Drafting Exercise (PDE) | Completed |
| Final Plagiarism Report | Completed |
| Final Project Report | Completed |

Name of Student : Patel Ayushi Pankajbhai

Name of Guide : Mrs. Panam J. Shah

Signature of Student : _____

*Signature of Guide : _____

Disclaimer :

This is a computer generated copy and does not indicate that your data has been evaluated. This is the receipt that GTU has received a copy of the data that you have uploaded and submitted as your project work.

*Guide has to sign the certificate, Only if all above activities has been Completed.



GUJARAT TECHNOLOGICAL UNIVERSITY

CERTIFICATE FOR COMPLETION OF ALL ACTIVITIES AT ONLINE PROJECT PORTAL

B.E. SEMESTER VIII, ACADEMIC YEAR 2016-2017

Date of certificate generation : 09 April 2017 (17:50:17)

This is to certify that, *Patel Koshaben Kamaleshkumar* (Enrolment Number - 130770107097) working on project entitled with *Hntechno Followup* from *Computer Engineering* department of *Silver Oak College Of Engineering & Technology, Ahmedabad* had submitted following details at online project portal.

| | |
|---------------------------------|-----------|
| Periodic Progress Reports (PPR) | Completed |
| Business Model Canvas (Image) | Completed |
| Business Model Canvas (Report) | Completed |
| Patent Drafting Exercise (PDE) | Completed |
| Final Plagiarism Report | Completed |
| Final Project Report | Completed |

Name of Student : Patel Koshaben
Kamaleshkumar

Name of Guide : Mrs. Panam J. Shah

Signature of Student : _____

*Signature of Guide : _____

Disclaimer :

This is a computer generated copy and does not indicate that your data has been evaluated. This is the receipt that GTU has received a copy of the data that you have uploaded and submitted as your project work.

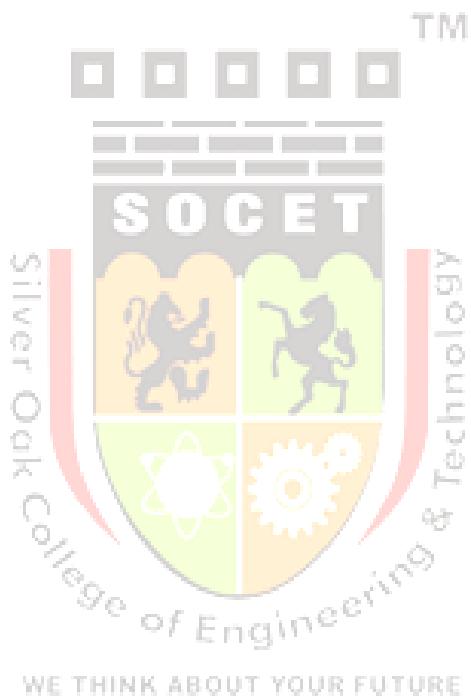
*Guide has to sign the certificate, Only if all above activities has been Completed.

Silver Oak College of Engineering and Technology
Computer Engineering
2017

CERTIFICATE

Date:

This is to certify that the Project Work entitled “HNTechno Follow Up” has been carried out by Patel Ayushi Pankajbhai (130770107085) under my guidance in fulfilment of the degree of Bachelor of Engineering in Computer (8th Semester) of Gujarat Technological University, Ahmedabad during the academic year 2016-17.



Internal Guide

SOCET

Head of Computer Engineering Department

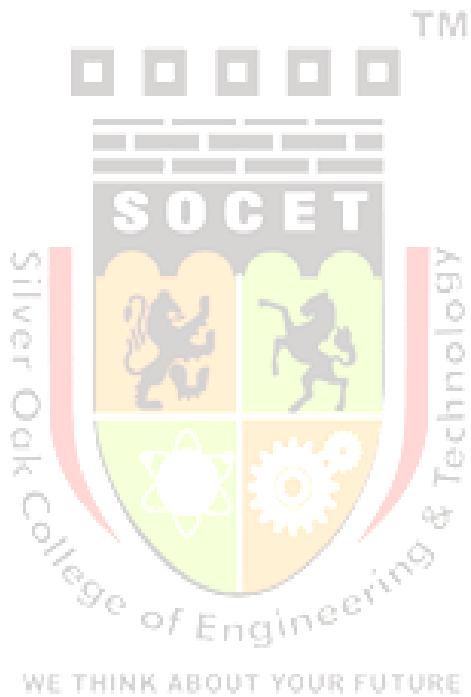
SOCET

Silver Oak College of Engineering and Technology
Computer Engineering
2017

CERTIFICATE

Date:

This is to certify that the Project Work entitled “HNTechno Follow Up” has been carried out by Patel Kosha Kamaleshbhai (130770107097) under my guidance in fulfilment of the degree of Bachelor of Engineering in Computer (8th Semester) of Gujarat Technological University, Ahmedabad during the academic year 2016-17.



Internal Guide

SOCET

Head of Computer Engineering Department

SOCET



Email Id:hr@hntechno.com
3 ,Sagar Tower,
Opp. Kalasagar Mall,
Ghatlodia.
Ahmedabad-380061
Contact:9173026598

22nd April, 2017

Ahmedabad

Sub: To Whomsoever It May Concern

This is to inform whomsoever it may concern and certify that **PATEL AYUSHI PANKAJBHAI** has worked as a **Trainee of JAVA Language** at **HN Techno** since **15th July, 2016 till 31st May, 2017**.

She has worked on project title **HNTechno Follow Up**. This project was developed in **JAVA Technology**. She demonstrated good skill with a self-motivated attitude to learn new things.

The company will not provide any type of source Code outside the premises. The management wish them good luck for future.

Techno
Delivering Advance Technologies

Regards,
For, HN Techno

Vijay Prajapati
(CEO and Founder)

Deal in Web site designing and Development.

| www.hntechno.com



Email Id:hr@hntechno.com
3 ,Sagar Tower,
Opp. Kalasagar Mall,
Ghatlodia.
Ahmedabad-380061
Contact:9173026598

22nd April, 2017

Ahmedabad

Sub: To Whomsoever It May Concern

This is to inform whomsoever it may concern and certify that **PATEL KOSHA KAMALESHBHAI** has worked as a Trainee of JAVA Language at HN Techno since 15th July, 2016 till 31st May, 2017.

She has worked on project title **HNTechno Follow Up**. This project was developed in **JAVA Technology**. She demonstrated good skill with a self-motivated attitude to learn new things.

The company will not provide any type of source Code outside the premises. The management wish them good luck for future.

Techno
Delivering Advance Technologies

Regards,
For, HN Techno

Vijay Prajapati
(CEO and Founder)

Deal in Web site designing and Development.

| www.hntechno.com

Candidate's Declaration

We hereby declare that project report titled "**HNTechno Follow Up**" submitted towards the completion of project in 8th semester of bachelor of Computer in Silver Oak College Of Engineering & Technology, Ahmedabad is an authenticate record of our work carried out.

We further declare that to the best of our knowledge the report of C.E. 8th semester.

| | |
|-----------------------|--------------------------|
| Candidate's signature | : |
| Candidate's name | : Ayushi P. Patel |
| Branch | : CE |
| Enrollment Number | : 130770107085 |

| | |
|-----------------------|-------------------------|
| Candidate's signature | : |
| Candidate's name | : Kosha P. Patel |
| Branch | : CE |
| Enrollment Number | : 130770107097 |



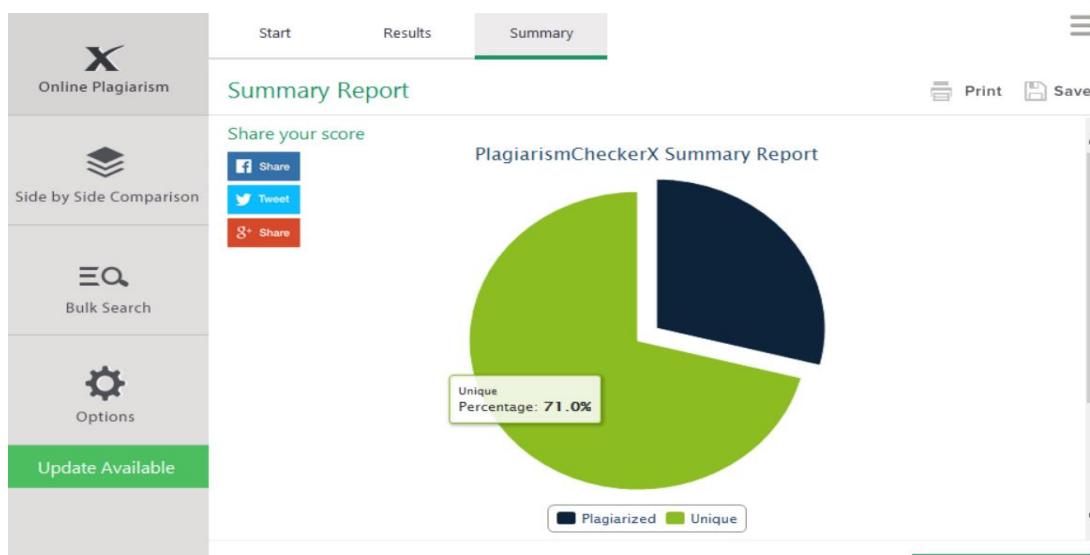
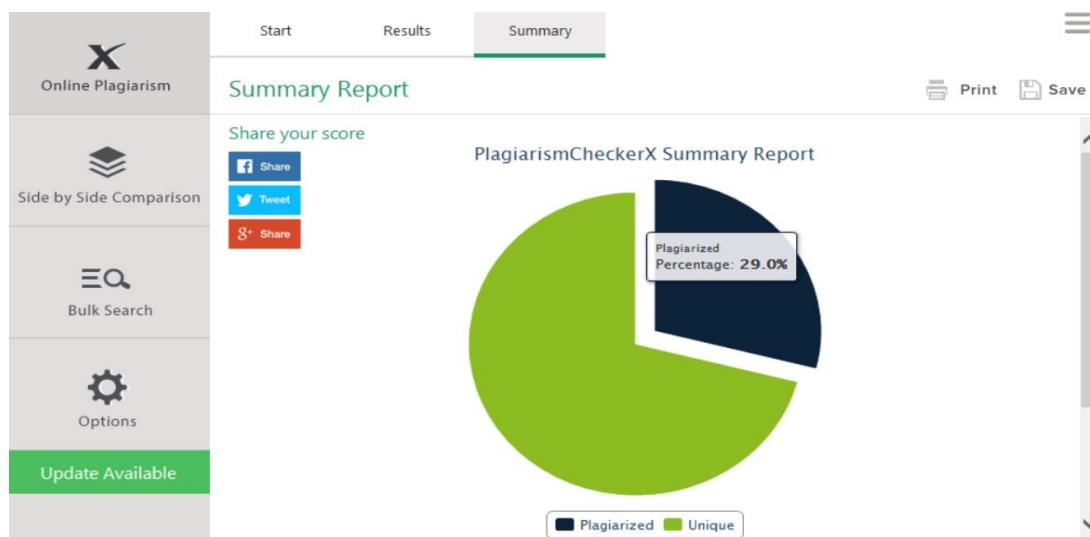
Plagiarism Checker X Originality Report

Similarity Found: 28%

Date: Monday, April 03, 2017

Statistics: 4016 words Plagiarized / 14200 Total words

Remarks: Medium Plagiarism Detected - Your Document needs Selective Improvement.



ACKNOWLEDGEMENT

The successful completion of this project rests on the shoulders of many persons who have helped us directly or indirectly. I wish to take this opportunity to express to all those, without whose help, completion of this project would have been difficult.

I am glad to submit my project on **HNTechno Follow Up**. The system which I have designed and created is the result of many people's dedication. I would like to convey our heartiest thanks to **Ms. Pooja Jardosh** for providing me a proper guideline in development of project and also thanks to our Head of Department for solving my difficulties and bringing this project together.

At last I would like to thanks all faculty members of College who gave us proper guideline and helped me a lot in completing this project successfully.

Yours Sincerely,

Patel Ayushi P.

(130770107085)

Patel Kosha K.

(130770107097)

ABSTRACT

To make a best product for an IT company named HNTechno which is known for its developing and training in Ahmedabad. This web application of HNTechno Company will include basic and advanced functionality any IT company can desire for.

There are features like viewing demo lectures, viewing faculty list, taking appointment, task submission, attendance, getting certificate, downloading material etc. in the system.

The main aim of our project is to create convenient and easy to use system for users of the system, students, and faculties. User will register themselves in HNTechno to get involved in the system. Student will be able to download lectures and submitting task and getting certificate in the end. System will be developed in such a way that it is useful and easy in use.

COMPANY PROFILE

Welcome to HN Techno

HN Techno always strives to ensure that trainings get better and better. We constantly upgrade ourselves, implement new ideas, revise curriculum and constantly work on the feedback. We strive for the betterment of our trainees. We know the real time work would be very different and challenging, we try our best to bridge the gap between training and the real time world. We try to focus on each and every individual. We ensure that every individual gets attention. We always assist our trainees in the practice lab to enhance their confidence.

Our courses are designed in such ways that are aligned to match current industry standards. It not only helps students to get real time experience but also helps the Companies to hire them and put them on to the projects within their timelines.



We aim at providing excellent customer satisfaction at all levels of Training & Development through constant self- improvement. Our mission is to materialize our vision of being a globally recognized and respected training organization through innovative and systematic approach. We want to enhance and nurture talent to reach its highest potential for the growth of technology.

Training

Started with a vision of feeding skills required to talent created by colleges, today HN Techno is one of the IT Training provider in PHP Training, Java Training, Android Training, Web Design Training and ASP.Net. Today we have Software Development training center in Ahmedabad as one of the largest Software Development Company in Gujarat.

Development

Creativity, Innovation and Professionalism are the essence of every website that HN Techno has developed. Every facet of our designs are alluring and aesthetically chiseled. We believe in designs that are simple, attractive and yet, fast to load on the browser. Furthermore, we offer expert search engine optimization, superior designs and programming to ensure your website has a positive impact for your brand.

Outsourcing

HN Techno has emerged as a leading offshore Outsourcing company. We work on the model of offshore Outsourcing services with Local Presence. By working in our client's time zones, the locally based team members ensure that optimal communication is maintained for the successful outcome of outsourced projects. Our project leader conduct on site documentation of requirements. They ensure the offshore team has a pipeline of work orders to zero out idle

time. They ensure consistent measurable delivery on or ahead of schedule throughout our engagement to bring great products to market in less time and at less cost.

LIST OF FIGURE

| Fig. No | Description | Page No |
|----------------|--------------------------------|----------------|
| 2.1 | Waterfall Model | 8 |
| 2.2 | Gantt Chart | 10 |
| 4.1 | Use Case – Visitor | 28 |
| 4.2 | Use Case – Student | 29 |
| 4.3 | Use Case – Faculty | 30 |
| 4.4 | Use Case – Admin | 31 |
| 4.5 | E-R Diagram | 34 |
| 4.6 | Activity Diagram | 36 |
| 4.7 | Context Diagram | 52 |
| 4.8 | 0 level DFD – 1 | 54 |
| 4.9 | 0 level DFD – 2 | 54 |
| 4.10 | 1 level DFD – Admin | 55 |
| 4.11 | 1 level DFD – Faculty | 56 |
| 4.12 | 1 level DFD – Student | 57 |
| 5.1 | Table and Relationship Diagram | 64 |
| 5.2 | Flow Chart | 66 |
| 8.1 | Home Page | 84 |
| 8.2 | Services | 84 |
| 8.3 | Inquiry | 85 |
| 8.4 | Contact us | 85 |
| 8.5 | Login Page | 86 |
| 8.6 | Language Page | 86 |
| 8.7 | Batch Details | 87 |
| 8.8 | Student Registration | 87 |
| 8.9 | Field Page | 88 |
| 8.10 | Faculty and student details | 88 |

LIST OF TABLE

| Table No | Description | Page No |
|-----------------|---------------------------|----------------|
| 2.1 | Risk Analysis | 13 |
| 4.1 | Use Case Symbols | 27 |
| 4.2 | E-R Diagram Symbols | 33 |
| 4.3 | Activity Diagram Symbols | 35 |
| 4.4 | Inquiry | 37 |
| 4.5 | Registration | 38 |
| 4.6 | User | 39 |
| 4.7 | Appointment | 40 |
| 4.8 | Demo | 41 |
| 4.9 | Batch | 42 |
| 4.10 | Course | 42 |
| 4.11 | Sub Course | 43 |
| 4.12 | College | 44 |
| 4.13 | State | 45 |
| 4.14 | Field | 45 |
| 4.15 | University | 46 |
| 4.16 | Language | 47 |
| 4.17 | Create Batch | 47 |
| 4.18 | City | 48 |
| 4.19 | Task | 49 |
| 4.20 | Material | 49 |
| 4.21 | Role | 50 |
| 4.22 | Faculty | 50 |
| 4.23 | Privilege | 51 |
| 4.24 | Data Flow Diagram Symbols | 53 |
| 4.25 | Hardware Justification | 59 |
| 4.26 | Software Justification | 60 |
| 7.1 | Login | 80 |
| 7.2 | Registration | 81 |
| 7.3 | File Upload Format | 81 |
| 7.4 | File Size Upload Format | 81 |

ABBREVIATION

| | |
|--------------|---------------------------------------|
| SQL | Structured Query Language |
| CSS | Cascading Style Sheet |
| HTML | Hypertext Markup Language |
| GB | Gigabyte |
| JSP | Java Server Pages |
| UML | Unified Modeling Language |
| API | Application Programming Interface |
| RDBMS | Relational Database Management System |
| DFD | Data Flow Diagram |
| RAM | Random Access Memory |
| PPR | Periodic Progress Report |
| BMC | Business Model Canvas |
| PDE | Patent Drafting Exercise |

TABLE OF CONTENT

| | |
|---|-----|
| ACKNOWLEDGEMENT | x |
| ABSTRACT | xi |
| COMPANY PROFILE..... | xii |
| LIST OF FIGURE | xiv |
| LIST OF TABLE | xv |
| ABBREVIATION..... | xvi |
| CHAPTER: 1..... | 1 |
| 1.1 PROJECT SUMMARY | 2 |
| 1.2 PURPOSE | 2 |
| 1.3 SCOPE | 3 |
| 1.4 TECHNOLOGY AND LITERATURE REVIEW..... | 3 |
| CHAPTER: 2..... | 7 |
| 2.1 PROJECT PLANNING & SCHEDULING | 8 |
| 2.1.1 Project Development Approach | 8 |
| 2.1.2 Project Plan | 8 |
| 2.1.3 Schedule Representation | 9 |
| 2.2 RISK MANAGEMENT | 10 |
| 2.2.1 Risk Identification | 11 |
| 2.2.2 Risk Analysis..... | 12 |
| 2.2.3 Risk Planning | 13 |
| 2.3 ESTIMATION | 14 |
| 2.3.1 Effort Estimation | 14 |
| 2.3.2 Cost Analysis..... | 14 |
| CHAPTER: 3..... | 17 |
| 3.1 USER CHARACTERISTICS | 18 |
| 3.2 HARDWARE AND SOFTWARE REQUIREMENTS..... | 19 |
| 3.3 CONSTRAINTS | 19 |
| CHAPTER: 4..... | 22 |
| 4.1 STUDY OF CURRENT SYSTEM | 23 |

| | |
|---|----|
| 4.2 PROBLEM AND WEAKNESSES OF CURRENT SYSTEM | 23 |
| 4.3 REQUIREMENTS OF NEW SYSTEM | 23 |
| 4.4 FEASIBILITY STUDY | 25 |
| 4.5 REQUIREMENTS VALIDATION | 26 |
| 4.6 FUNCTIONS OF SYSTEM..... | 27 |
| 4.6.1 USE CASE | 27 |
| 4.7 DATA MODELING | 32 |
| 4.7.1 E-R DIAGRAMS | 32 |
| 4.7.2 SYSTEM ACTIVITY | 35 |
| 4.7.3 DATA DICTIONARY | 37 |
| 4.8 FUNCTIONAL AND BEHAVIORAL MODELING | 52 |
| 4.8.1 CONTEXT DIAGRAM | 52 |
| 4.8.2 DATA FLOW DIAGRAM (0 and 1 level) | 53 |
| 4.9 MAIN MODULES OF NEW SYSTEM | 58 |
| 4.10 SELECTION OF HARDWARE AND SOFTWARE AND JUSTIFICATION | 59 |
| CHAPTER: 5..... | 61 |
| 5.1 DATABASE DESIGN/DATABASE STRUCTURE DESIGN | 62 |
| 5.1.1 Mapping Objects | 62 |
| 5.1.2 Table and Relationship..... | 63 |
| 5.2 SYSTEM PROCEDURAL DESIGN..... | 64 |
| 5.2.1 Flow Chart..... | 66 |
| CHAPTER: 6..... | 67 |
| 6.1 IMPLEMENTATION ENVIRONMENT | 68 |
| 6.2 MODULES SPECIFICATION..... | 68 |
| 6.3 SECURITY FEATURES | 69 |
| 6.4 CODING STANDARDS | 69 |
| 6.5 SAMPLE CODING..... | 70 |
| CHAPTER: 7..... | 76 |
| 7.1 TESTING PLAN..... | 77 |
| 7.2 TESTING STRATEGY | 77 |
| 7.3 TESTING METHODS | 78 |

| | |
|------------------------------|----|
| 7.4 TEST CASES | 79 |
| CHAPTER: 8..... | 83 |
| 8.1 SCREEN SHOTS..... | 84 |
| CHAPTER: 9..... | 89 |
| 9.1 LIMITATION | 90 |
| 9.2 FUTURE ENHANCEMENT | 90 |
| CHAPTER: 10..... | 91 |
| 10.1 CONCLUSION | 92 |
| REFERENCES..... | 93 |
| APPENDIX | 95 |

CHAPTER: 1

INTRODUCTION

1.1 PROJECT SUMMARY

1.2 PURPOSE

1.3 SCOPE

1.4 TECHNOLOGY AND LITERATURE REVIEW

1.1 PROJECT SUMMARY

This Project is aimed at developing a system for HNTechno. It is an IT company named which is known for its developing and training in Ahmedabad.

This System can be used to solve problems of users such as visiting, asking for an appointment to resolve doubts in their mind. Any user who wishes to visit company website can get basic information, demo lecture, demo material, faculty list. User gets himself registered into the system to interact with it.

Students who gets an admission and wish to interact through website can login into the system.

After logging in student can ask for a solution of a problem openly to other students and share his/her knowledge to other students not only that chatting with faculties and other students is also available. Students can download Video, material also. Alert of daily work is done via message or notification. Online payment can also be done. This web application will include a compiler so task given to students can be validated. Students who missed lectures can apply for recovery session. After finishing training, soft copy of training completion certificate is sent to students.

Admin manages number of faculties having different authority to manage sections. Admin will provide video of a lecture to students. Admin manages individual batch and corresponding timetable for each batch. Admin manages syllabus of each courses and training. Admin handles inquiries.

Faculty manages Syllabus of each courses and training is managed by them. Submitted task of students are reviewed by them and grades are also given.

1.2 PURPOSE

Goal & Objective:

The main aim of our project is to create convenient and easy to use system for users of the system, students, and faculties. The purpose behind developing this web application is to provide basic information to user, to facilitate them to get an appointment, to provide them list of faculties, to get payment done, to make registration process easy. Student will be facilitated

to request for leave, to view fees schedule, to chat with other users, to download material and videos, to submit a task, to get certificate. Faculty will manage syllabus and batch of the HNTechno and will upload tasks to students and uploads material and videos and gives grade. Admin will manage certificate and will handle inquiries, syllabus, and time table.

1.3 SCOPE

This system will be made to facilitate users to get connected with HNTechno.

They can get connected in the system by registering themselves in the system and they will be aware of any updation in HNTechno.

The System can facilitate users, faculty to reduce their work load and can perform their tasks smoothly. Student doesn't have to submit proof for registration process. Students can download videos and material from their lectures so that they don't have to go to faculty to ask for material. Student can submit task and so they don't have to go to faculty manually to check their tasks. Student doesn't have to ask for fees receipts and certificate after completion of the course or training. From Student's grade history faculty can hire them and offer them a job.

1.4 TECHNOLOGY AND LITERATURE REVIEW

Technology and Tools

➤ PLATFORM :-

- Technology :- PHP
- Front end :- HTML, JavaScript, CSS
- Back end :- MySQL server

➤ LITERATURE REVIEW:-

HTML:

- HTML an initialize of Hyper Text Markup Language for web pages.

- It provides a means to describe the structure of text based information in document by denoting text as headings, paragraphs, lists and so on and to supplement that text with interactive forms, embedded images and other objects.

JavaScript:

- JavaScript supports the development of both client and server components of web based applications.
- On the client side, it can be used to write programs that are executed by a web browser within the context of the web page.
- On the server side, it can be used to write web server programs that can be process information submitted by a web browser and then update the web browser display accordingly.

Advantages:

- It can be used for server side and client side scripting.
- It is more flexible than JavaScript.

PHP Introduction:

- PHP is an acronym for "PHP: Hypertext Preprocessor"
- PHP is a widely-used, open source scripting language
- PHP scripts are executed on the server

PHP File

- PHP files can contain text, HTML, CSS, JavaScript, and PHP code
- PHP code are executed on the server, and the result is returned to the browser as plain HTML.
- PHP files have extension “.php”

What Can PHP Do?

- PHP can generate dynamic page content
- PHP can create, open, read, write, delete, and close files on the server
- PHP can collect form data
- PHP can send and receive cookies

- PHP can add, delete, modify data in your database
- PHP can be used to control user-access
- PHP can encrypt data

Characteristics of PHP

Five important characteristics make PHP's practical nature possible –

- Simplicity
- Efficiency
- Security
- Flexibility
- Familiarity

Advantages of PHP

- Open source: It is developed and maintained by large group of PHP developers, this will helps in creating a support community.
- Speed: It is relative fast since it uses much system resource.
- Easy to use: It uses like C syntax, so for those who are familiar with C, it's very easy to create website scripts.
- Stable: Since it is maintained by many developers, so then bugs are found, it can be quickly fixed.
- Can be run on many platforms, including Windows, Linux and Mac, it's easy for users to find hosting service providers.
- PHP can be easily embedded directly into HTML.
- Interfaces very easily with Apache/MySQL.

MySQL, My Database

- MySQL is a database system used on the web
- MySQL is a database system that runs on a server
- MySQL is ideal for both small and large applications
- MySQL is very fast, reliable, and easy to use
- MySQL uses standard SQL

- MySQL compiles on a number of platforms
- MySQL is free to download and use
- MySQL is developed, distributed, and supported by Oracle Corporation
- The data in a MySQL database are stored in tables. A table is a collection of related data, and it consists of columns and rows.
- Databases are useful for storing information categorically. A company may have a database with the following tables:

Advantages of MySQL:

- It's fast. The main goal of the folks who developed MySQL was speed. Thus, the software was designed from the beginning with speed in mind.
- It's inexpensive. MySQL is free under the open source GPL license, and the fee for a commercial license is reasonable.
- It's easy to use. You can build and interact with a MySQL database by using a few simple statements in the SQL language, which is the standard language for communicating with RDBMSs.
- It can run on many operating systems. MySQL runs on many operating systems—Windows, Linux, Mac OS, most varieties of Unix (including Solaris and AIX), FreeBSD, OS/2, Irix, and others.
- It's available on almost all Web hosts. If you're going to run your Web site on a Web hosting company, MySQL is widely available without extra cost.
- Technical support is widely available. A large base of users provides free support through mailing lists. The MySQL developers also participate in the e-mail lists.
- It's secure MySQL's flexible system of authorization allows some or all database privileges (such as the privilege to create a database or delete data) to specific users or groups of users. Passwords are encrypted.

CHAPTER: 2

PROJECT MANAGEMENT

2.1 PROJECT PLANNING AND SCHEDULING

2.1.1 Project Development Approach

2.1.2 Project Plan

2.1.3 Schedule Representation

2.2 RISK MANAGEMENT

2.2.1 Risk Identification

2.2.2 Risk Analysis

2.2.3 Risk Planning

2.3 ESTIMATION

2.3.1 Effort Estimation

2.3.2 Cost Analysis

2.1 PROJECT PLANNING & SCHEDULING

2.1.1 Project Development Approach

There are times when the requirements of a problem are reasonably well understood when work flows from communication through deployment in a reasonably linear fashion. This situation is sometimes encountered when well-defined adaptations or enhancements to an existing system must be made. It may also occur in a limited number of new development efforts, but only when requirements are well-defined and reasonably stable.

The Waterfall model, sometimes called the classic life cycle, suggests a systematic, sequential approach to software development that begins with customer specification of requirements and progresses through Requirement, Design, Implementation, Verification and Maintenance culminating in on-going support of the completed software.

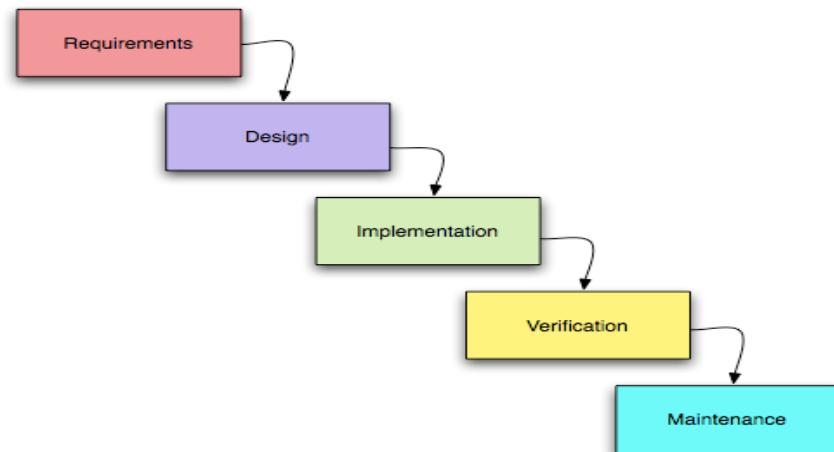


Figure 2.1 Waterfall Model

2.1.2 Project Plan

We have completed our project by Water Fall Model as listed below:

- 1. Requirement:** The basic requirements we have approached are to gathering our project.
- 2. Design:** The requirement are translated in some easy to represent from using which coding can be done effectively and efficiently. The design need to be documented for further use.

3. Implementation: If the design is done sufficient detail then Implementation can be done effectively.

4. Verification: We were toughly concentrating for the coding and testing and validation in modules, we succeeded at the end by imposing them on each other.

5. Maintenance: It enhancing system service as new requirement are discovered is again maintenance of the system.

When to use Waterfall Model:

- This model is used only when the requirements are very well known, clear and fixed.
- Product definition is stable.
- Technology is understood.
- There are no ambiguous requirements
- Ample resources with required expertise are available freely
- The project is short.

Risk in Waterfall Model

- Once an application is in the testing stage, it is very difficult to go back and change something.
- No working software is produced until late during the life cycle.
- High amounts of risk and uncertainty.

2.1.3 Schedule Representation

Gantt Charts:

Gantt Charts are useful tools for analyzing and planning more complex projects. When a project is under way, Gantt Charts help you to monitor whether the project is on schedule.

They are used to:

- Help you to plan out the tasks that need to be completed.
- Give you a basis for scheduling when these tasks will be carried out.
- Allow you to plan the allocation of resources needed to complete the project, and help

you to work out the critical path for a project where you must complete it by a particular date.

In below Gantt chart

- All the tasks are listed at the leftmost column.
- The horizontal bars indicate the time required by the corresponding task.
- When multiple horizontal bars occur at the same time on the calendar, then that means concurrency can be applied for performing the tasks.

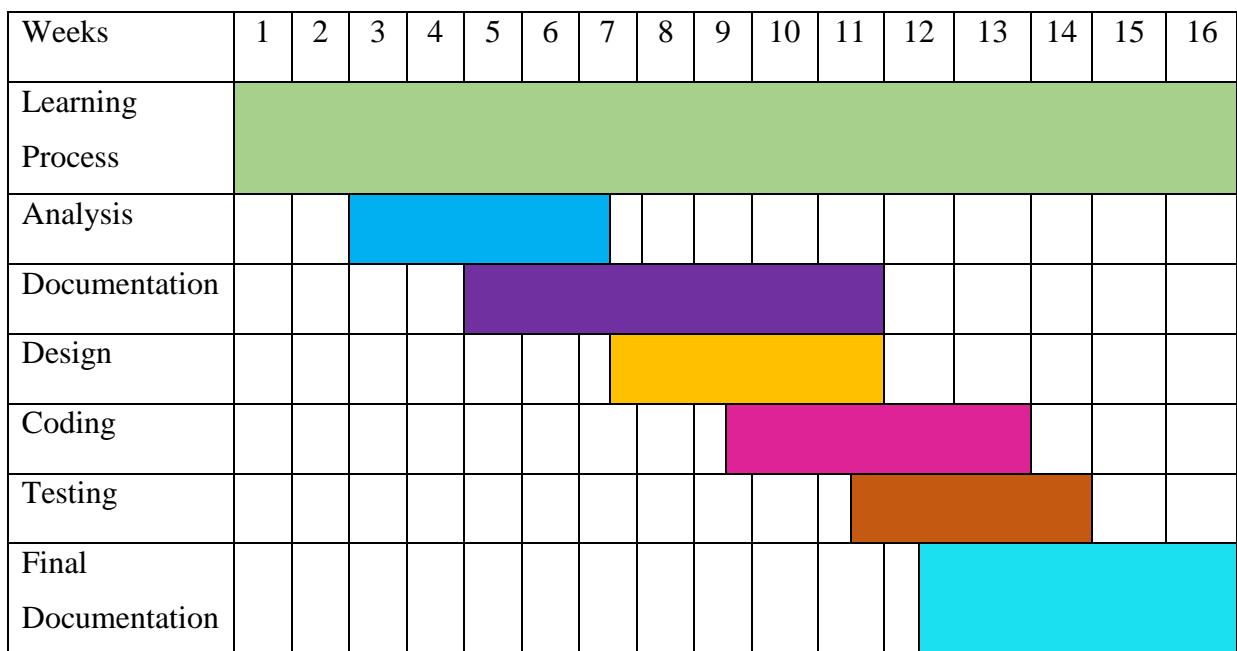


Figure 2.2 Gantt chart

2.2 RISK MANAGEMENT

Software Risk Management is a proactive approach for minimizing the uncertainty and potential loss associated with a project. Some categories of risk include product size, business impact, customer-related, process, technology, development environment, staffing (size and experience), schedule, and cost. Risk Management is a practice with processes, methods, and tools for managing risks in a project.

- Assess continuously what could go wrong (risks)

- Determine which risks are important to deal with
- Implement strategies to deal with those risks

2.2.1 Risk Identification

Risk identification is a systematic attempt to specify threats to the project plan. By identifying known and predictable risks, we can take a first step toward avoiding them when possible and controlling them when necessary. To perform the risk identification, we categorized the risk into different categories as:

- Project Risk
- Technical Risk
- Business Risk
- Known Risk
- Predictable Risk
- Unpredictable Risk

➤ Project Risk:

The Project Risk threatens the project plan. The project risks here are:

- Schedule slippage.
- Incomplete requirement specification.
- Change in user Requirements.
- Non-availability of required resources.
- Lack of communication with end user.
- Improper vision about the project.
- Staffing and organization problems.
- Non-technical customer with high technical expectations.

➤ Technical Risk:

The Technical Risk threatens the quality and timeliness of the software to be produced. If the technical risk becomes a reality, implementation may become difficult or impossible. The technical risks identified in our project are:

- Unavailable library files.
- Problem in connection to database server.
- Problem in application server.
- Problem in browser view.

➤ **Business Risk:**

The Business Risk threatens the viability of the software to be built.

- Project not delivered on time.
- Switching of database structure.

➤ **Predictable Risk:**

The Predictable risks are extrapolated from past project experience. Since we have not done any live industry project during the academic years, the predictable risks were very few. The predictable risk include mainly:

- Language error predictions.
- Lack of End user support in future project enhancement.

➤ **Unpredictable Risk:**

The Unpredictable risks are the joker in the deck. They can and do occur, but they are extremely difficult to identify in advance.

2.2.2 Risk Analysis

During risk analysis process, we considered each identified risk and made a judgment about the probability and the seriousness of it. The risk estimates are not generally precise numeric assessments but based around a number of bands. The probability of the risk might be assessed as very low (<10%), low (10-25%), moderate (25-50%), high (50-75%) or very high (>75%).

The effects of the risk are assessed as catastrophic, serious, and tolerable. We have tabulated the results of this analysis process using a table ordered according to the seriousness of the risk. Once the risks were analyzed and ranked, we assessed which were the most significant. Our

judgment was dependent on a combination of the probability of the risk arising and the effects of that risk.

| RISK | PROBABILITY | EFFECTS |
|--|-------------|--------------|
| Organizational financial problems force reductions in the project budget. | Low | Catastrophic |
| Key staffs are ill at critical times in the project. | Moderate | Serious |
| Software components which should be reused contain defects which limit their functionality. | Moderate | Serious |
| Changes to requirements which require major design rework are proposed | Moderate | Serious |
| The organization is reconstructed so that different management are responsible for the project | High | Serious |
| The database cannot process as many transactions per second as expected | Moderate | Serious |
| The time required to develop the software is underestimated. | High | Serious |
| Customers fail to understand the impact of requirements changes | Moderate | Serious |
| Required training of staff is not available | Moderate | Serious |
| The size of the software is underestimated | High | Serious |

Table 2.1 Risk Analysis

2.2.3 Risk Planning

Risk planning lists the checkpoints that are made continually to find out situation where the risk can becomes reality.

- Plan entire schedule on paper in the beginning and follow it.
- Understand the scope from external guide to have the correct design.
- Find out proper documentation, manuals and guides from the person having the required knowledge.
- Schedule should not be delayed too much.

- Take backups regularly.
- Perform thorough requirement gathering and analysis. Confirm the collected requirements with the guide.

2.3 ESTIMATION

In software development, effective software project estimation is most challenging and important activities. Without a reliable estimate, proper project planning and control is not possible. The software industry doesn't determine the project well and not using the estimate appropriately. Due to this, we have to suffer far more than we should, we need to focus on effort on improving the situation.

2.3.1 Effort Estimation

The size of the product ends up in either Line once estimated, from that effort, the estimate can be derived. The conversion of software size to total project effort can only be done if defined SDLC and development process that follow to specify, design, develop, and test the software. The software development project involves far more than simply coding software; coding is the smallest part of the effort.

2.3.2 Cost Analysis

The total cost of the project is to be considered on many factors. It includes the factors like labor, hardware and software purchases or on rent, travel for meeting, telecommunications, training courses, office space. Some of the cost is assigned to the particular project may be taken care by adding an overhead value to labor. A software development project manager is estimating only energy cost and additional project cost not considered overhead by the organization. The pure labor cost is obtained by multiplying the effort project estimate by a general work rate.

Basic Website Components and Costs

On average, the following figures can be applied to estimating the cost of a small business website (if you'd like a custom estimate for your website, call us at 310-754-3807):

- **Domain Name** – 500rs/year

- **Hosting** – 500rs to 5000rs a year (depending on traffic and hosting services)
- **Web Planning, Design and Development Time** – 1 month and up
- **Continued Website Maintenance** – 2500rs a year and up (depending on number/type of updates required)
- **Marketing Your Website Online** – 35000rs/year

Important Factors that Contribute to Website Cost

When preparing to budget web design costs, be sure to ask yourself the following questions:

- Is this a brand new site or a redesign of an existing site?
- How prepared are you to ask for bids? Have you prepared a detailed requirements document?
- Do you need a blog or a content management system (CMS)?
- Have graphics already been created for the site?
- Do you want the site to automatically resize for mobile and tablets?
- Do you need multimedia elements (Flash, video, etc.) on the site?
- How much content do you currently have and how much will need to be created?
- Do you need other special features such as social media channels, SEO (search engine optimization) or ecommerce?
- Who is going to maintain the site after it has been launched?

Below we go into these items in greater detail and provide an estimate of how much you should budget for each. The prices listed are estimates based on our 16 years of experience designing and building business websites. Prices will vary depending on your specific requirements. Be sure to contact us for an estimate.

Using the a la carte estimates above you can see how quickly elements and functionality add up. Another way to break the budget down is to assume:

- 15% Planning
- 25% Interface design
- 40% Programming
- 20% Project Management

FP & COCOMO Model

| Type of Components | Complexity of Components | | |
|--------------------------------|--------------------------|---------|---------|
| | Simple | Average | Complex |
| No of External Inputs | 3 | 4 | 6 |
| No of External Outputs | 4 | 5 | 7 |
| No of External Inquiries | 4 | 4 | 6 |
| No of Internal Logical Files | 7 | 10 | 15 |
| No of External Interface Files | 5 | 7 | 10 |

Calculation of FP:-

We have assumed $\sum f_i = 45$.

For our project, we assume

No of External Inputs=8

No of External Outputs=10

No of External Inquiries=5

No of Internal Logical Files=30

No of External Interface Files=2

For Simple Category,

$$\text{Count Total} = 8*3 + 10*4 + 5*3 + 30*7 + 2*5$$

$$\text{Count Total} = 299$$

$$FP = \text{Count Total} * [0.65 + 0.01 * (\sum f_i)]$$

$$FP = 299 * [0.65 + 0.01 * (45)]$$

$$= 328.9 \approx 329$$

CHAPTER: 3

SYSTEM REQUIREMENTS STUDY

3.1 USER CHARACTERISTICS

3.2 HARDWARE AND SOFTWARE REQUIREMENTS

3.3 CONSTRAINTS

3.1 USER CHARACTERISTICS

It describes the type of user which deals with the applications. Basically, this application has three types of users as given below:

1. Visitor
2. Student
3. Admin
4. Faculty

➤ **Visitor :**

Visitor visits web application. Visitor gets basic information regarding a company. Visitor checks demo lecture, demo material, faculty list. Visitor gets an appointment. Visitor gets himself/herself registered into the system in order to interact with it.

➤ **Student :**

Student gets himself/herself login to system and can manage his/her profile. Attendance is given and he/she can ask for leave as well. Fees schedule is shown to them. Video, material is also downloaded by them. Newly added task is displayed to all students. Task submission is done by them and grade is also given by respective faculty. Notification to pay fees is shown to them. After completion of the course or training certificate is given and after completing all the activities logging out is done.

➤ **Admin :**

Admin login to system. Admin manages individual batch and corresponding timetable for each batch. Admin manages syllabus of each courses and training. Admin handles inquiries submitted by various user and give them an appointment. Admin manages certificate and fees receipts.

➤ **Faculty :**

Faculty login to system. Faculty can manage their profile. Syllabus of each courses and training is managed by them. Submitted task of students are reviewed by them and grades are also given. Material to make available is uploaded so that student can download and watch videos of their lectures. Query submitted is solved by faculty.

3.2 HARDWARE AND SOFTWARE REQUIREMENTS

Hardware & Software Specification required for running the system smoothly.

Minimum Hardware Requirement

| | |
|-----------|----------------|
| RAM | 1 GB |
| Processor | Intel Core i-3 |
| HDD | 300 Gigabyte |

Minimum Software Requirement

| | |
|----------------------|-------------------------------|
| Operating System | Windows 7/XP/2000/Vista/More |
| Database | MySQL / MongoDB |
| JDK | 8.0 |
| Application | NetBeans/Eclipse IDE |
| Web Server | Apache Tomcat |
| Document Tool | Microsoft Word 2007 |
| Browser | Google Chrome/Mozilla Firefox |
| Document Reader Tool | Adobe PDF Reader |

3.3 CONSTRAINTS

There are some constraints defined for the system. The topics are described as below.

3.3.1 REGULATORY POLICIES

Project must be completed within 4th semester. The system design must be user friendly. All the information about user is well managed.

3.3.2 HARDWARE LIMITATION

This topic includes hardware requirements for installing the XAMPP. The computer on which you install these software's should meet the following system requirements, Intel core processor or above. 100 GB HD and 2GB RAM.

3.3.3 INTERFACE TO OTHER APPLICATION

Basically this application will be independent will be capable to operate alone with provided that inputs are proper. This application will be made such that it will be compatible for use with other application.

3.3.4 PARALLAEL OPERATIONS

The site can be used by administrator as well as user at the same time. An Administrator will system at the same time.

3.3.5 HIGHER ORDER LANGUAGE REQUIREMENTS

Since the application has been created in XAMPP it gets easily dissolved with any operating system.

3.3.6 RELIABILITY REQUIREMENTS

The main reliability requirement is the validation used. Without proper validation the system does not allow to enter that value into the database. For e.g.: In the email id, the user cannot enter any dummy value, the validation checks that whether there is '@' or '.' symbol in that. Also any null value is not allowed in place of compulsory fields, string cannot be allowed into numeric field ex: phone number.

3.3.7 CRITICALITY OF APPLICATION

If the system is of lower configuration and the user operates on larger Forms or database then there may be chances of memory overflow. This problem is mainly related with operating system memory management but we can neglect it for regular operation of software.

3.3.8 SAFETY AND SECURITY CONSIDERATION

The system is secured with tremendous security. Used Good encryption like md5.

3.3.9 ASSUMPTIONS AND DEPENDENCY

For the existence of this system we need to clarify some of the assumptions and dependency.

➤ Assumptions

Each and every user must be registered to the system and he has a valid username and password to use this system. If user is not registered, he/she will not be able to use the system. User should have the basic knowledge of internet.

➤ **Dependencies**

This system is depended user upon the type of. According to the type and permissions assigned to the user, he can access the functionalities of the system. This system is depended upon the user's valid authentication. If user inputs wrong password or username then user will not able to perform any operations of the system.

CHAPTER: 4

SYSTEM ANALYSIS

4.1 STUDY OF CURRENT SYSTEM

4.2 PROBLEM AND WEAKNESSES OF CURRENT SYSTEM

4.3 REQUIREMENTS OF NEW SYSTEM

4.4 FEASIBILITY STUDY

4.5 REQUIREMENTS VALIDATION

4.6 FUNCTIONS OF SYSTEM

4.7 DATA MODELING

4.7.1 Class Diagram / E-R Diagrams

4.7.2 System Activity or Object Integration Diagram

4.7.3 Data Dictionary

4.8 FUNCTIONAL AND BEHAVIORAL MODELING

4.8.1 Context Diagram

4.8.2 Data Flow Diagram (0 and 1 Level)

4.9 MAIN MODULES OF NEW SYSTEM

**4.10 SELECTION OF HARDWARE AND SOFTWARE AND
JUSTIFICATION**

4.1 STUDY OF CURRENT SYSTEM

- Current system is a static website.
- It doesn't provide interaction between user and company.
- User can visit a company website and take information regarding company, its developments and training that is provided. User can only contact with the company.

4.2 PROBLEM AND WEAKNESSES OF CURRENT SYSTEM

- It doesn't provide interaction between user and company.
- Only static information is provided about company and training that is offered.
- There is communication gap between users, students and company, faculty.
- There is not much facility provided user such as downloading video, material of lectures. Task submission is done manually and also time consuming.
- Attendance is taken manually again time consuming.
- Student's history of completing task is not recorded.
- There is not any online editor available which helps to build program, compile it and run on it.
- Batch is not managed properly sometimes which leads clashing of lecture.
- Appointment is not taken and also lots of paper work is involved such as at a registration process you have to bring proof of residence, you have to carry NOC letter manually and also collecting certificate of completion training.

4.3 REQUIREMENTS OF NEW SYSTEM

Functional Requirements:

- Develop a website that is dynamic.
- Features and functionalities that is not present in past system will be developed and provided to give user wonderful experience to use the new system.
- User will be registering with a proof of residence to get involved into the system. After completing registration process, Login will be done. With attachment of proof of residence and after verification from respective member, a lot of paper work and amount

of time is saved by user and user is set to free from such a tedious task such as submission of a proof of residence.

- User who wishes to have information of training or courses provided and also basic question-answers will be shown to user. And also if he wishes to visit company can directly ask for an appointment for further inquiry. User will be made available to have a glance at demo lecture and material also. User can go through a list of faculty members having description.
- Students who finds attendance process as boring will be provided to give attendance with face recognition. Student doesn't have to go to faculty member manually to ask for a leave. He can request for the leave on system and if he is granted that permission he will be allowed to remain absent. Online payment for fees will also be there. Downloading content of the lecture for a particular batch will be uploaded so that student doesn't need to ask for content of a lecture every day. Submission of task will be done online by editing program online with the compiler provided by a system. Notification will be sent to user as a reminder to pay fees.
- Faculty will manage syllabus, material, task, and batch. Faculty will solve doubts of student online via chat. Admin will track student's history and manage certificates and receipts.

Non Functional Requirements:

- **Usability:** The interface should use terms and concepts, which are drawn from the experience of people who will make most of the system.
- **Efficiency:** The portal must provide easy and fast access without consuming more cost.
- **Readability:** User should never be surprised by the behaviour of the system and it should also provide meaningful feedback when error occurs so that user can recover from the error.
- **Accuracy:** The user should require that data are obtained from database and stored in database must be accurate.
- **Security:** The user wants the data stored in database must be secured and cannot be accessed by unauthorized user.

- **Maintainability:** User wants that the system should be maintained easily means that if there are some changes required in the system that can be done easily

4.4 FEASIBILITY STUDY

- Feasibility is the measure of how beneficial the development of information system will be to an organization.
- The feasibility analysis is categorized under four different types.
 1. Operational Feasibility
 2. Technical Feasibility
 3. Schedule Feasibility
 4. Economic Feasibility

1. Operational Feasibility:

- The System is to be developed for any user who wants to use it. We want our system user friendly and easy to use.
- The administrator also may be non-technical, so the user interface will be designed in such a way that it gets comfortable for non-technical person to operate easily.

2. Technical Feasibility:

- It is a partially measurement of specific technical solution and the availability of technical resorts and expertise.
- The analyst must find out whether the current technical resources, which are available in the system is capable of handling the job.
- If not, then the analyst with the help of developer should confirm whether the technology is available and capable or not.

➤ Better Considering:

- Here we have to consider those tools which are required for developing the project.
- As far as basic knowledge concerned we have studied basic of objective-C and SQL.

3. Schedule Feasibility:

- Schedule feasibility corresponds to whether sufficient time is available to complete the project.
- Factor considered:
 - Schedule of the project
 - Time by which project has to be completed
 - Reporting period
- 4. Economic feasibility:**
 - Economic feasibility is a measure of cost effectiveness of a project or solution,
 - For declaring that the system is economically feasible, the benefits from the project should exceed or at least to the equal to the cost of development.

4.5 REQUIREMENTS VALIDATION

- Requirement validation examines this specification to ensure that all the system requirements have been stated unambiguously.
- These inconsistent, error have been detected and corrected and the work products confirmed to the standard.
- Source of the requirement are identified, final Statement of requirement has been examined by original source.
- Requirements related to main requirements are founds.
- Requirements are clarifying stated and are not misinterpreted.
- All sources of requirements are covered to get a maximum requirement.
- All method of finding requirements is applied.

4.6 FUNCTIONS OF SYSTEM

4.6.1 USE CASE

Use case diagrams used to describe a set of actions (use cases) that some system or systems (subject) should or can perform in collaboration with one or more external users of the system (actors). Each use case should provide some observable and valuable result to the actors or other stakeholders of the system. Followings are the components of use case diagram:

- **Use Case:** A Use Case describes a sequence of actions that provide something of measurable value to the actor and is drawn as horizontal ellipse.
- **Actors:** An Actor is a Person, Organization, or external system that plays a role in one or more interactions with your systems. Actors are drawn as stick figures.
- **Associations:** Association between actors and use cases are indicated in use case diagrams using solid lines. An association exists whenever an actor is involved with an interaction described by the use case.

Symbols

| | |
|--|---|
| | Actors |
| | Use Cases |
| | Association |
| | <<Include>>/<<Extend>> |

Table 4.1 Use Case Symbols



Figure 4.1 Use Case – Visitor



Figure 4.2 Use Case – Student

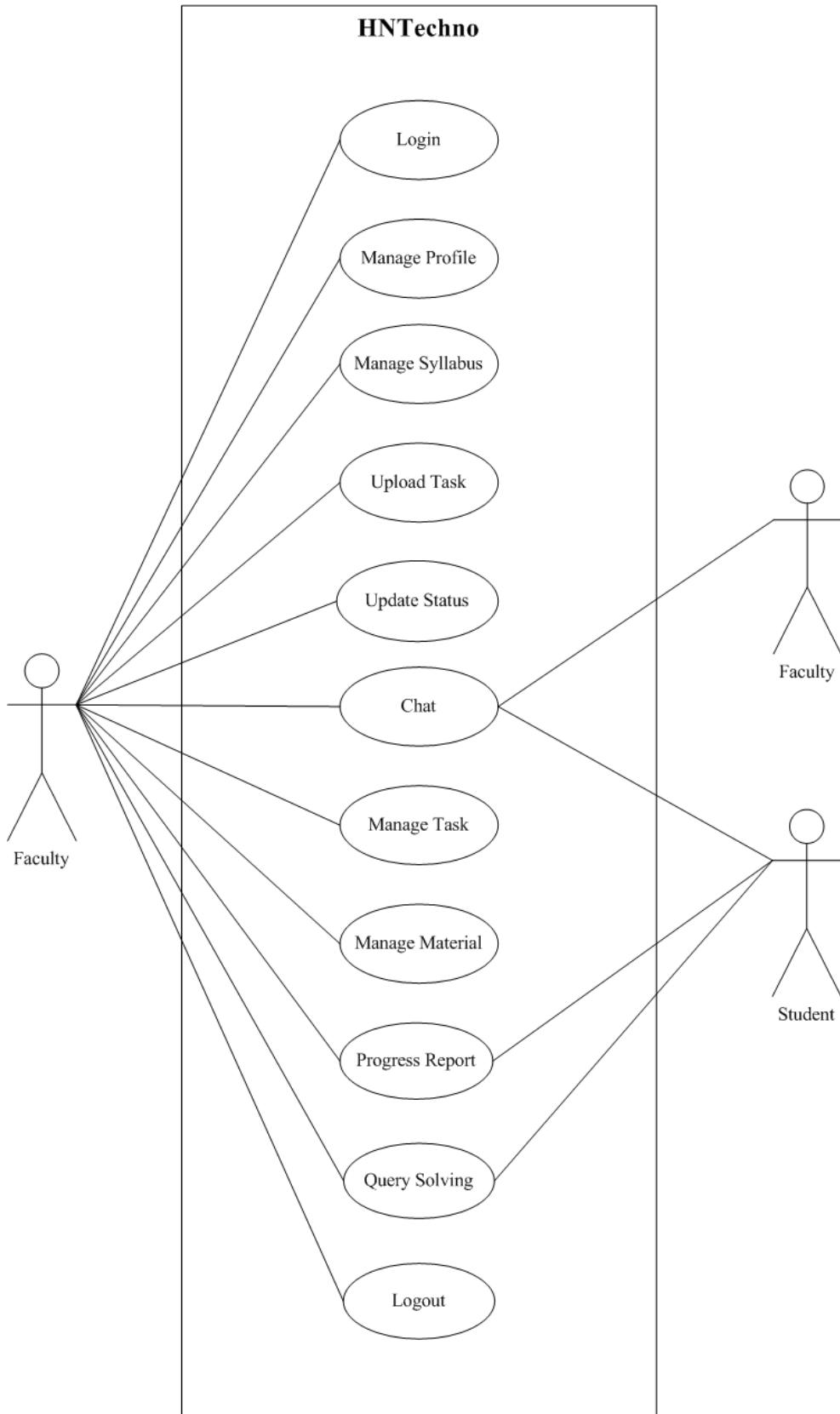


Figure 4.3 Use Case – Faculty

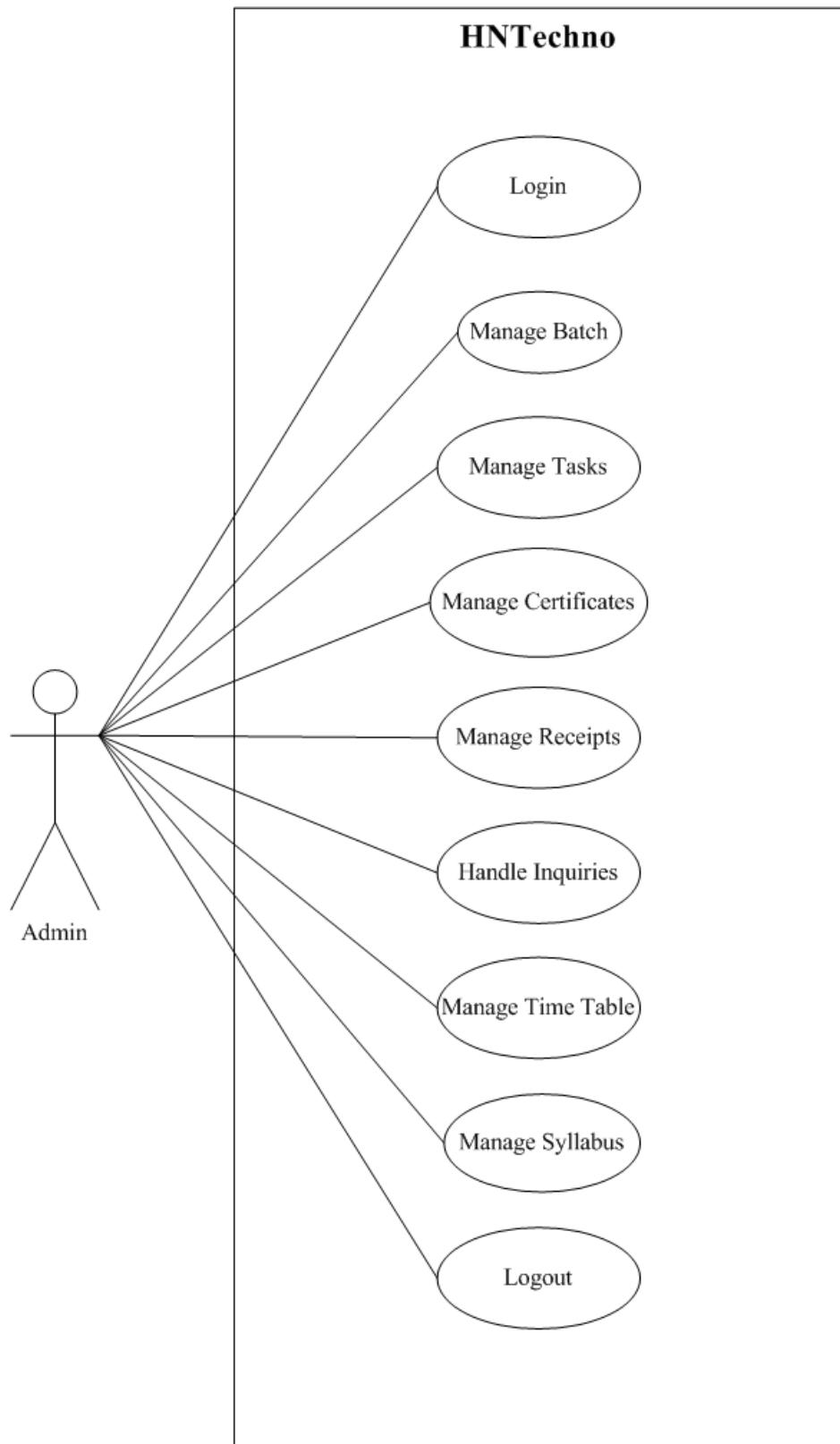


Figure 4.4 Use Case – Admin

4.7 DATA MODELING

4.7.1 E-R DIAGRAMS

An entity-relationship diagram is a data modeling technique that creates a graphical representation of the entities, and the relationships between entities, within an information system. (View diagram.)

An Entity Relationship Diagram (ERD) is a **snapshot of data structures**. ERDs show entities in a database and relationships between tables within that database. It is essential to have one of these if you want to create a good database design. The patterns help focus on how the database actually works with all of the interactions and data flows, although another useful tool is a Data Flow Diagram (DFD) which more directly describes this.

The **entity** is a person, object, place or event for which data is collected. For example, if you consider the information system for a business, entities would include not only customers, but the customer's address, and orders as well. The entity is represented by a rectangle and labeled with a singular noun.

The steps involved in creating an ERD are:

- Identify the entities.
 - Determine all significant interactions.
 - Analyze the nature of the interactions.
 - Draw the ERD.
- Also called an **entity-relationship model**. A graphical representation of entities and their relationships to each other, typically used in computing in regard to the organization of data within databases or information systems. An entity is a piece of data-an object or concept about which data is stored. A relationship is how the data is shared between entities.
- There are three types of relationships between entities:
- **One-to-one**: one instance of an entity (A) is associated with one other instance of another entity (B). For example, in a database of employees, each employee name (A) is associated with only one social security number (B).

- **One-to-many:** one instance of an entity (A) is associated with zero, one or many instances of another entity (B), but for one instance of entity B there is only one instance of entity A. For example, for a company with all employees working in one building, the building name (A) is associated with many different employees (B), but those employees all share the same singular association with entity A.
- **Many-to-many:** one instance of an entity (A) is associated with one, zero or many instances of another entity (B), and one instance of entity B is associated with one, zero or many instances of entity A. For example, for a company in which all of its employees work on multiple projects, each instance of an employee (A) is associated with many instances of a project (B), and at the same time, each instance of a project (B) has multiple employees (A) associated with it.

Symbols

| | |
|--|------------------------------|
| | Attribute |
| | Condition |
| | Entity |
| | Multi value Attribute |
| | Derived Attribute |
| | Relationship |
| | One to one |
| | One to many |

Table 4.2 E-R Diagram Symbols

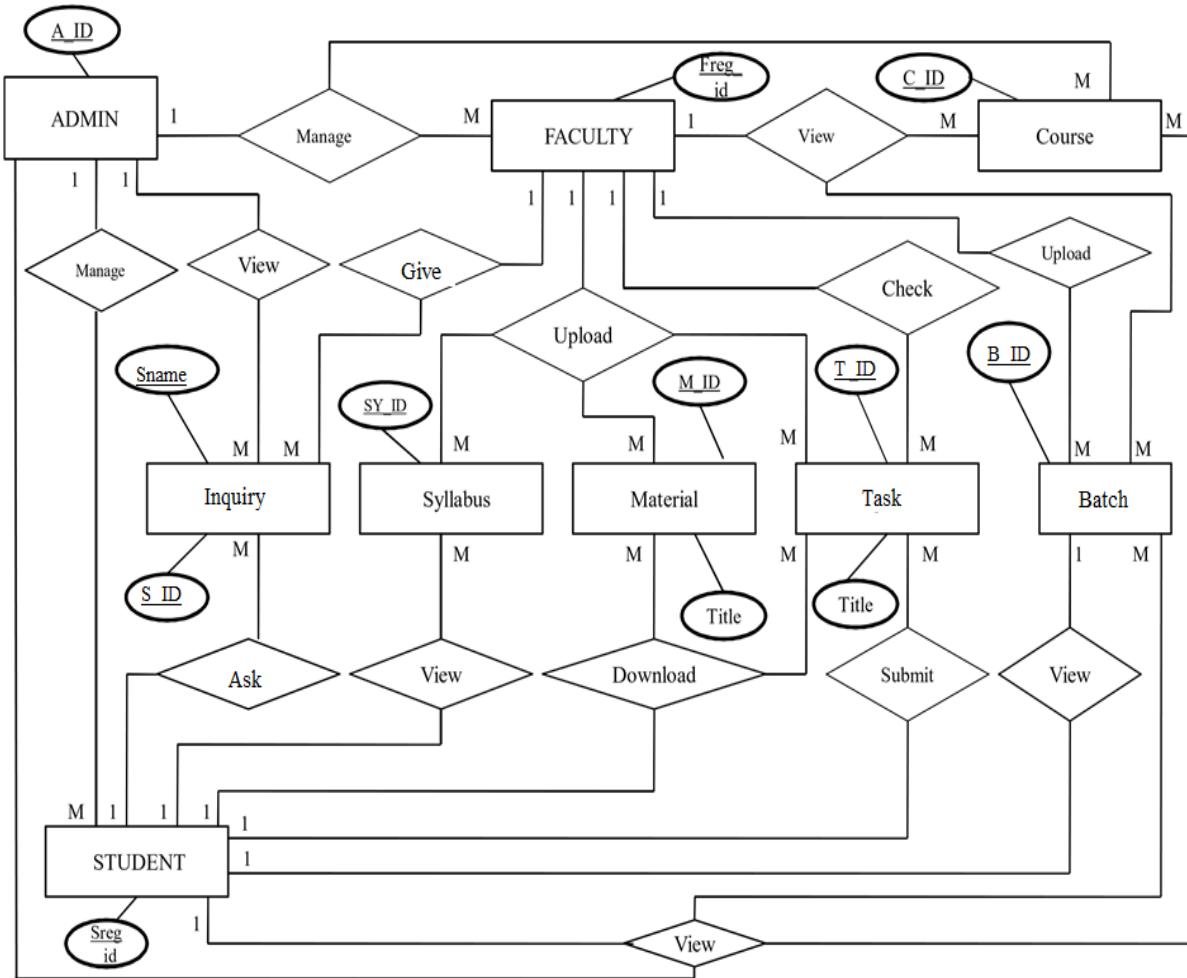


Figure 4.5 E-R Diagram

4.7.2 SYSTEM ACTIVITY

Activity diagrams are the object-oriented equivalent of flow charts and data-flow diagrams from structured development. It describes the workflow behavior of a system. The process flows in the system are captured in the activity diagram. Activity diagram illustrates the dynamic nature of a system by modeling the flow of control from Activity to activity

Symbols

| | |
|--|--|
| | Start |
| | Condition |
| | Process |
| | End |
| | A Fork Node: Is used to split behaviors into a set of parallel or concurrent flows of activities(or actions) |
| | A Join Node: Is used to bring back together a set of parallel or concurrent flows of activities(or actions) |

Table 4.3 Activity Diagram Symbols

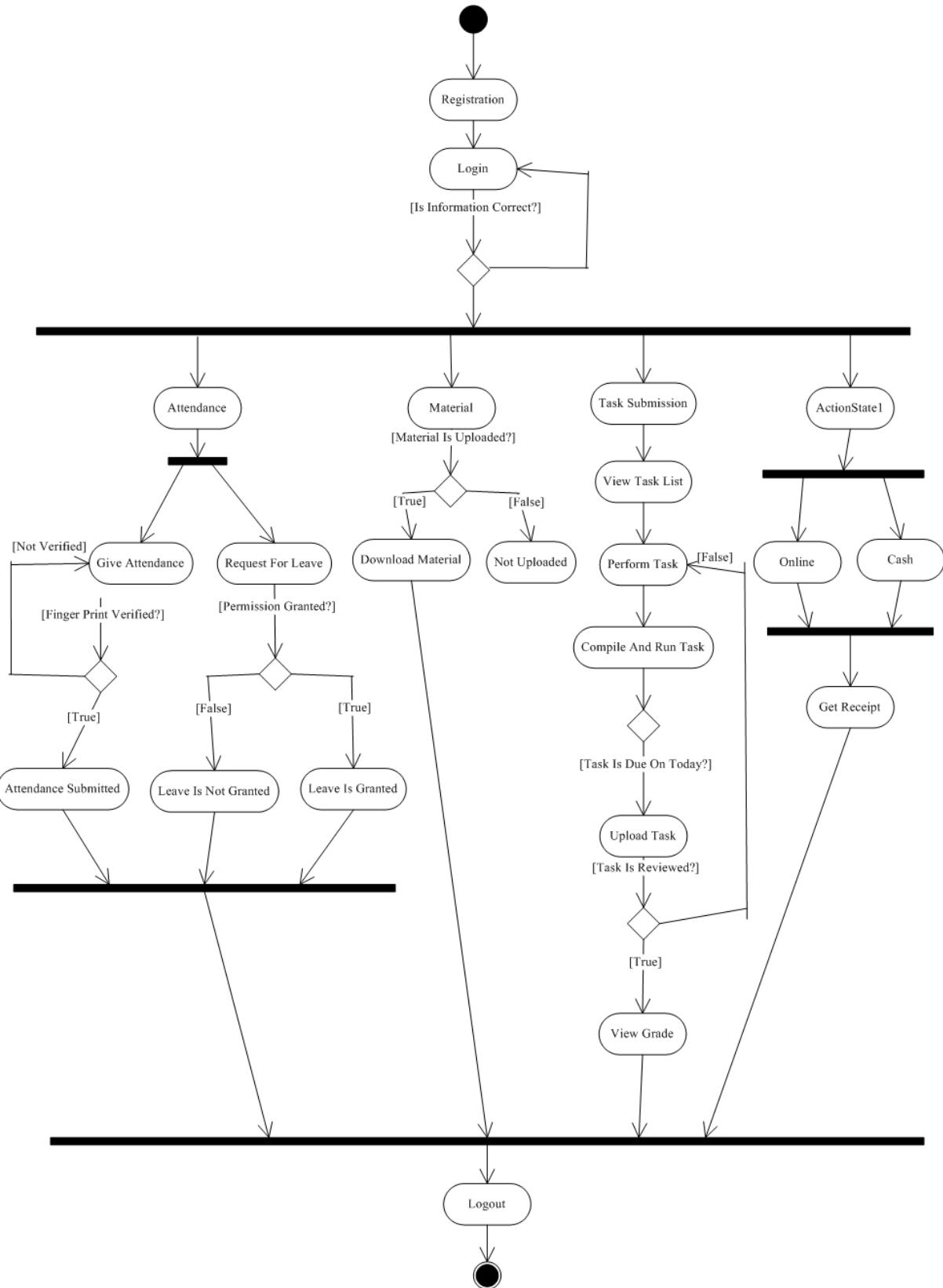


Figure 4.6 Activity Diagram

4.7.3 DATA DICTIONARY

1. Inquiry

| Name | Data Type | Constraint | Description |
|----------------|--------------|---------------|--|
| inq_id | bigint(11) | PK | Store the inquiry id |
| first_name | varchar(20) | - | Store the first name |
| middle_name | varchar(20) | - | Store the middle name |
| last_name | varchar(20) | - | Store the last name |
| Contact | int(10) | - | Store the contact of student |
| Email | varchar(50) | - | Store the email id of student |
| field_id | bigint(11) | FK (field) | Store the field |
| lan_id | bigint(11) | FK (language) | Store the language |
| clg_id | bigint(11) | FK (college) | Store the college |
| state_id | bigint(11) | FK (state) | Store the state |
| city_id | bigint(11) | FK (city) | Store the city |
| Reference | varchar(20) | - | Store the reference name |
| Remarks | varchar(300) | - | Store the remarks |
| Status | varchar(10) | - | Pending/Rejected/Confirm |
| create_by_id | bigint(11) | - | Store the id of user who fills the form |
| create_by_date | Timestamp | - | Store the date when form is filled |
| update_by_id | bigint(11) | - | Store the id of user who update the form |
| update_by_date | Datetime | - | Store the date when form is updated |
| delete_by_id | bigint(11) | - | Store the id of user who delete the form |
| delete_by_date | Datetime | - | Store the date when form is deleted |
| mac_id | varchar(20) | - | Store the mac address |
| active_flag | tinyint(1) | - | Block/unblock the data |

Table 4.4 Inquiry

2. Registration

| Name | Data Type | Constraint | Description |
|----------------|--------------|------------------|--|
| reg_id | bigint(11) | PK | Store the Registration Id |
| reg_no | varchar(10) | FK(registration) | Store the Registration Number |
| reg_fnm | varchar(15) | - | Store the Registration First Name |
| reg_mnm | varchar(15) | - | Store the Registration Middle Name |
| reg_lnm | varchar(15) | - | Store the Registration Las Name |
| reg_email | varchar(30) | - | Store the Registration Email |
| gender | varchar(6) | - | Store the gender |
| state_id | bigint(11) | FK (state) | Store the State Id |
| city_id | bigint(11) | FK (city) | Store the City Id |
| reg_address | varchar(100) | | Store the Registration Address |
| reg_username | varchar(15) | | Store the Registration User name |
| reg_password | varchar(16) | | Registration Password Store the |
| field_id | bigint(11) | FK (field) | Store the Field id |
| reg_remarks | varchar(300) | | Store the Ragistration remarks |
| create_by_id | bigint(11) | | Store the id of user who fills the form |
| create_by_date | Timestamp | | Store the date when form is filled |
| update_by_id | bigint(11) | | Store the id of user who update the form |
| update_by_date | Datetime | | Store the date when form is updated |
| delete_by_id | bigint(11) | | Store the id of user who delete the form |
| delete_by_date | datetime | - | Store the date when form is deleted |
| mac_id | varchar(20) | - | Store the mac address |
| active_flag | int(1) | - | Block/unblock the data |

Table 4.5 Registration

3. User

| Name | Data Type | Constraint | Description |
|--------------------|--------------|-----------------|------------------------------|
| user_id | bigint(11) | PK | Store the user id |
| user_fnm | varchar(20) | - | name Store the user first |
| user_mnm | varchar(20) | - | store the user middle name |
| user_lnm | varchar(20) | - | Store the user last name |
| user_contact | int(10) | - | store the user contact |
| user_email | varchar(30) | - | store the email |
| user_state_id | bigint(11) | - | store the user state id |
| user_city_id | bigint(20) | - | store the user city id |
| Address | Text | - | store the address |
| Gender | varchar(6) | - | store the Gender |
| lan_id | bigint(11) | FK (language) | store the language id |
| course_id | bigint(11) | FK (course) | store the course id |
| Concept | varchar(50) | - | store the concept |
| Duration | varchar(20) | - | store the duration |
| total_fees | varchar(10) | - | store the total fees |
| Username | varchar(20) | - | store the user name |
| Password | varchar(10) | - | store the password |
| con_pass | varchar(10) | - | store the confirm password |
| filed_id | bigint(11) | FK (field) | store the field id |
| clg_id | bigint(11) | FK (college) | store the clg id |
| uni_id | bigint(11) | FK (university) | Store the university id |
| city_id | bigint(11) | FK (city) | Store the city id |
| state_id | bigint(11) | FK (state) | Store the state id |
| project_defination | varchar(50) | - | Store the project Defination |
| time_period | time(6) | - | Store the timne period |
| start_date | Date | - | Store the starting date |
| end_date | Date | - | store the ending date |
| Noc | tinyint(1) | - | Store the Noc |
| Remarks | varchar(300) | - | Remarks Store the |

| | | | |
|----------------|----------------|---|--|
| create_by_id | bigint(11) | - | Store the id of user who fills the form |
| create_by_date | Datetime | - | Store the date when form is filled |
| update_by_id | update_by_date | - | Store the id of user who update the form |
| update_by_date | Datetime | - | Store the date when form is updated |
| delete_by_id | bigint(11) | - | Store the id of user who delete the form |
| delete_by_date | Datetime | - | Store the date when form is deleted |
| mac_id | varchar(20) | - | Store the mac address |
| active_flag | tinyint(1) | - | Block/unblock the data |

Table 4.6 User

4. Appointment

| Name | Data Type | Constraint | Description |
|----------------|--------------|--------------|--|
| app_id | bigint(11) | PK | Store an appointment id |
| lan_id | bigint(11) | FK(language) | Store a language id |
| inq_id | bigint(11) | FK(inquiry) | Store an inquiry id |
| app_owner | varchar(20) | - | Store owner name |
| app_date | Date | - | Store appointment date |
| app_time | Time | - | Store appointment time |
| Remarks | varchar(300) | - | Store remarks |
| Status | varchar(10) | - | Pending/Rejected/Confirm |
| create_by_id | bigint(11) | - | Store the id of user who fills the form |
| create_by_date | Timestamp | - | Store the date when form is filled |
| update_by_id | bigint(11) | - | Store the id of user who update the form |
| pdate_by_date | Datetime | - | Store the date when form is updated |
| delete_by_id | bigint(11) | - | Store the id of user who delete the form |
| delete_by_date | Datetime | - | Store the date when form is deleted |
| mac_id | varchar(20) | - | Store the mac address |

| | | | |
|-------------|--------|---|------------------------|
| active_flag | int(1) | - | Block/unblock the data |
|-------------|--------|---|------------------------|

Table 4.7 Appointment

5. Demo

| Name | Data Type | Constraint | Description |
|----------------|--------------|--------------|--|
| demo_id | bigint(11) | PK | Store the demo id |
| lan_id | bigint(20) | FK(language) | Store the language id |
| course_id | bigint(11) | FK (course) | Store the course id |
| fac_id | bigint(20) | FK (faculty) | Store the faculty id |
| demo_date | Date | - | Store the date of demo |
| demo_time | Time | - | Store the time of demo |
| Remarks | varchar(300) | - | Store the remarks |
| Status | varchar(10) | - | Pending/Rejected/Confirm |
| create_by_id | bigint(11) | - | Store the id of user who fills the form |
| create_by_date | Timestamp | - | Store the date when form is filled |
| update_by_id | bigint(11) | - | Store the id of user who update the form |
| update_by_date | Datetime | - | Store the date when form is updated |
| delete_by_id | bigint(11) | - | Store the id of user who delete the form |
| delete_by_date | Datetime | - | Store the date when form is deleted |
| mac_id | varchar(20) | - | Store the mac address |
| active_flag | tinyint(1) | - | Block/unblock the data |
| inq_id | bigint(11) | - | Store the inquiry id |

Table 4.8 Demo

6. Batch

| Name | Data Type | Constraint | Description |
|----------------|--------------|------------|--|
| batch_id | bigint(11) | PK | Store a batch id |
| course_id | int(10) | FK(course) | Store a course id |
| batch_name | varchar(20) | - | Store a batch name |
| Remarks | varchar(300) | - | Store remarks |
| create_by_id | bigint(11) | - | Store the id of user who fills the form |
| create_by_date | Datetime | - | Store the date when form is filled |
| update_by_id | bigint(11) | - | Store the id of user who update the form |
| update_by_date | Datetime | - | Store the date when form is updated |
| delete_by_id | bigint(11) | - | Store the id of user who delete the form |
| delete_by_date | Datetime | - | Store the date when form is deleted |
| mac_id | varchar(20) | - | Store the mac address |
| active_flag | tinyint(1) | - | Pending/Rejected/Confirm |

Table 4.9 Batch

7. Course

| Name | Data Type | Constraint | Description |
|----------------|--------------|--------------|--|
| course_id | bigint(11) | PK | Store the course id |
| lan_id | bigint(10) | FK(language) | Store the language id |
| Course | varchar(20) | - | Store the course |
| Fees | varchar(20) | - | Store the fees of course |
| Duration | varchar(20) | - | Store the duration of course |
| Description | varchar(100) | - | Store the remarks |
| create_by_id | bigint(11) | - | Store the id of user who fills the form |
| create_by_date | Timestamp | - | Store the date when form is filled |
| update_by_id | bigint(11) | - | Store the id of user who update the form |

| | | | |
|----------------|-------------|---|--|
| update_by_date | Datetime | - | Store the date when form is updated |
| delete_by_id | bigint(11) | - | Store the id of user who delete the form |
| delete_by_date | Datetime | - | Store the date when form is deleted |
| mac_id | varchar(20) | - | Store the mac address |
| active_flag | tinyint(1) | - | Block/unblock the data |

Table 4.10 Course

8. Sub Course

| Name | Data Type | Constraint | Description |
|----------------|--------------|--------------|--|
| subcourse_id | int(11) | PK | Store the sub-course id |
| lan_id | bigint(11) | FK(language) | Store the language id |
| course_id | bigint(11) | FK(course) | Store the course id |
| sub_course | varchar(20) | - | Store the sub-course |
| Description | varchar(500) | - | Store the description |
| create_by_id | bigint(11) | - | Store the id of user who fills the form |
| create_by_date | Timestamp | - | Store the date when form is filled |
| update_by_id | bigint(11) | - | Store the id of user who update the form |
| update_by_date | Datetime | - | Store the date when form is updated |
| delete_by_id | bigint(11) | - | Store the id of user who delete the form |
| delete_by_date | Datetime | - | Store the date when form is deleted |
| mac_id | varchar(20) | - | Store the mac address |
| active_flag | tinyint(1) | - | Block/unblock the data |

Table 4.11 sub course

9. College

| Name | Data Type | Constraint | Description |
|----------------|--------------|------------|--|
| clg_id | bigint(11) | PK | Store a college id |
| uni_id | bigint(11) | FK | Store a university id |
| clg_name | varchar(100) | - | Store a college name |
| clg_address | varchar(300) | - | Store a college address |
| clg_contact | bigint(10) | - | Store a college contact |
| clg_email | varchar(30) | - | Store the email id of college |
| create_by_id | bigint(15) | - | Store the id of user who fills the form |
| create_by_date | Datetime | - | Store the date when form is filled |
| update_by_id | bigint(15) | - | Store the id of user who update the form |
| update_by_date | Datetime | - | Store the date when form is updated |
| delete_by_id | bigint(15) | - | Store the id of user who delete the form |
| delete_by_date | Datetime | - | Store the date when form is deleted |
| mac_id | varchar(20) | - | Store the mac address |
| active_flag | int(1) | - | Block/unblock the data |

Table 4.12 College

10. State

| Name | Data Type | Constraint | Description |
|----------------|-------------|------------|---|
| state_id | bigint(11) | PK | Store the state id |
| state_name | varchar(20) | - | Store the state |
| create_by_id | bigint(11) | - | Store the id of user who fills the form |
| create_by_date | Timestamp | - | Store the date when form is filled |

| | | | |
|----------------|-------------|---|--|
| update_by_id | bigint(11) | - | Store the id of user who update the form |
| update_by_date | Datetime | - | Store the date when form is updated |
| delete_by_id | bigint(11) | - | Store the id of user who delete the form |
| delete_by_date | Datetime | - | Store the date when form is deleted |
| mac_id | varchar(20) | - | Store the mac address |
| active_flag | tinyint(1) | - | Block/unblock the data |

Table 4.13 State

11. Field

| Name | Data Type | Constraint | Description |
|----------------|-------------|------------|--|
| field_id | bigint(11) | PK | Store the field id |
| field_name | varchar(10) | - | Store the field name |
| create_by_id | bigint(11) | - | Store the id of user who fills the form |
| create_by_date | Timestamp | - | Store the date when form is filled |
| update_by_id | bigint(11) | - | Store the id of user who update the form |
| update_by_date | Datetime | - | Store the date when form is updated |
| delete_by_id | bigint(11) | - | Store the id of user who delete the form |
| delete_by_date | Datetime | - | Store the date when form is deleted |
| mac_id | varchar(20) | - | Store the mac address |
| active_flag | int(1) | - | Block/unblock the data |

Table 4.14 field

12. University

| Name | Data Type | Constraint | Description |
|------|-----------|------------|-------------|
| | | | |

| | | | |
|----------------|--------------|----|--|
| uni_id | bigint(11) | PK | Store the university id |
| uni_name | varchar(30) | - | Store the university name |
| Remarks | varchar(300) | - | Store the remarks |
| create_by_id | bigint(15) | - | Store the id of user who fills the form |
| create_by_date | Timestamp | - | Store the date when form is filled |
| update_by_id | bigint(15) | - | Store the id of user who update the form |
| update_by_date | Datetime | - | Store the date when form is updated |
| delete_by_id | bigint(15) | - | Store the id of user who delete the form |
| delete_by_date | Datetime | - | Store the date when form is deleted |
| mac_id | varchar(20) | - | Store the mac address |
| active_flag | int(1) | - | Block/unblock the data |

Table 4.15 university

13. Language

| Name | Data Type | Constraint | Description |
|----------------|--------------|------------|--|
| lan_id | bigint(11) | PK | Store the language id |
| lan_name | varchar(10) | - | Store the language |
| lan_duration | varchar(10) | - | Store the duration of the full language |
| lan_fees | varchar(10) | - | Store the fess of the language |
| Description | varchar(300) | - | Store the description |
| create_by_id | bigint(15) | - | Store the id of user who fills the form |
| create_by_date | Timestamp | - | Store the date when form is filled |
| update_by_id | bigint(15) | - | Store the id of user who update the form |

| | | | |
|----------------|-------------|---|--|
| update_by_date | Datetime | - | Store the date when form is updated |
| delete_by_id | bigint(15) | - | Store the id of user who delete the form |
| delete_by_date | Datetime | - | Store the date when form is deleted |
| mac_id | varchar(20) | - | Store the mac address |
| active_flag | int(1) | - | Block/unblock the data |

Table 4.16 language

14. Create Batch

| Name | Data Type | Constraint | Description |
|-----------------|-------------|------------|--|
| cbat_id | bigint(11) | PK | Store the ID of Create Batch |
| cbat_course | varchar(20) | - | Store the course of the create batch |
| cbat_batch | varchar(20) | - | Store the batch |
| cbat_stud_list | Text | - | store the student details |
| cbat_start_dt | Date | - | store the starting date |
| cbat_end_dt | Date | - | store the ending date |
| cbat_faculty | varchar(20) | - | store the faculty |
| cbat_no_of_stud | int(5) | - | store the No. of Student |
| create_by_id | bigint(11) | - | Store the id of user who fills the form |
| create_by_date | Timestamp | - | Store the date when form is filled |
| update_by_id | bigint(11) | - | Store the id of user who update the form |
| update_by_date | Datetime | - | Store the date when form is updated |
| delete_by_id | bigint(11) | - | Store the id of user who delete the form |
| delete_by_date | Datetime | - | Store the date when form is deleted |
| mac_id | varchar(20) | - | Store the mac address |

| | | | |
|-------------|------------|---|------------------------|
| active_flag | tinyint(1) | - | Block/unblock the data |
|-------------|------------|---|------------------------|

Table 4.17 Create Batch

15. City

| Name | Data Type | Constraint | Description |
|----------------|-------------|------------|--|
| city_id | bigint(11) | PK | Store a cityid |
| state_id | int(20) | FK | Store a state id |
| create_by_id | bigint(11) | - | Store the id of user who fills the form |
| create_by_date | Datetime | - | Store the date when form is filled |
| update_by_id | bigint(11) | - | Store the id of user who update the form |
| update_by_date | Datetime | - | Store the date when form is updated |
| delete_by_id | bigint(11) | - | Store the id of user who delete the form |
| delete_by_date | Datetime | - | Store the date when form is deleted |
| mac_id | varchar(20) | - | Store the mac address |
| active_flag | tinyint(1) | - | Pending/Rejected/Confirm |

Table 4.18 City

16. Task

| Name | Data Type | Constraint | Description |
|----------------|--------------|------------|--|
| tk_id | bigint(11) | PK | Store the task id |
| batch_id | int(10) | FK (batch) | Store the batch id |
| Task | varchar(200) | - | Store the task name |
| tk_file | varchar(50) | - | Store the uploaded task |
| Remarks | varchar(300) | - | Store the remarks |
| create_by_id | bigint(11) | - | Store the id of user who fills the form |
| create_by_date | Timestamp | - | Store the date when form is filled |
| update_by_id | bigint(11) | - | Store the id of user who update the form |

| | | | |
|----------------|-------------|---|--|
| update_by_date | Datetime | - | Store the date when form is updated |
| delete_by_id | bigint(11) | - | Store the id of user who delete the form |
| delete_by_date | Datetime | - | Store the date when form is deleted |
| mac_id | varchar(20) | - | Store the mac address |
| active_flag | tinyint(1) | - | Block/unblock the data |

Table 4.19 task

17. Material

| Name | Data Type | Constraint | Description |
|----------------|--------------|----------------|--|
| mat_id | bigint(11) | PK | Store the material id |
| lan_id | bigint(11) | FK (language) | Store the language id |
| subcourse_id | bigint(11) | FK (subcourse) | Store the sub course id |
| mat_file | varchar(100) | - | Store the uploaded material |
| Remarks | varchar(300) | - | Store the remarks |
| create_by_id | bigint(11) | - | Store the id of user who fills the form |
| create_by_date | Timestamp | - | Store the date when form is filled |
| update_by_id | bigint(11) | - | Store the id of user who update the form |
| update_by_date | Datetime | - | Store the date when form is updated |
| delete_by_id | bigint(11) | - | Store the id of user who delete the form |
| delete_by_date | Datetime | - | Store the date when form is deleted |
| mac_id | varchar(20) | - | Store the mac address |
| active_flag | tinyint(1) | - | Block/unblock the data |

Table 4.20 material

18. Role

| Name | Data Type | Constraint | Description |
|------------------|--------------|------------|--|
| role_id | bigint(11) | PK | Store the role id |
| Role | varchar(20) | - | Store the role |
| role_description | varchar(300) | - | Store the description |
| create_by_id | bigint(11) | - | Store the id of user who fills the form |
| create_by_date | Timestamp | - | Store the date when form is filled |
| update_by_id | bigint(11) | - | Store the id of user who update the form |
| update_by_date | Datetime | - | Store the date when form is updated |
| delete_by_id | bigint(11) | - | Store the id of user who delete the form |
| delete_by_date | Datetime | - | Store the date when form is deleted |
| mac_id | varchar(20) | - | Store the mac address |
| active_flag | tinyint(1) | - | Block/unblock the data |

Table 4.21 role

19. Faculty

| Name | Data Type | Constraint | Description |
|--------------|--------------|------------|-------------------------|
| fac_id | bigint(11) | PK | Store the faculty id |
| role_id | bigint(11) | FK (role) | Store the role id |
| faculty_name | varchar(50) | - | Store the faculty name |
| Username | varchar(30) | - | Store the user name |
| Password | joining_dt | - | Store the password |
| joining_dt | Date | - | Store the joining date |
| leaving_dt | Date | - | Store the ending date |
| Email | varchar(50) | - | Store the email address |
| Contact | int(10) | - | Store the contact |
| Address | varchar(300) | - | Store the address |

| | | | |
|----------------|--------------|---------------|--|
| salary_type | varchar(20) | - | Store the type of salary |
| Amount | varchar(20) | - | Store the Amount |
| upload_img | varchar(500) | - | Upload the Images |
| upload_doc | varchar(500) | - | Upload the Documents |
| lan_id | bigint(11) | FK (language) | Store the Language Id |
| course_id | bigint(11) | FK (course) | Store the Course Id |
| Remarks | varchar(300) | - | Store the Remarks |
| create_by_id | bigint(11) | - | Store the id of user who fills the form |
| create_by_date | Timestamp | - | Store the date when form is filled |
| update_by_id | bigint(11) | - | Store the id of user who update the form |
| update_by_date | Datetime | - | Store the date of user who update the form |
| delete_by_id | bigint(11) | - | Store the id of user who delete the form |
| delete_by_date | Datetime | - | Store the id of user who delete the form |
| mac_id | varchar(20) | - | Store the mac address |
| Status | int(2) | - | Pending/Rejected/Confirm |
| active_flag | tinyint(1) | - | Block/unblock the data |

Table 4.22 faculty

20. Privilege

| Name | Data Type | Constraint | Description |
|-----------------------|--------------|------------|---|
| privilage_id | int(11) | PK | store the privilege id |
| privilage_name | varchar(30) | - | store the privilege name |
| privilage_description | varchar(300) | - | store the description |
| create_by_id | bigint(11) | - | Store the id of user who fills the form |
| create_by_date | timestamp | - | Store the date when form is filled |

| | | | |
|----------------|-------------|---|--|
| update_by_id | bigint(11) | - | Store the id of user who update the form |
| update_by_date | Datetime | - | Store the date when form is updated |
| delete_by_id | bigint(11) | - | Store the id of user who delete the form |
| delete_by_date | Datetime | - | Store the date when form is deleted |
| mac_id | varchar(20) | - | Store the mac address |
| active_flag | int(10) | - | Block/unblock the data |

Table 4.23 Privilege

4.8 FUNCTIONAL AND BEHAVIORAL MODELING

4.8.1 CONTEXT DIAGRAM

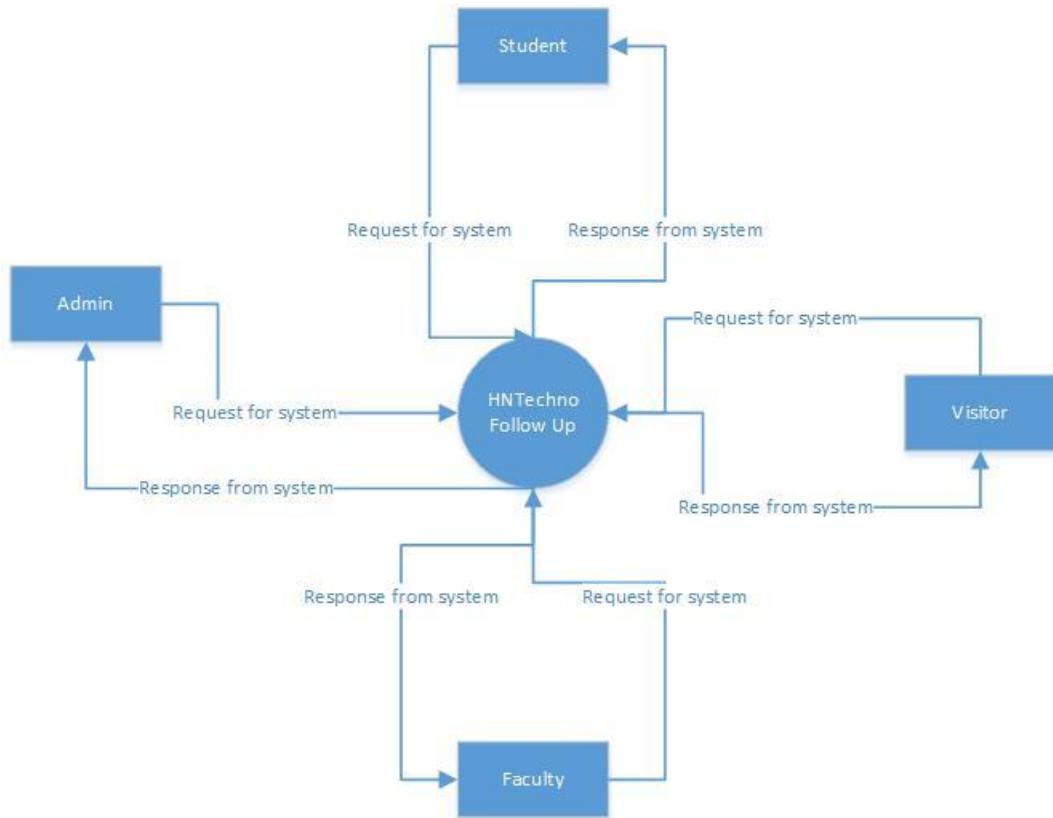


Figure 4.7 Context Diagram

4.8.2 DATA FLOW DIAGRAM (0 and 1 level)

A Data Flow Diagram models a System by using external entities from which A Data Flows to a process, Which transform the data and creates output data Flow Which goes to other processes or external entity or data store. A Stored data may also flow to process as input.

The main merit of DFD is that it can provide an overview of what data system would processes. What transactions of data are done? What data are store. Which stored data are used? Where the result flows.

The graphical representation of the system makes it good communication tool between a user and system analyst on the one hand and a system Designer on the other.

In practice, DFD are used for representing logical processing of data. It is however useful to involve a logical DFD after first developing a physical DFD which shows a person's performing various operations performing operations.

External entities are represented rectangle. They are outside the System, with which the system interacts. Entity supplying data are known as sources and those consuming data are known as sinks.

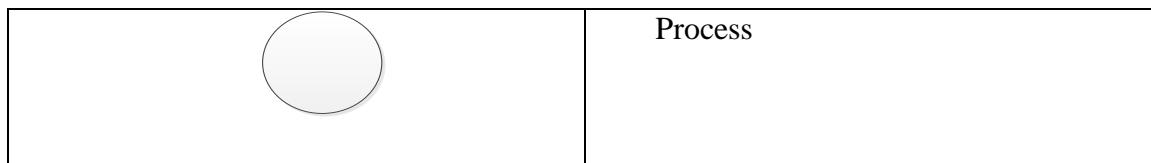
A Data stores data. Processes may store or retrieve data from a Data store. The data store is depicted by two parallel lines.

If an arrow points to a store, it indicates operation of writing in the Store. If it indicates the operation of the reading from store.

Data flow can take place between

- A data store and a process.
- A process to a data store.
- An external entity to a process.
- From process to an external entity.

Symbols:



| | | |
|--|--|----------------|
| | | Input / Output |
| | | Data Base |

Table 4.24 Data Flow Diagram Symbols

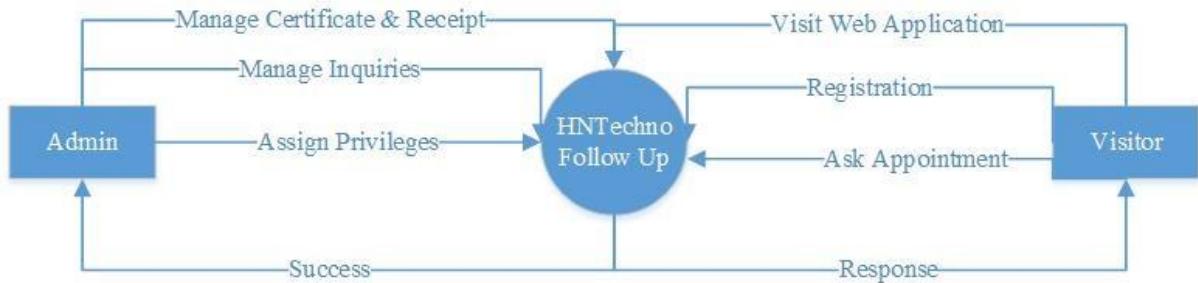


Figure 4.8 0 Level DFD – 1

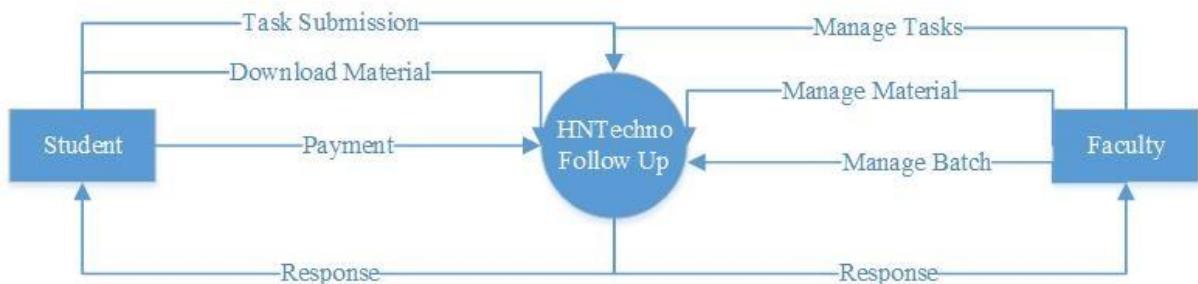


Figure 4.9 0 Level DFD – 2

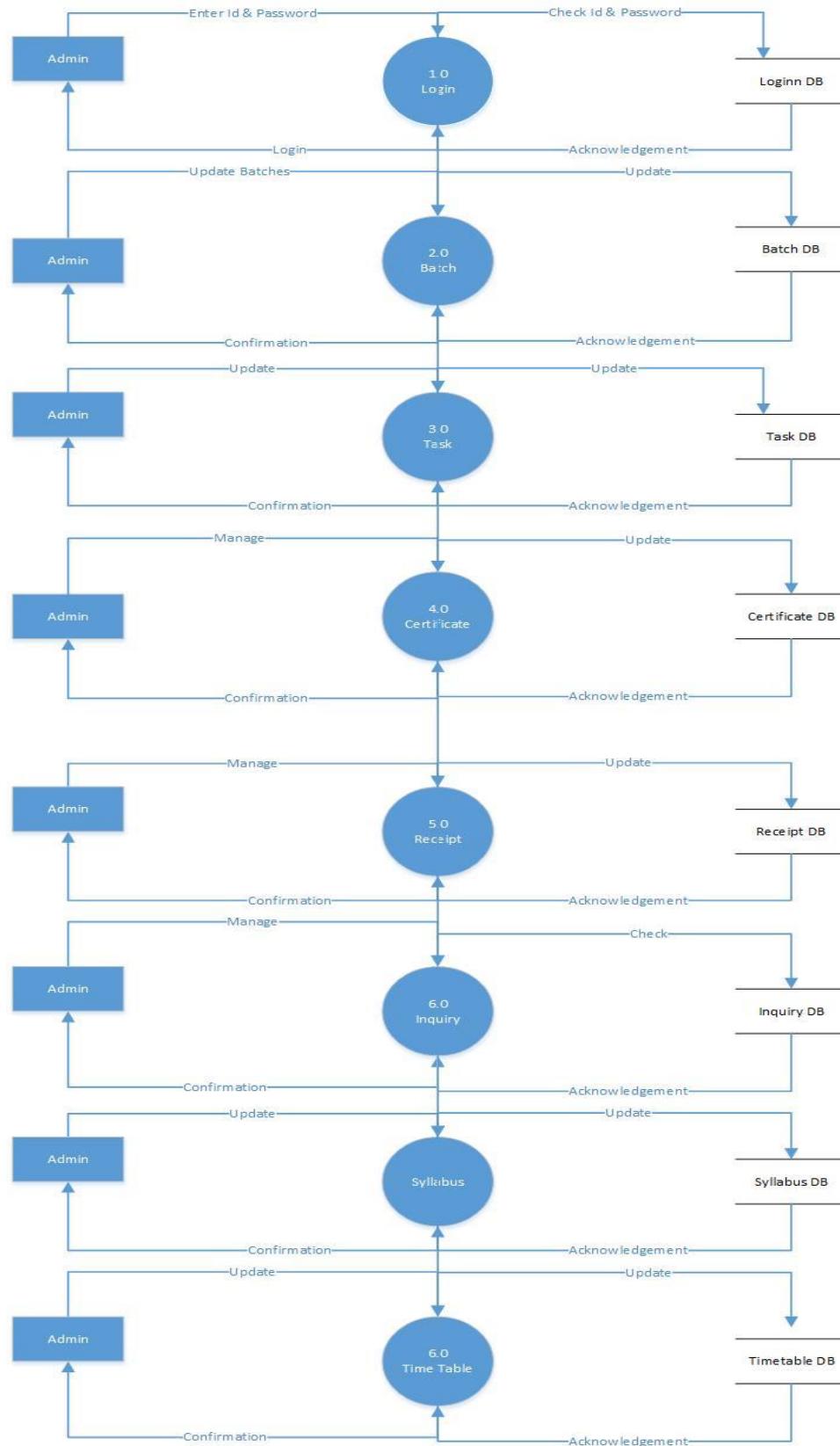


Figure 4.10 1 level DFD – Admin

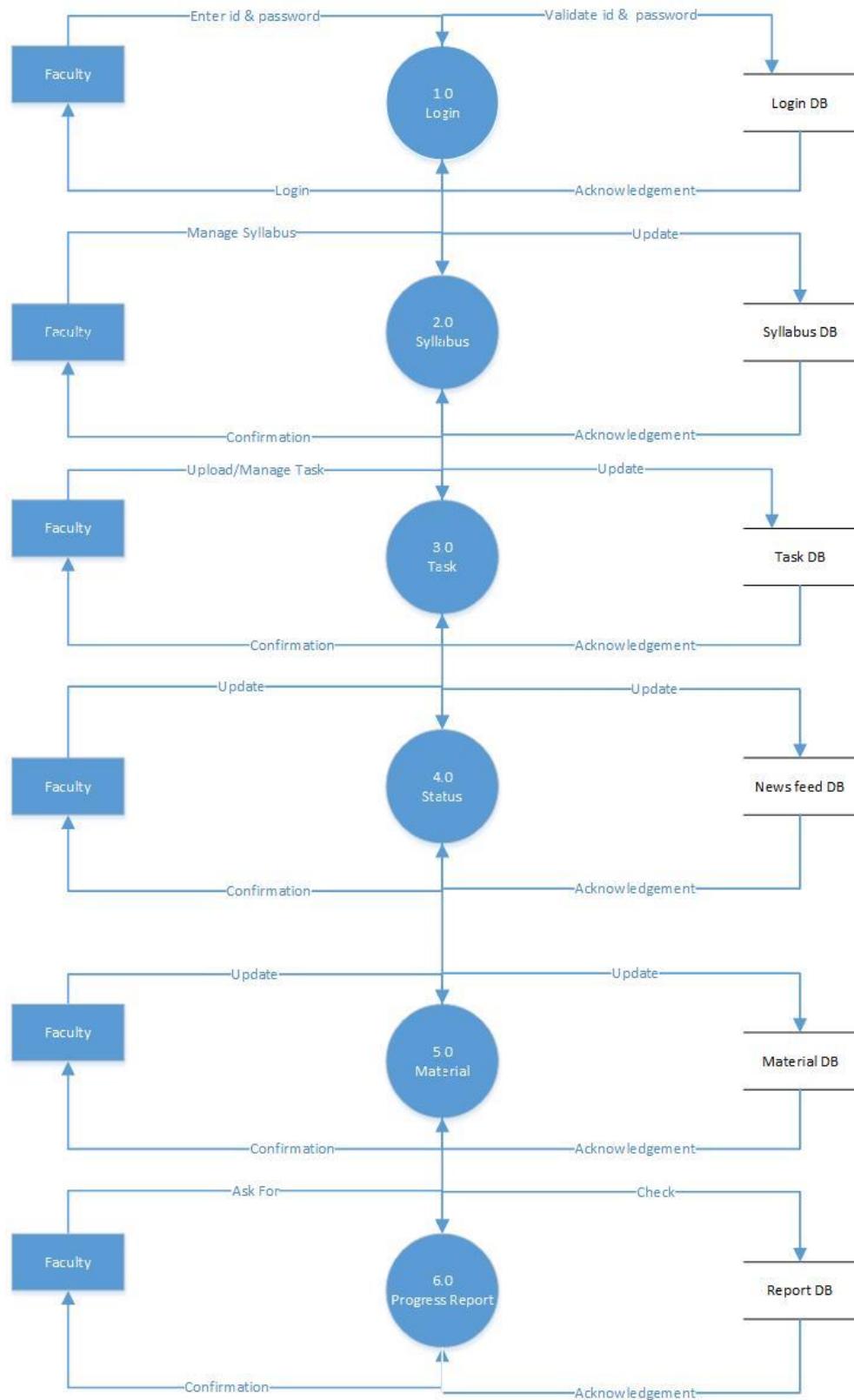


Figure 4.11 1 level DFD – Faculty

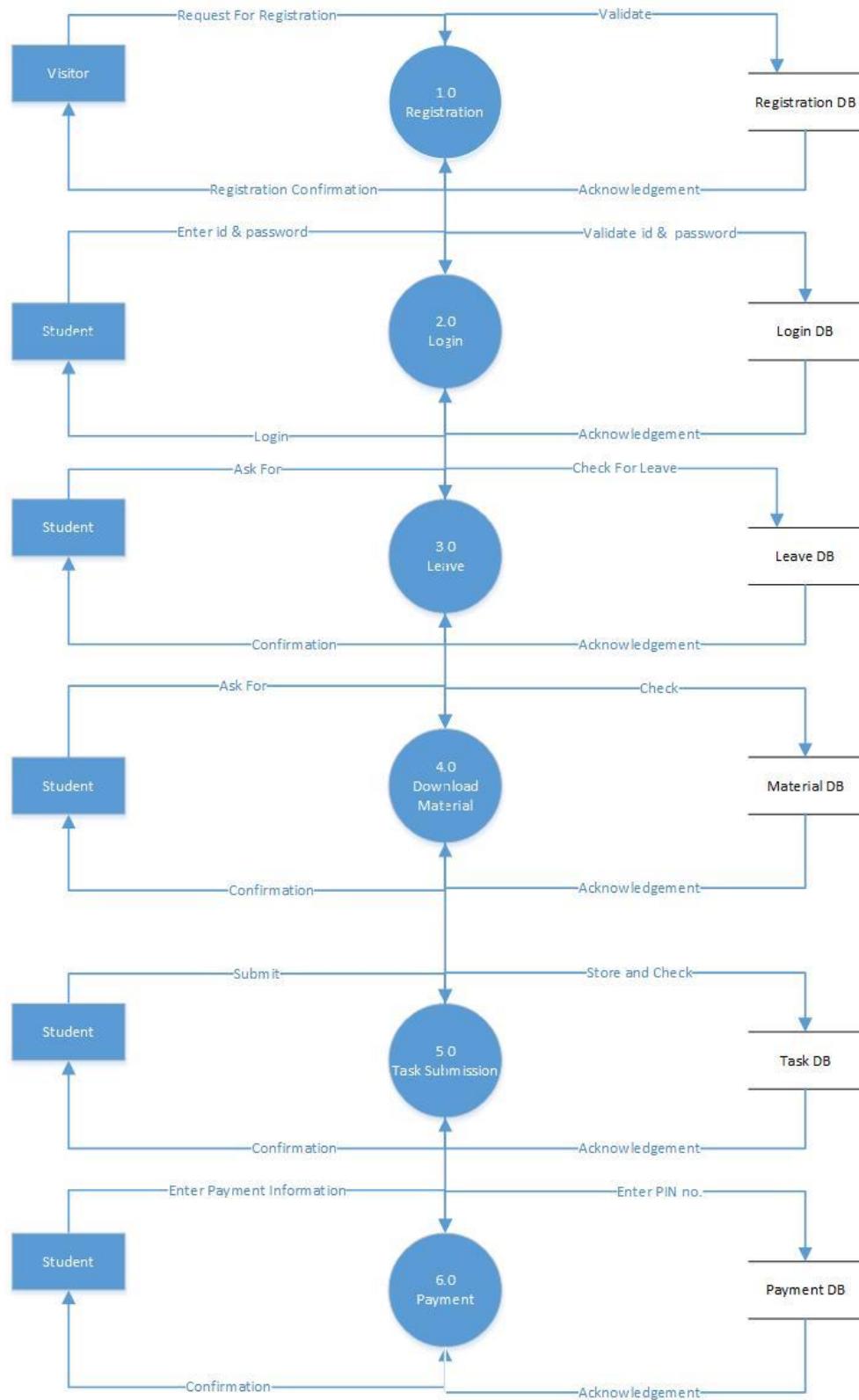


Figure 4.12 1 level DFD - Student

4.9 MAIN MODULES OF NEW SYSTEM

1. User

➤ Visitor:

Visitor visits web application. Visitor gets basic information regarding a company. Visitor checks demo lecture, demo material, faculty list. Visitor gets an appointment. Visitor gets himself/herself registered into the system in order to interact with it.

➤ Student:

Student gets himself/herself login to system and can manage his/her profile. Attendance is given and he/she can ask for leave as well. Fees schedule is shown to them. Video, material is also downloaded by them. Newly added task is displayed to all students. Task submission is done by them and grade is also given by respective faculty. Notification to pay fees is shown to them. After completion of the course or training certificate is given and after completing all the activities logging out is done.

➤ Faculty:

Faculty login to system. Faculty can manage their profile. Syllabus of each courses and training is managed by them. Submitted task of students are reviewed by them and grades are also given. Material to make available is uploaded so that student can download and watch videos of their lectures. Query submitted is solved by faculty.

i. Home Page Module

The Home page is a main page. This is a simple front page.

ii. About us Module

This page provide the all information about the college, Mission of college and developer's Information.

iii. Course Module

This page contains the detail of the courses of training/languages.

iv. Faculty Module

The Faculty Module provides the information about the Faculty in which his/her qualifications, his/her Designation, his/her more details etc.

v. Download Module

The Download Module is one of the most important for the user because we can easily download the syllabus of our courses.

2. Admin:

Admin login to system. Admin manages individual batch and corresponding timetable for each batch. Admin manages syllabus of each courses and training. Admin handles inquiries submitted by various user and give them an appointment. Admin manages certificate and fees receipts.

i. Home Page Module

This is the main page of Education Portal. That contains the information about information of user.

ii. Result Module

This page contains the results of all the students and the admin also allow admin to add, update or delete the results.

iii. Attendance Module

This page contains the attendance of all the students and it allows admin to add, update and delete the records.

4.10 SELECTION OF HARDWARE AND SOFTWARE AND JUSTIFICATION

Hardware Justification

| Requirement | Specification | Justification of Selection |
|--------------------|-----------------------|---|
| Processor | Intel Corei3 or Above | Processor can be any Intel or any Above as Our System require High Processor (Server) |
| Hard Drive | 500 GB | We need Hard Drive 500 GB at Current Requirement Analysis |
| Operating System | Windows 7 or Above | Our System can Work in any Operating System so no any Specific OS is defined |
| RAM | 2 GB or Above | Our System require Fast Performance so minimum RAM requirement is defined |

| | | |
|--------------|--------------|--|
| Wi-Fi Router | I ball Baton | Very good Performance and provide lots of features |
|--------------|--------------|--|

Table 4.25 Hardware Justification

Software Justification

| Requirement | Specification | Justification of Selection |
|---------------|---------------|---|
| Database | MySQL | MySQL have easier Query having lots of advantages |
| Application | Net Beans | Net Beans Application is famous for Developing Java Application |
| Web Server | Tomcat Server | tomcat Server has lot of Advantages and Work well with Net beans while developing Application |
| Documentation | MS Office | MS PowerPoint and MS Word is used for Documentation works. |
| Diagram | MS Visio | MS Visio provide facility for making different type of diagrams. |

Table 4.26 Software Justification

CHAPTER: 5

SYSTEM DESIGN

5.1 DATABASE DESIGN/DATABASE STRUCTURE DESIGN

5.1.1 Mapping Objects

5.1.2 Table and Relationship

5.2 SYSTEM PROCEDURAL DESIGN

5.2.1 Flow Chart

5.1 DATABASE DESIGN/DATABASE STRUCTURE DESIGN

Database design is the process of producing a detailed data model of a database this logical data model contains all the needed logical and physical design choices and physical storage parameters needed to generate a design in a Data Definition Language, which can then be used to create a database. A fully attributed data model contains detailed attributes for each entity.

5.1.1 Mapping Objects

Mapping cardinalities express the number of entities to which another entity can be associated via relationship set.

Mapping cardinalities is also referred as cardinality ratio. Based on mapping, a binary relationship set can be divided into four categories as given below.

1. One to One

- A one entity is associated with at most one entity.
- Example: - in banking system, no multiple accounts are allowed. In this situation one customer can have only one account and one account can be associated with only one Customer.

2. One to many

- A one entity is associated with zero or more entity.
- Example: - multiple accounts are allowed. So, one customer can have any number of accounts, but one account will have only one customer.

3. Many to one

- A one or more entity is associated with at most one entity.
- Example: - No multiple account are allowed, but join account are allowed so, one customer can have only one account, but one account can be associated with any number of Accounts.

4. Many to Many

- A one or more entity is associated with at zero or more entity.
- Example: - Multiple account as well as join account both are allowed. in this situation one customer can have any number of account and one account can have any number of customers

5.1.2 Table and Relationship

Identify the Relationships

The next step is to determine the relationships between the entities and to ascertain the cardinality of each relationship. The relationship is the connection between the entities, just like in the real world: what does one entity do with the other, how do they relate to each other? For example, customers buy products, products are sold to customers, a sale comprises products, a sale happens in a shop. The cardinality shows how much of one side of the relationship belongs to how much of the other side of the relationship. First, you need to state for each relationship, how much of one side belongs to exactly 1 of the other side. For example, how many customers belong to 1 sale? How many sales belong to 1 customer? How many sales take place in 1 shop?

Recursive Relationships

Sometimes an entity refers to itself. For example, think of a work hierarchy: an employee has a boss, and the boss chef is an employee too. The attribute 'boss' of the entity 'employees' refers to the entity 'employees.'

Redundant Relationships

Sometimes in your model, you will get a 'redundant relationship.' These are relationships that are already indicated by other relationships, although not directly.

In the case of our example, there is a direct relationship between customers and products. But there is also relationship from customers to sales and from sales to products, so indirectly there already is a relationship between customers and products through sales. The relationship 'Customers <----> Products' is made twice, and one of them is, therefore, redundant. In this case, products are only purchased through a sale, so the relationships 'Customers <----> Products' can be deleted. The model will then look like this:

Identifying Attributes

The data elements that you want to save for each entity are called 'attributes.' About the products that you sell, you want to know, for example, what the price is, what the name of the manufacturer is, and what the type number is. About the customers, you know their customer number, their name, and address. About the shops, you know the location code, the name, the address.

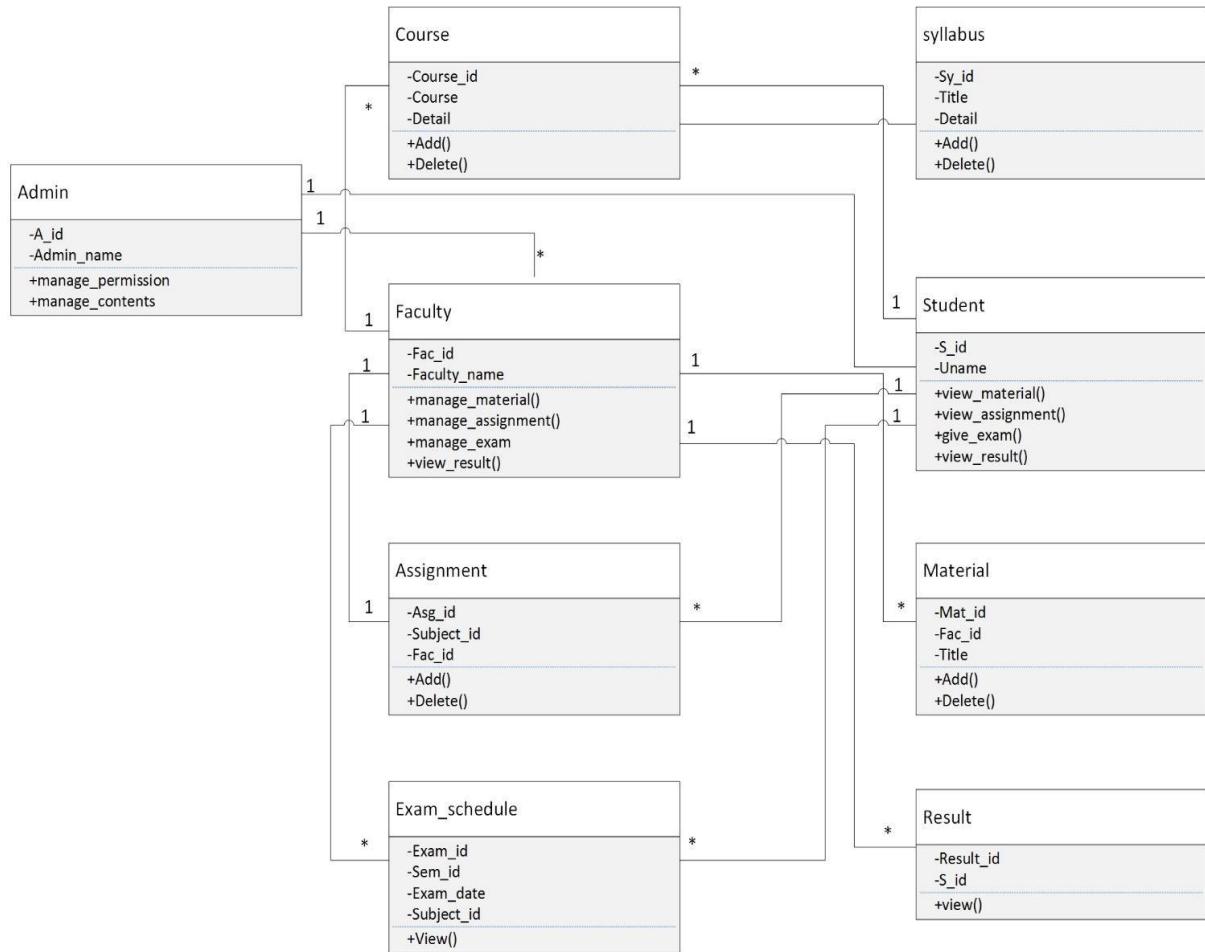


Figure 5.1 Table and Relationship Diagram

5.2 SYSTEM PROCEDURAL DESIGN

The Procedural Design is also commonly called as Component level Design. The purpose of component level design is to define data structures, algorithms, interface characteristics, and communication mechanisms for each software component identified in the architectural design. Component level design occurs after the data, architectural, and interface designs are established. The component-level design represents the software in a way that allows the designer to review it for correctness and consistency before it is built.

Cohesion

- Utility cohesion - components grouped within the same category but are otherwise unrelated
- Temporal cohesion - operations are performed to reflect a specific behavior or state

- Procedural cohesion - components grouped to allow one be invoked immediately after the preceding one was invoked with or without passing data
- Communicational cohesion - operations required same data are grouped in same class
- Sequential cohesion - components grouped to allow input to be passed from first to second and so on
- Layer cohesion - exhibited by package components when a higher level layer accesses the services of a lower layer, but lower level steps do not access higher level layer services
- Functional cohesion - module performs one and only one function

Coupling

- Data coupling - occurs when long strings of arguments are passed between components
- Stamp coupling - occurs when parts of larger data structures are passed between components
- Control coupling - occurs when one component passes control flags as arguments to another
- External coupling - occurs when a component communicates or collaborates within infrastructure component (e.g., database)
- Common coupling - occurs when several components make use of a global variable
- Content coupling - occurs when one component surreptitiously modifies internal data in another component
- Routine call coupling - occurs when one operator invokes another

Type use coupling - occurs when one component uses a data type defined in another

- Inclusion or import coupling - occurs when one component imports a package or uses the content of another

Design Notation

- Flowcharts - arrows for flow of control, diamonds for decisions, rectangles for processes
- Decision table - subsets of system conditions and actions are associated with each other to define the rules for processing inputs and events

- Program Design Language (PDL) - structured English or pseudo code used to describe processing details

5.2.1 Flow Chart

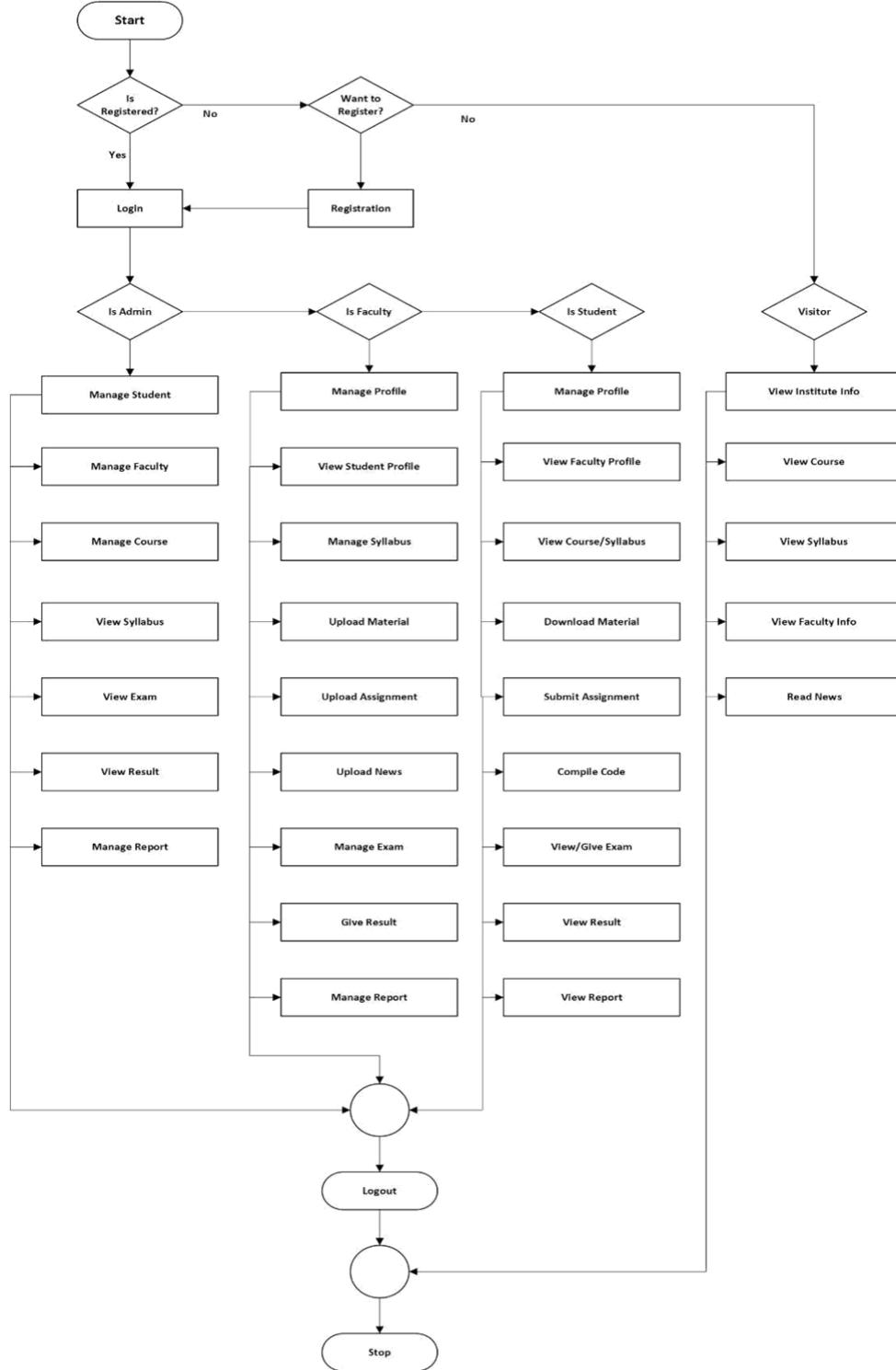


Figure 5.2 Flow Chart

CHAPTER: 6

IMPLEMENTATION PLANNING AND DETAILS

6.1 IMPLEMENTATION ENVIRONMENT

6.2 PROGRAM/MODULE SPECIFICATION

6.3 SECURITY FEATURES

6.4 CODING STANDARDS

6.5 SAMPLE CODING

6.1 IMPLEMENTATION ENVIRONMENT

1. Single vs. Multi user:

This system is web application and it calls web services behind. It is going to be used in distribute environment i.e. any number of users can access it from anywhere at the same time. So it is Multi user application.

2. GUI vs. Non-GUI:

The system is GUI. This is because today customer feels comfortable while working with images or picture. As the project is web based, it can be used by any end-user. Thus user should feel convenient while working with the system. Thus system is user-friendly.

6.2 MODULES SPECIFICATION

1. User:

➤ Visitor:

Visitor visits web application. Visitor gets basic information regarding a company. Visitor checks demo lecture, demo material, faculty list. Visitor gets an appointment. Visitor gets himself/herself registered into the system in order to interact with it.

➤ Student:

Student gets himself/herself login to system and can manage his/her profile. Attendance is given and he/she can ask for leave as well. Fees schedule is shown to them. Video, material is also downloaded by them. Newly added task is displayed to all students. Task submission is done by them and grade is also given by respective faculty. Notification to pay fees is shown to them. After completion of the course or training certificate is given and after completing all the activities logging out is done.

➤ Faculty:

Faculty login to system. Faculty can manage their profile. Syllabus of each courses and training is managed by them. Submitted task of students are reviewed by them and grades are also given. Material to make available is uploaded so that student can download and watch videos of their lectures. Query submitted is solved by faculty.

2. Admin:

Admin login to system. Admin manages individual batch and corresponding timetable for each batch. Admin manages syllabus of each courses and training. Admin handles inquiries submitted by various user and give them an appointment. Admin manages certificate and fees receipts.

6.3 SECURITY FEATURES

We have basically only one user with its own accessibility rights. User has its own username and password to manage its own confidential details. This password and username to protect user details. Failure to do this will not allow to user to access the application.

6.4 CODING STANDARDS

The coding standards are well defined and standard style of coding. with the help of the coding standard any person can go into any code and figure out what's going on and new people can get up to speed quickly. A coding standard sets out standards of doing several things such as the way variable are to be named, the code is to be laid out, the comment are to be described, etc.

This section describes coding standards, which we have used in the program. We have adopted the following coding standards.

Variable Declaration:

- We have placed the variable declaration at the beginning of the script.
- Block of declaration has aligned.
- For multiple declarations we have used new declaration on the next line.

Creating Source Code:

- We have added the comment for each and every line of the code that we have made.
- We have added the description of each and every function and their attributes and their parameters.

General Coding:

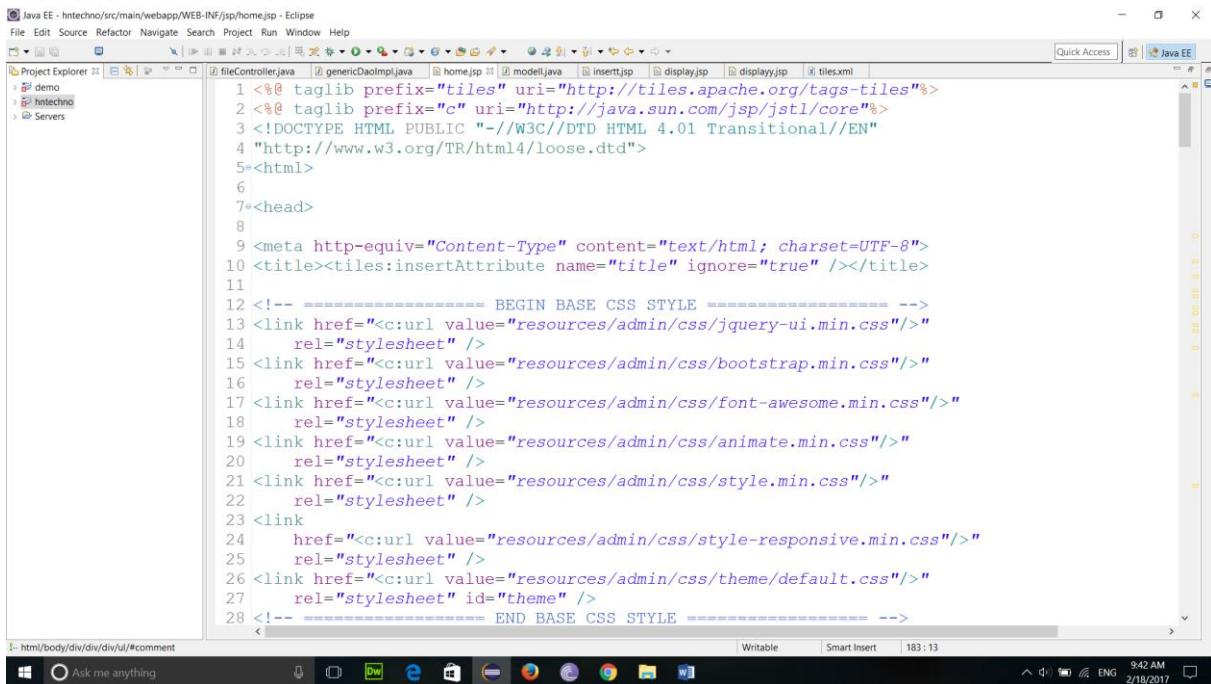
- It is acceptable to break long statements on to multiple line or leave it to one single line.
- Do not put multiple statements in one line.

- Every coding block must be indented.
- The standards indentation is three spaces.
- The use of white space and simple comments is strongly encouraged.
- All the functions, variables and constants have to be declared on the top of the file in particular form.

Comment Standards:

- Every script should begin with a comment block, which describes the script purpose; any argument used, and return values, input out-puts, and name of script.
- Comment may be used in the body of the script to explain individual sections or lines of code.
- It is also used to describe variable definition or declaration.
- Each part of the project has a specific comment layout. Inline comment layout. Inline comments should be made with the “ // ”.

6.5 SAMPLE CODING

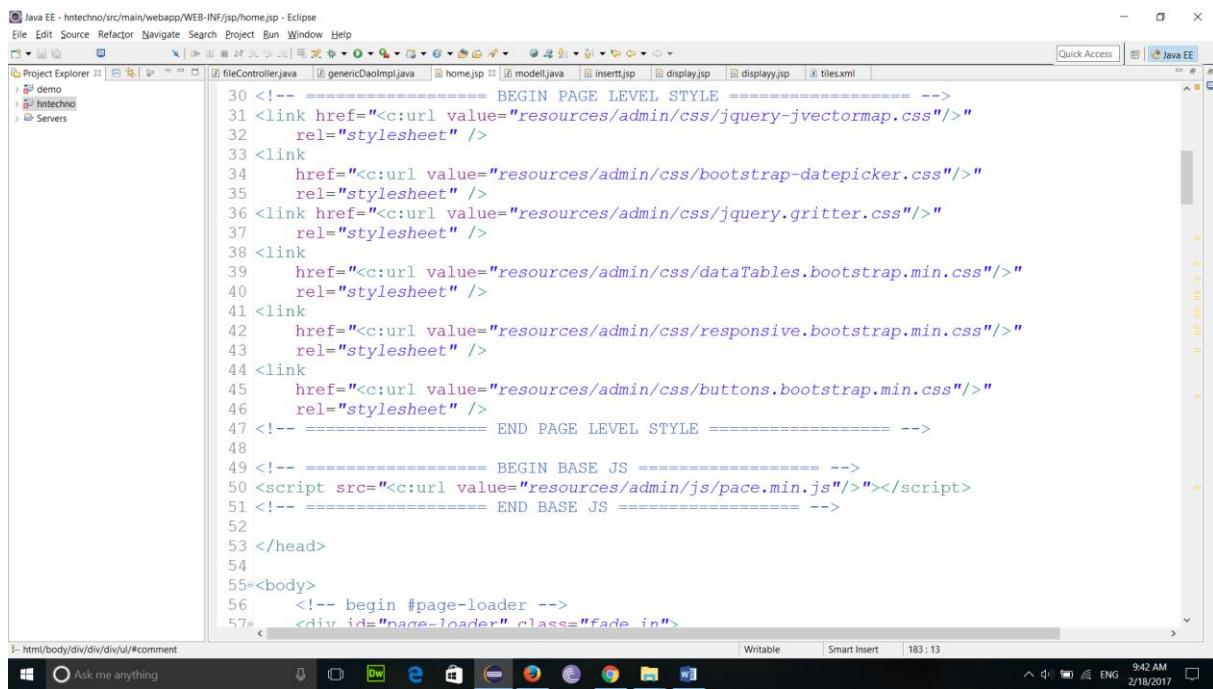


The screenshot shows the Eclipse IDE interface with the title "Java EE - hntechno/src/main/webapp/WEB-INF/jsp/home.jsp - Eclipse". The Project Explorer view on the left shows a project named "hntechno" containing a "demo" folder and a "Servers" section. The main editor area displays the content of the "home.jsp" file. The code includes JSTL tags like <%@ taglib prefix="tiles" uri="http://tiles.apache.org/tags-tiles"%> and <%@ taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core"%>. It also includes a block of CSS imports starting with <!-- ===== BEGIN BASE CSS STYLE ===== --> and ending with <!-- ===== END BASE CSS STYLE ===== -->. The code is numbered from 1 to 28. The status bar at the bottom shows "Writable" and "Smart Insert" along with a timestamp of "183:13". The taskbar at the bottom includes icons for Ask me anything, File, Open, Save, Find, Copy, Paste, Cut, Undo, Redo, Firefox, Chrome, and others.

```

1 <%@ taglib prefix="tiles" uri="http://tiles.apache.org/tags-tiles"%>
2 <%@ taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core"%>
3 <!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
4 "http://www.w3.org/TR/html4/loose.dtd">
5<html>
6
7<head>
8
9 <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
10 <title><tiles:insertAttribute name="title" ignore="true" /></title>
11
12 <!-- ===== BEGIN BASE CSS STYLE ===== -->
13 <link href=<c:url value="resources/admin/css/jquery-ui.min.css"/>" rel="stylesheet" />
14 <link href=<c:url value="resources/admin/css/bootstrap.min.css"/>" rel="stylesheet" />
15 <link href=<c:url value="resources/admin/css/font-awesome.min.css"/>" rel="stylesheet" />
16 <link href=<c:url value="resources/admin/css/animate.min.css"/>" rel="stylesheet" />
17 <link href=<c:url value="resources/admin/css/style.min.css"/>" rel="stylesheet" />
18 <link href=<c:url value="resources/admin/css/style-responsive.min.css"/>" rel="stylesheet" />
19 <link href=<c:url value="resources/admin/css/theme/default.css"/>" rel="stylesheet" id="theme" />
20 <!-- ===== END BASE CSS STYLE ===== -->
21 <!-->
22 <!-->
23 <!-->
24 <!-->
25 <!-->
26 <!-->
27 <!-->
28 <!-->

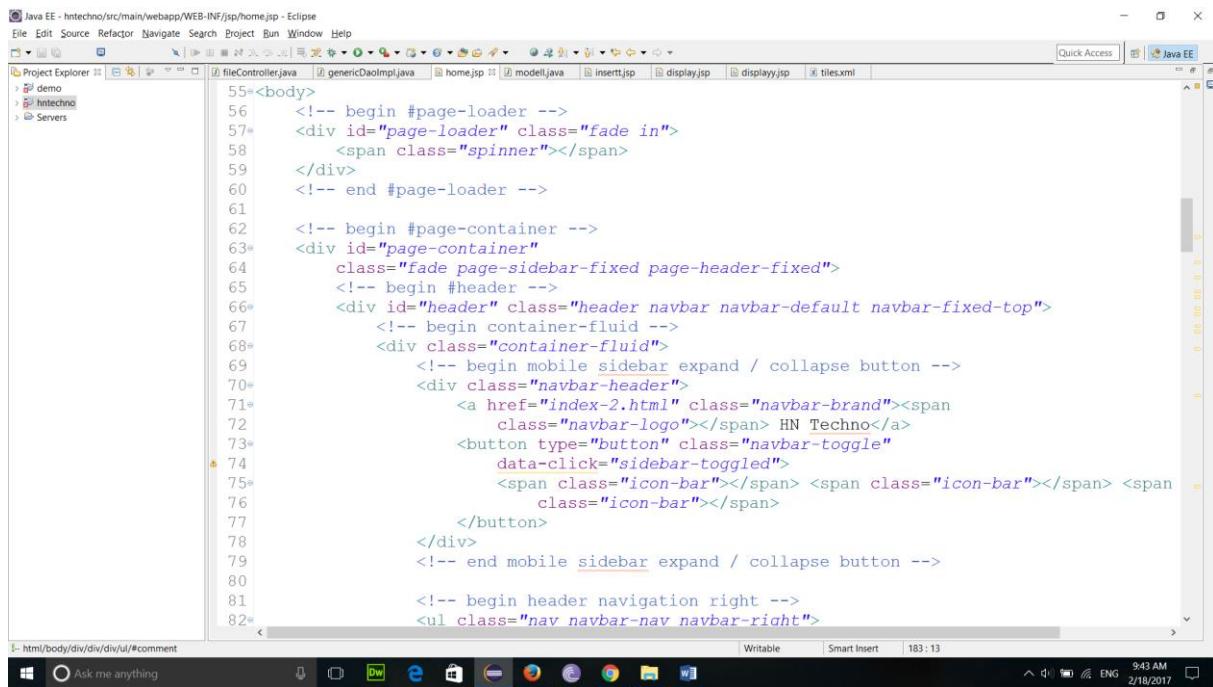
```



```

30 <!-- ===== BEGIN PAGE LEVEL STYLE ===== -->
31 <link href=<c:url value="resources/admin/css/jquery-jvectormap.css"/>
32   rel="stylesheet" />
33 <link
34   href=<c:url value="resources/admin/css/bootstrap-datepicker.css"/>
35   rel="stylesheet" />
36 <link href=<c:url value="resources/admin/css/jquery.gritter.css"/>
37   rel="stylesheet" />
38 <link
39   href=<c:url value="resources/admin/css/dataTables.bootstrap.min.css"/>
40   rel="stylesheet" />
41 <link
42   href=<c:url value="resources/admin/css/responsive.bootstrap.min.css"/>
43   rel="stylesheet" />
44 <link
45   href=<c:url value="resources/admin/css/buttons.bootstrap.min.css"/>
46   rel="stylesheet" />
47 <!-- ===== END PAGE LEVEL STYLE ===== -->
48
49 <!-- ===== BEGIN BASE JS ===== -->
50 <script src=<c:url value="resources/admin/js/pace.min.js"/>></script>
51 <!-- ===== END BASE JS ===== -->
52
53 </head>
54
55<body>
56   <!-- begin #page-loader -->
57   <div id="page-loader" class="fade in">

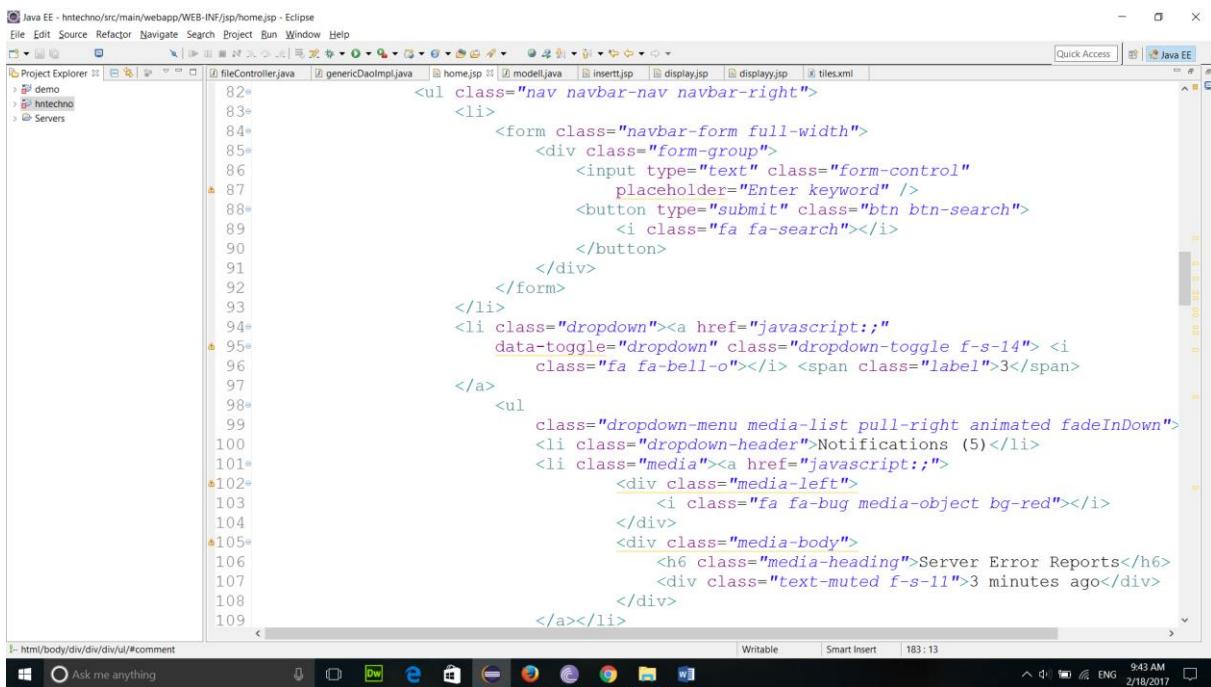
```



```

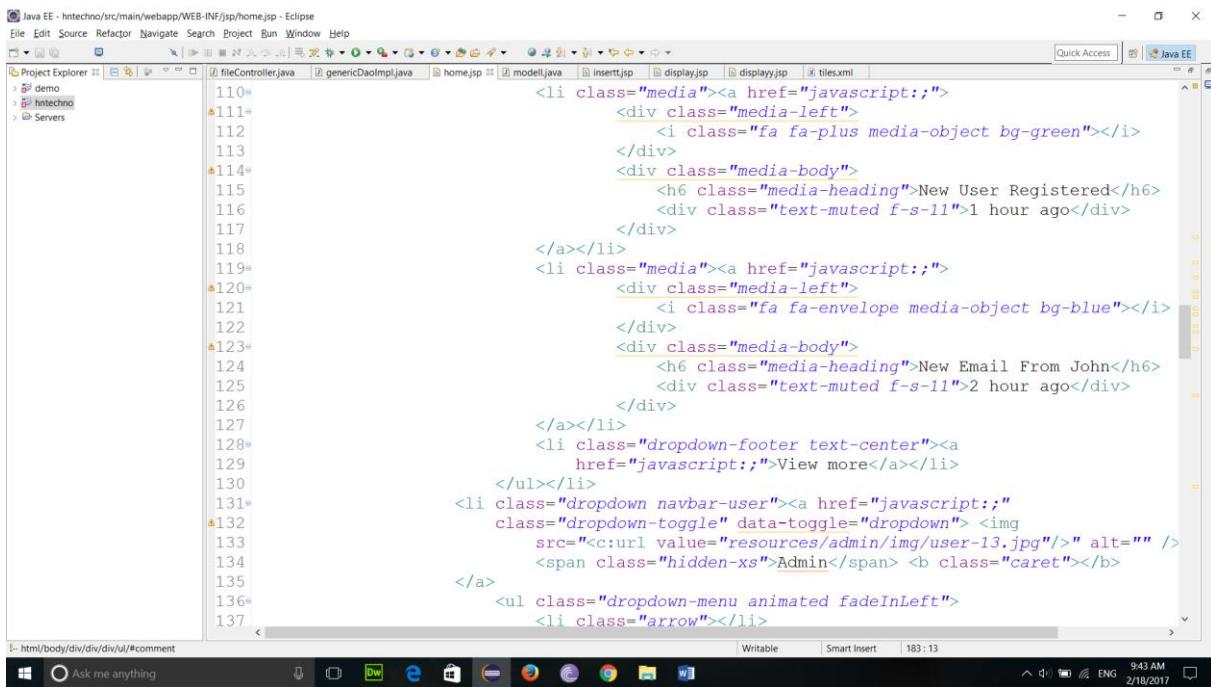
55<body>
56   <!-- begin #page-loader -->
57   <div id="page-loader" class="fade in">
58     <span class="spinner"></span>
59   </div>
60   <!-- end #page-loader -->
61
62   <!-- begin #page-container -->
63   <div id="page-container"
64     class="fade page-sidebar-fixed page-header-fixed">
65     <!-- begin #header -->
66     <div id="header" class="header navbar navbar-default navbar-fixed-top">
67       <!-- begin container-fluid -->
68       <div class="container-fluid">
69         <!-- begin mobile sidebar expand / collapse button -->
70         <div class="navbar-header">
71           <a href="index-2.html" class="navbar-brand"><span
72             class="navbar-logo"></span> HN Techno</a>
73           <button type="button" class="navbar-toggle"
74             data-click="sidebar-toggler">
75             <span class="icon-bar"></span> <span class="icon-bar"></span> <span
76               class="icon-bar"></span>
77           </button>
78         </div>
79         <!-- end mobile sidebar expand / collapse button -->
80
81       <!-- begin header navigation right -->
82       <ul class="nav navbar-nav navbar-right">

```



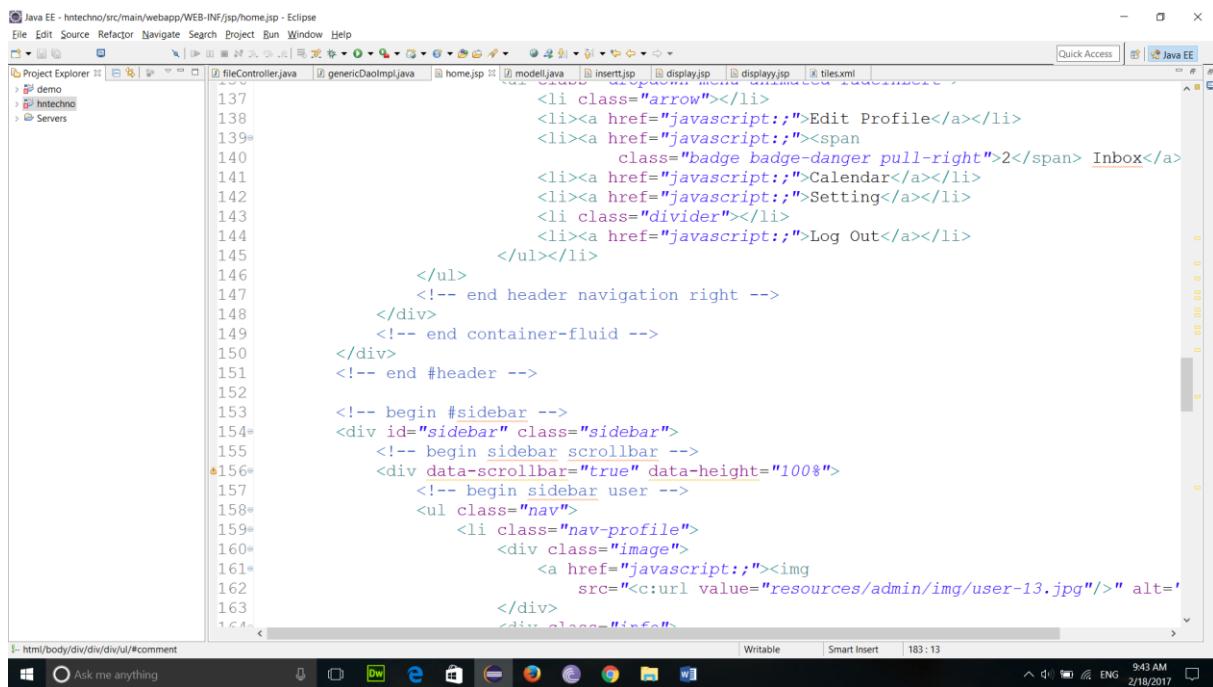
```

<ul class="nav navbar-nav navbar-right">
    <li>
        <form class="navbar-form full-width">
            <div class="form-group">
                <input type="text" class="form-control" placeholder="Enter keyword" />
                <button type="submit" class="btn btn-search">
                    <i class="fa fa-search"></i>
                </button>
            </div>
        </form>
    </li>
    <li class="dropdown"><a href="javascript:;">
        <span data-toggle="dropdown" class="dropdown-toggle f-s-14"> <i class="fa fa-bell-o"></i> <span class="label">3</span>
    </a>
    <ul class="dropdown-menu media-list pull-right animated fadeInDown">
        <li class="dropdown-header">Notifications (5)</li>
        <li class="media"><a href="javascript:;">
            <div class="media-left">
                <i class="fa fa-bug media-object bg-red"></i>
            </div>
            <div class="media-body">
                <h6 class="media-heading">Server Error Reports</h6>
                <div class="text-muted f-s-11">3 minutes ago</div>
            </div>
        </a></li>
    
```



```

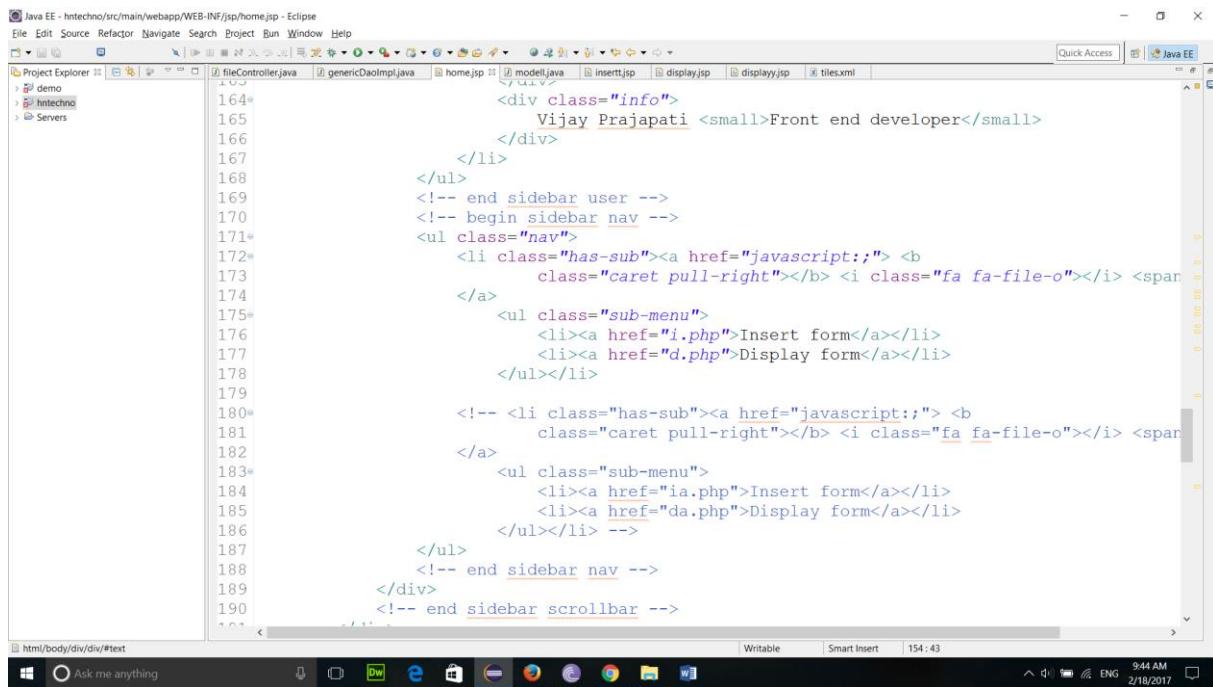
    <li class="media"><a href="javascript:;">
        <div class="media-left">
            <i class="fa fa-plus media-object bg-green"></i>
        </div>
        <div class="media-body">
            <h6 class="media-heading">New User Registered</h6>
            <div class="text-muted f-s-11">1 hour ago</div>
        </div>
    </a></li>
    <li class="media"><a href="javascript:;">
        <div class="media-left">
            <i class="fa fa-envelope media-object bg-blue"></i>
        </div>
        <div class="media-body">
            <h6 class="media-heading">New Email From John</h6>
            <div class="text-muted f-s-11">2 hour ago</div>
        </div>
    </a></li>
    <li class="dropdown-footer text-center"><a href="javascript:;*>View more</a></li>
</ul></li>
<li class="dropdown navbar-user"><a href="javascript:;*>
    <span class="hidden-xs">Admin</span> <b class="caret"></b>
</a>
    <ul class="dropdown-menu animated fadeInLeft">
        <li class="arrow"></li>
    
```



```

137             <li class="arrow"></li>
138             <li><a href="javascript:;">Edit Profile</a></li>
139             <li><a href="javascript:;">span
140                 class="badge badge-danger pull-right">2</span> Inbox</a>
141             <li><a href="javascript:;">Calendar</a></li>
142             <li><a href="javascript:;">Setting</a></li>
143             <li class="divider"></li>
144             <li><a href="javascript:;">Log Out</a></li>
145         </ul></li>
146         <!-- end header navigation right -->
147     </div>
148     <!-- end container-fluid -->
149 </div>
150 <!-- end #header -->
151
152 <!-- begin #sidebar -->
153 <div id="sidebar" class="sidebar">
154     <!-- begin sidebar scrollbar -->
155     <div data-scrollbar="true" data-height="100%">
156         <!-- begin sidebar user -->
157         <ul class="nav">
158             <li class="nav-profile">
159                 <div class="image">
160                     <a href="javascript;"><img
161                         src=<c:url value="resources/admin/img/user-13.jpg"/></a> alt='
162                     </div>
163             </li>
164         </ul>
165         <div class="info">
166             Vijay Prajapati <small>Front end developer</small>
167         </div>
168     </div>
169     <!-- end sidebar user -->
170     <!-- begin sidebar nav -->
171     <ul class="nav">
172         <li class="has-sub"><a href="javascript;"> <b
173             class="caret pull-right"></b> <i class="fa fa-file-o"></i> <span
174             ></a>
175             <ul class="sub-menu">
176                 <li><a href="i.php">Insert form</a></li>
177                 <li><a href="d.php">Display form</a></li>
178             </ul></li>
179         <!-- <li class="has-sub"><a href="javascript;"> <b
180             class="caret pull-right"></b> <i class="fa fa-file-o"></i> <span
181             ></a>
182             <ul class="sub-menu">
183                 <li><a href="ia.php">Insert form</a></li>
184                 <li><a href="da.php">Display form</a></li>
185             </ul></li> -->
186         </ul>
187         <!-- end sidebar nav -->
188     </div>
189     <!-- end sidebar scrollbar -->
190 </div>

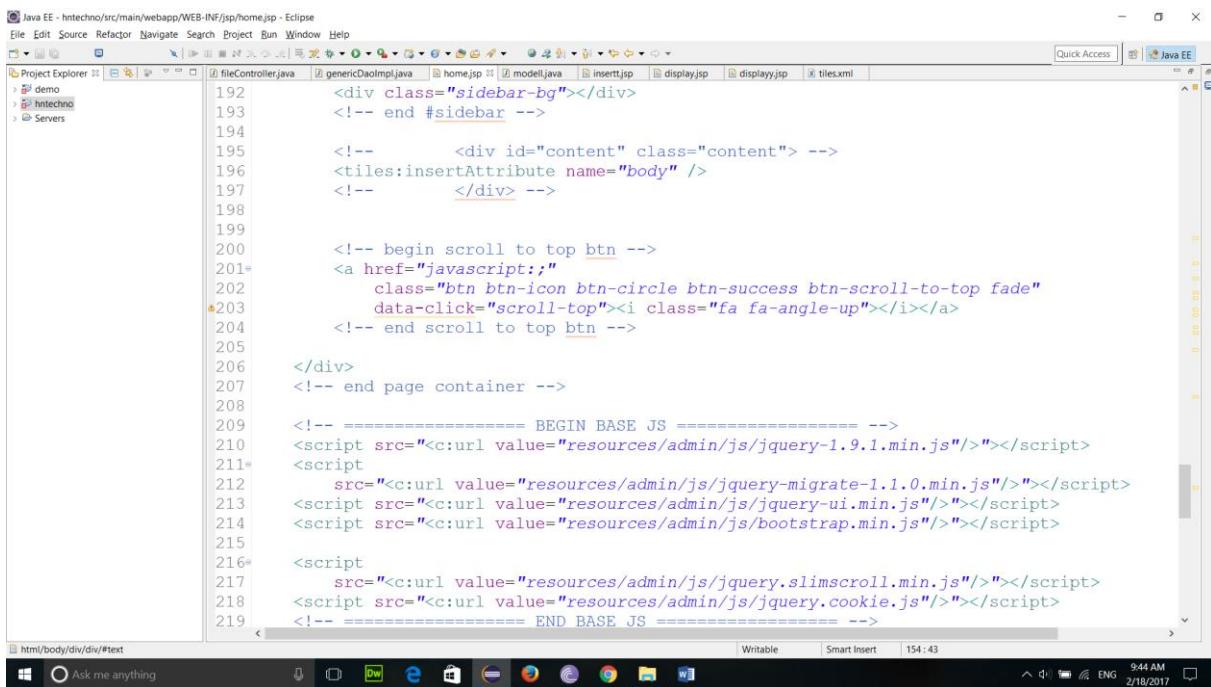
```



```

164         <div class="info">
165             Vijay Prajapati <small>Front end developer</small>
166         </div>
167     </li>
168     <!-- end sidebar user -->
169     <!-- begin sidebar nav -->
170     <ul class="nav">
171         <li class="has-sub"><a href="javascript;"> <b
172             class="caret pull-right"></b> <i class="fa fa-file-o"></i> <span
173             ></a>
174             <ul class="sub-menu">
175                 <li><a href="i.php">Insert form</a></li>
176                 <li><a href="d.php">Display form</a></li>
177             </ul></li>
178         <!-- <li class="has-sub"><a href="javascript;"> <b
179             class="caret pull-right"></b> <i class="fa fa-file-o"></i> <span
180             ></a>
181             <ul class="sub-menu">
182                 <li><a href="ia.php">Insert form</a></li>
183                 <li><a href="da.php">Display form</a></li>
184             </ul></li> -->
185         </ul>
186         <!-- end sidebar nav -->
187     </div>
188     <!-- end sidebar scrollbar -->
189 </div>

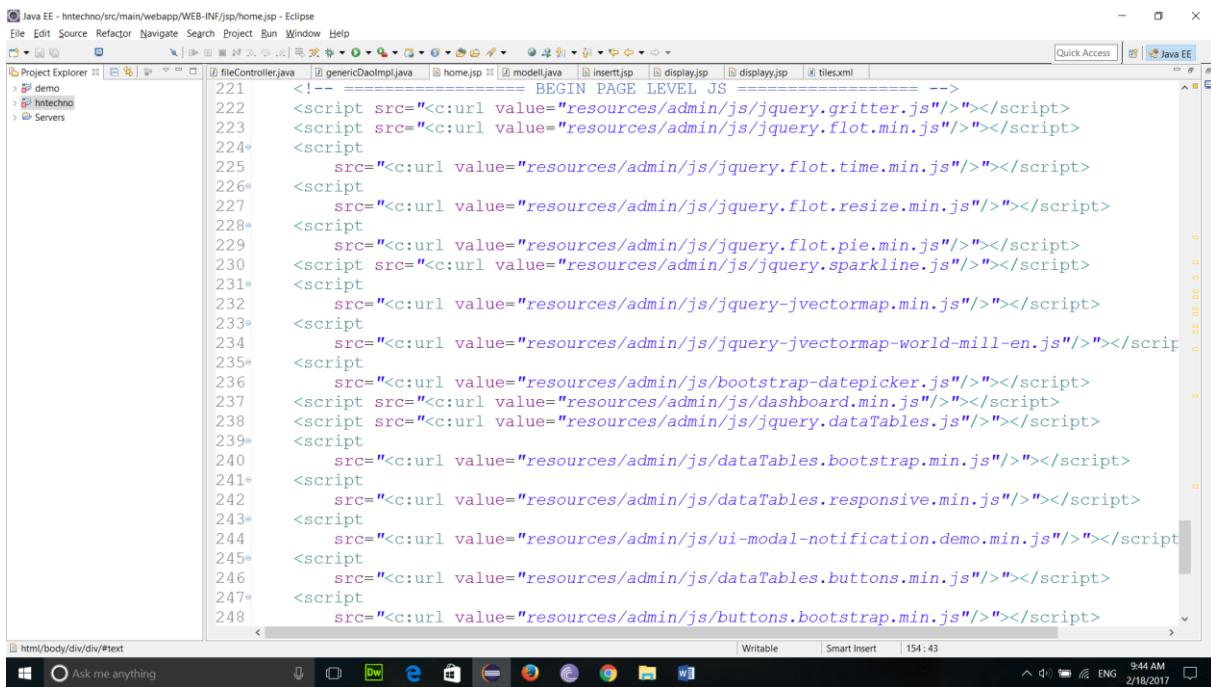
```



```

Java EE - hntchno/src/main/webapp/WEB-INF/jsp/home.jsp - Eclipse
File Edit Source Refactor Navigate Search Project Run Window Help
Project Explorer Servers
fileController.java genericDaoImpl.java home.jsp model.java insert.jsp display.jsp display.jsp tiles.xml
192     <div class="sidebar-bg"></div>
193     <!-- end #sidebar -->
194
195     <!-- begin content -->
196     <tiles:insertAttribute name="body" />
197     </div> -->
198
199
200     <!-- begin scroll to top btn -->
201     <a href="javascript:;" class="btn btn-icon btn-circle btn-success btn-scroll-to-top fade" data-click="scroll-top">i class="fa fa-angle-up"></i></a>
202     <!-- end scroll to top btn -->
203
204
205
206     </div>
207     <!-- end page container -->
208
209     <!-- ===== BEGIN BASE JS ===== -->
210     <script src=""></script>
211     <script
212         src=""></script>
213     <script src=""></script>
214     <script src=""></script>
215
216     <script
217         src=""></script>
218     <script src=""></script>
219     <!-- ===== END BASE JS ===== -->

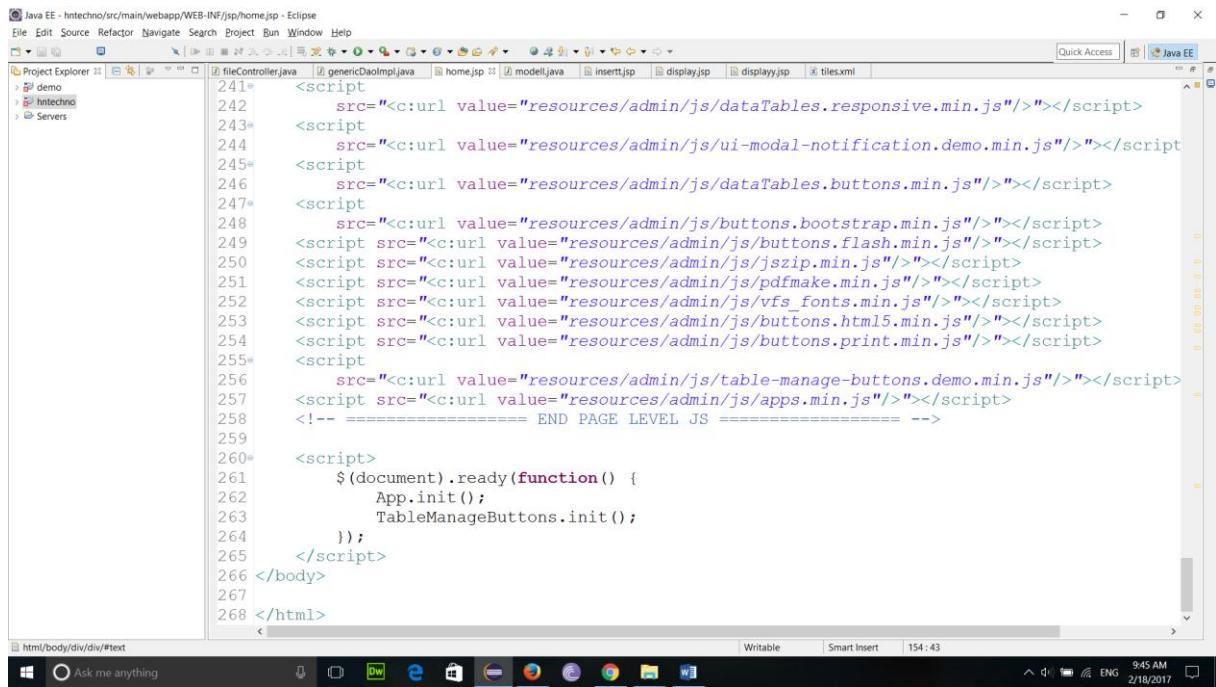
```



```

Java EE - hntchno/src/main/webapp/WEB-INF/jsp/home.jsp - Eclipse
File Edit Source Refactor Navigate Search Project Run Window Help
Project Explorer Servers
fileController.java genericDaoImpl.java home.jsp model.java insert.jsp display.jsp display.jsp tiles.xml
221     <!-- ===== BEGIN PAGE LEVEL JS ===== -->
222     <script src=""></script>
223     <script src=""></script>
224     <script
225         src=""></script>
226     <script
227         src=""></script>
228     <script
229         src=""></script>
230     <script src=""></script>
231     <script
232         src=""></script>
233     <script
234         src=""></script>
235     <script
236         src=""></script>
237     <script src=""></script>
238     <script src=""></script>
239     <script
240         src=""></script>
241     <script
242         src=""></script>
243     <script
244         src=""></script>
245     <script
246         src=""></script>
247     <script
248         src=""></script>

```



The screenshot shows the Eclipse IDE interface with the title "Java EE - htechno/src/main/webapp/WEB-INF/jsp/home.jsp - Eclipse". The code editor displays Java code for a JSP page, specifically the `fileController.java` file. The code includes numerous script tags with URLs pointing to various JavaScript files such as `responsive.min.js`, `ui-modal-notification.demo.min.js`, `buttons.min.js`, `bootstrap.min.js`, `buttons.flash.min.js`, `jszip.min.js`, `pdfmake.min.js`, `vfs_fonts.min.js`, `buttons.html5.min.js`, and `buttons.print.min.js`. The code also contains a `$(document).ready(function() { ... })` block for initializing application logic.

```

241<script
242    src="<c:url value="resources/admin/js/dataTables.responsive.min.js"/>"></script>
243<script
244    src="<c:url value="resources/admin/js/ui-modal-notification.demo.min.js"/>"></script>
245<script
246    src="<c:url value="resources/admin/js/dataTables.buttons.min.js"/>"></script>
247<script
248    src="<c:url value="resources/admin/js/buttons.bootstrap.min.js"/>"></script>
249<script src="<c:url value="resources/admin/js/buttons.flash.min.js"/>"></script>
250<script src="<c:url value="resources/admin/js/jszip.min.js"/>"></script>
251<script src="<c:url value="resources/admin/js/pdfmake.min.js"/>"></script>
252<script src="<c:url value="resources/admin/js/vfs_fonts.min.js"/>"></script>
253<script src="<c:url value="resources/admin/js/buttons.html5.min.js"/>"></script>
254<script src="<c:url value="resources/admin/js/buttons.print.min.js"/>"></script>
255<script
256    src="<c:url value="resources/admin/js/table-manage-buttons.demo.min.js"/>"></script>
257<script src="<c:url value="resources/admin/js/apps.min.js"/>"></script>
258
259<script>
260    $(document).ready(function() {
261        App.init();
262        TableManageButtons.init();
263    });
264</script>
265</body>
266</html>

```

CHAPTER: 7

TESTING

7.1 TESTING PLAN

7.2 TESTING STRATEGY

7.3 TESTING METHODS

7.4 TEST CASES

7.1 TESTING PLAN

A test plan can be defined as a document describing the scope, approach, resources, and schedule of intended testing activities. It identifies test items, the features to be tested, the testing tasks, who will do each task, and any risks requiring contingency planning.

In software testing, a test plan gives detailed testing information regarding an upcoming testing effort, including

- Scope of testing
- Schedule
- Test Deliverables
- Release Criteria
- Risks and Contingencies

7.2 TESTING STRATEGY

Unit Testing

- A Unit test tests only the functionality of a certain component.
- In this testing individual components and models are tested to ensure that they operate correctly. We had tested each and every form of all the modules such as working address, public information, personal information, leave request etc.
- For these we have checked the database for particular entry whether it contains the valid entry or not which also includes the validation of the form.

Integration Test

- Integration tests are designed to test the way individual components work jointly. Modules that have been unit tested independently are now combined together to test the integration.
- This technique is a systematic technique for constructing the program structure while at the same time conducting tests to uncover errors associated with interfacing.
- It checks whether all Modules correctly works while integrating with each other. Alpha testing is conducted at the developer's site by customer.

Interface Testing

In the system, standards tests for GUIs have been performed, which are as follows:

- Testing the screen control for its position and size.

- The position and related labels for all controls were checked.
- Name of the form in system is given appropriately.
- All menu functions and sub functions were verified for correctness.
- Each menu functions were tested, whether it invokes the corresponding functionality properly.

7.3 TESTING METHODS

MODELS OF TESTING:

There are different Models of testing. On the basis of testing methods there are two types of testing:

1. Black-box testing
2. White-box testing

BLACK BOX TESTING:

1. EQUIVALENCE PARTITIONING:

Black-box technique that divides the input domain into classes of data from which test cases can be derived an ideal test case uncovers a class of errors that might require many arbitrary test cases to be executed before a general error is observed. Equivalence class guidelines:

- If input condition specifies a range, one valid and two invalid equivalence classes are defined.
- If an input condition requires a specific value, one valid and two invalid equivalence classes are defined.
- If an input condition specifies a member of a set, one valid and one invalid equivalence class is defined.

If an input condition is Boolean, one valid and one invalid equivalence class is defined.

2. BOUNDARY VALUE ANALYSIS:

Black-box technique that focuses on the boundaries of the input domain rather than its entering. BVA guidelines:

- If input condition specifies a range bounded by values a and b, test cases should include a and b, values just above and just below a and b.

- If an input condition specifies and number of values, test cases should be exercise the minimum and maximum numbers, as well as values just above and just below the minimum and maximum values.
- Apply guidelines 1 and 2 to output conditions, test cases should be designed to produce the minimum and maxim output reports If internal program data structures have boundaries (e.g. size limitations), be certain to test the boundaries.

WHITE-BOX TESTING:

1. STATEMENT COVERAGE:

Statement coverage methodology focuses on, designing test cases so that, every statement in a program is executed at least once. No statement in the program should remain unreachable.

2. BRANCH COVERAGE:

Test cases are designed such that, different branch conditions. Give true and false values in some execution. All branches are traversed. Branch testing guarantees statement coverage. It is a stronger testing compared to the statement coverage-based testing.

3. PATH COVERAGE:

Design test cases such that, all linearly independent paths (LIP) in the program are executed at least once. To understand the path coverage-based testing we need to learn how to draw control flow graph of a program. A control flow graph (CFG) describes the sequence in which different instructions or statements of a program get executed. The way control flows through the program.

4. CONDITION COVERAGE:

Test cases are designed such that, each component of a composite conditional expression It help us to, Gives both true and false values. To check for all combination of conditions.

7.4 TEST CASES

The purpose of a test case is to describe how you intend to empirically verify that the software being developed conforms to the specifications. In other words, you need to be able to show that it can correctly carry out its intended functions. The test case should be written with enough clarity and detail that it could be given to an independent tester and have the tests properly carried out.

Test Case Description:

A test case contains all the information necessary to verify some particular functionality of the software:

Purpose:

Describe the features of the software to be tested, and the particular behavior being verified by this test. Requirement Traceability: A cross reference to the numbers of the requirements (in the system specification) which are being verified in this test

Setup:

Describe all the steps necessary to setup the software environment necessary to carry out the test.

Test Data and get expected output:

- Write the actual input data to be provided and the expected output for your actual working product. You must provide the actual input data values, not just a description. Often the test data can be shown in tabular form, with a column of input items and the corresponding column of expected outputs.

CASE 1: FOR LOGIN

Project: HNTechno Follow Up

Objective: To check whether username and password is valid or not.

Page: LOGIN

| Test Data Sr. No | Steps | Expected Data | Status |
|----------------------------|--|---|---------------|
| 1 | Enter username, password and press submit button | Should navigate to user home page | PASS |
| 2 | Enter username and press submit button | Should display message "please enter password" | Invalid |
| 3 | Enter password and press submit button | Should display message "please enter username" | Invalid |
| 4 | Enter blank username, password and press submit button | Should display message "please enter username and password" | Invalid |
| 5 | Enter wrong username | Should display message | Invalid |

Table 7.1 Login

CASE 2:

Project: HNTechno Follow Up

Objective: To check whether username and password is valid or not.

Page: REGISTRATION

| Test Data Sr. No | Steps | Expected Result | Status |
|----------------------------|---|---|---------------|
| 1 | Enter all the required details on registration page | Should navigate to user login page | Pass |
| 2 | Enter invalid field on registration page | Should display a message "Please enter all details" | Invalid |
| 3 | Enter alphabetic character where only numeric are allowed | Should not allow any character in text box. | Invalid |
| 4 | Enter email in wrong format | Should display a message "please enter email in correct format" | Invalid |

Table 7.2 Registration

CASE 3:

Project: HNTechno Follow Up

Objective: To check whether format of uploading file is valid or not.

Page: MATERIAL UPLOAD

| Test Data Sr. No | Steps | Expected Result | Status |
|----------------------------|--|--|---------------|
| 1 | Get the file from the machine and check. | You will be able to browse your local machine for images | Pass |

Table 7.3 File Upload Format

CASE 4:

Project: HNTechno Follow Up

Objective: To check whether file size of uploading file is valid or not.

Page: MATERIAL UPLOAD

| Test Data Sr. No | Steps | Expected Result | Status |
|----------------------------|--|----------------------------|---------------|
| 1 | Get the file from the machine and check. | You file will be uploaded. | Pass |

Table 7.4 File Size Upload Format

CHAPTER: 8

SCREEN SHOTS

8.1 SCREEN SHOTS

8.1 SCREEN SHOTS

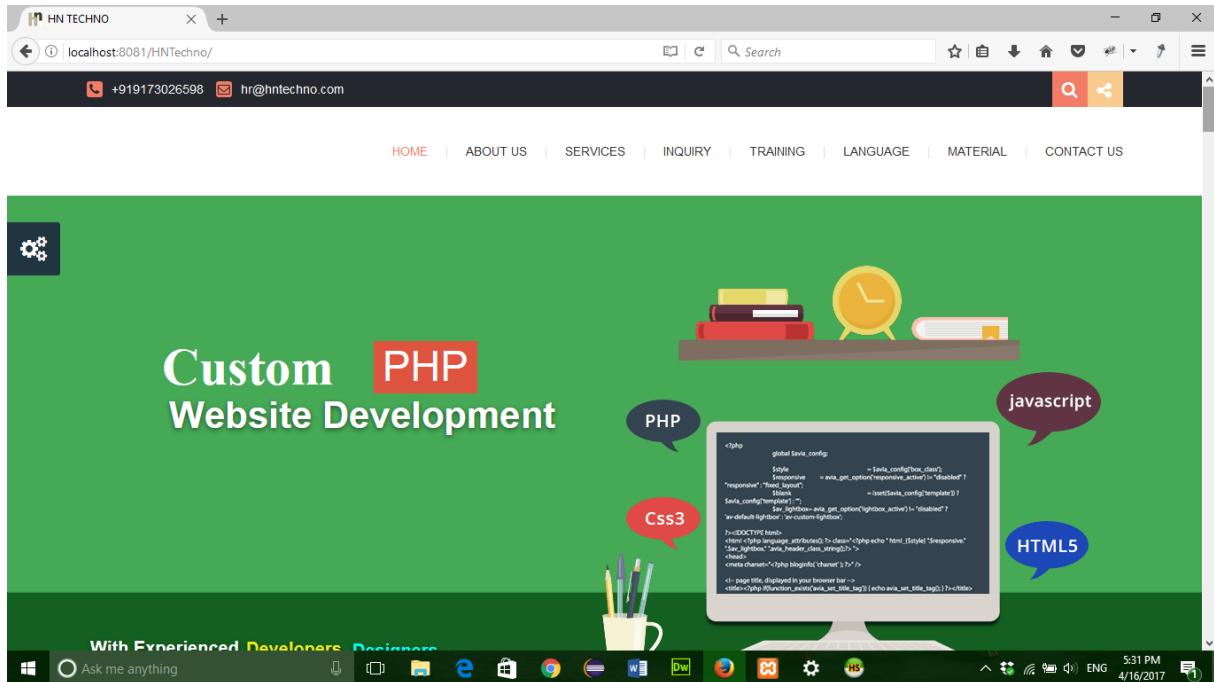


Figure 8.1 Home Page

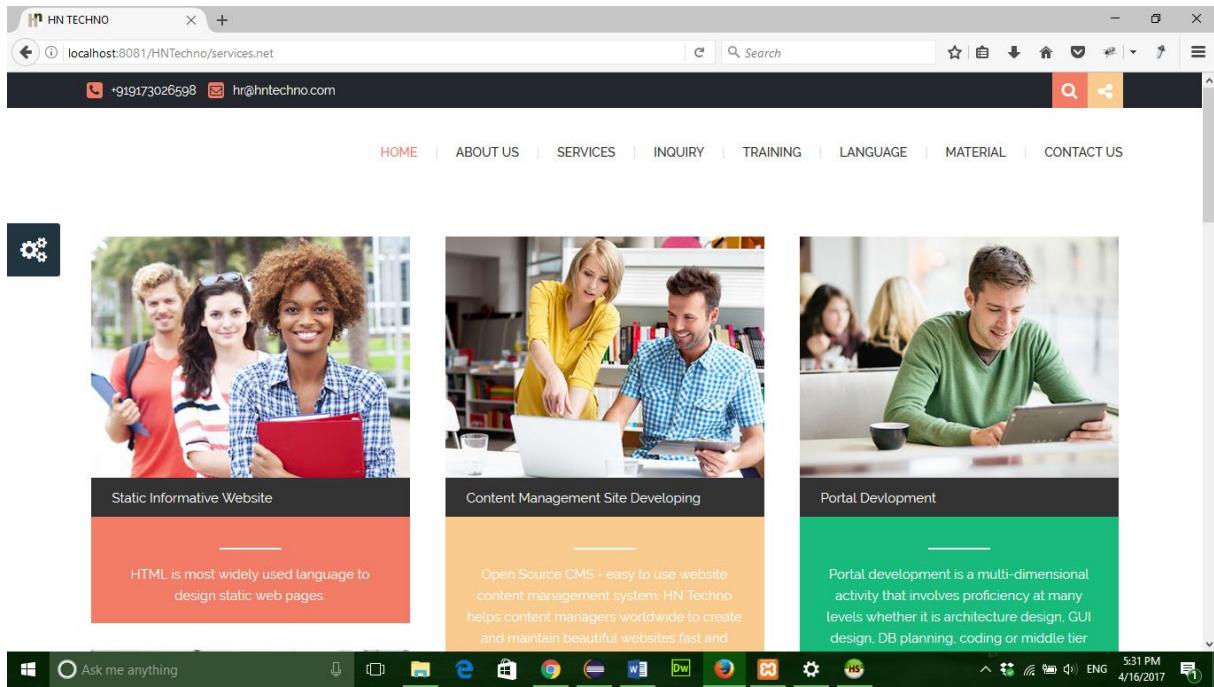


Figure 8.2 Services

The screenshot shows a web browser window for the HN TECHNO website at localhost:8081/HNTechno/userinquiry.net. The page has a black header with a gear icon and navigation links for HOME, ABOUT US, SERVICES, INQUIRY, TRAINING, LANGUAGE, MATERIAL, and CONTACT US. A search bar is at the top right. The main content area has a white background with a large red "INQUIRY" button. Below it is a section titled "Inquiry" with form fields for First Name, Middle Name, Last Name, Email Id, Phone No, and Gender. An "Add" button is at the bottom left.

INQUIRY

Home

Inquiry

First Name*: Enter First Name

Middle Name*: Enter Middle Name

Last Name*: Enter Last Name

Email Id *: Enter Email Id

Phone No *: 0

Gender *: Male Female

Add

Figure 8.3 Inquiry

The screenshot shows a web browser window for the HN TECHNO website at localhost:8081/HNTechno/contact.net. The page has a black header with a gear icon and navigation links for HOME, ABOUT US, SERVICES, INQUIRY, TRAINING, LANGUAGE, MATERIAL, and CONTACT US. A search bar is at the top right. The main content area has a white background with sections for "Contact us" and "Offices Info". The "Contact us" section includes input fields for Your Name, Your Email, Subject, and Your Message, along with a "Send" button. The "Offices Info" section lists the address, phone number (+91) 91730 26598, and email (hr@hntechno.com).

HN TECHNO

Ask me anything

+919173026598 hr@hntechno.com

HOME | ABOUT US | SERVICES | INQUIRY | TRAINING | LANGUAGE | MATERIAL | CONTACT US

Contact us

Your Name

Your Email

Subject

Your Message

Send

Offices Info

Address:
G/3,Sagar Apartment.
Nr. Kalasagar Mall,
Inside Sur-sagar.
Ghatodia,Ahmedabad 380061

Phone number:
(+91) 91730 26598

E-mail:
hr@hntechno.com
www.hntechno.com

Figure 8.4 Contact us

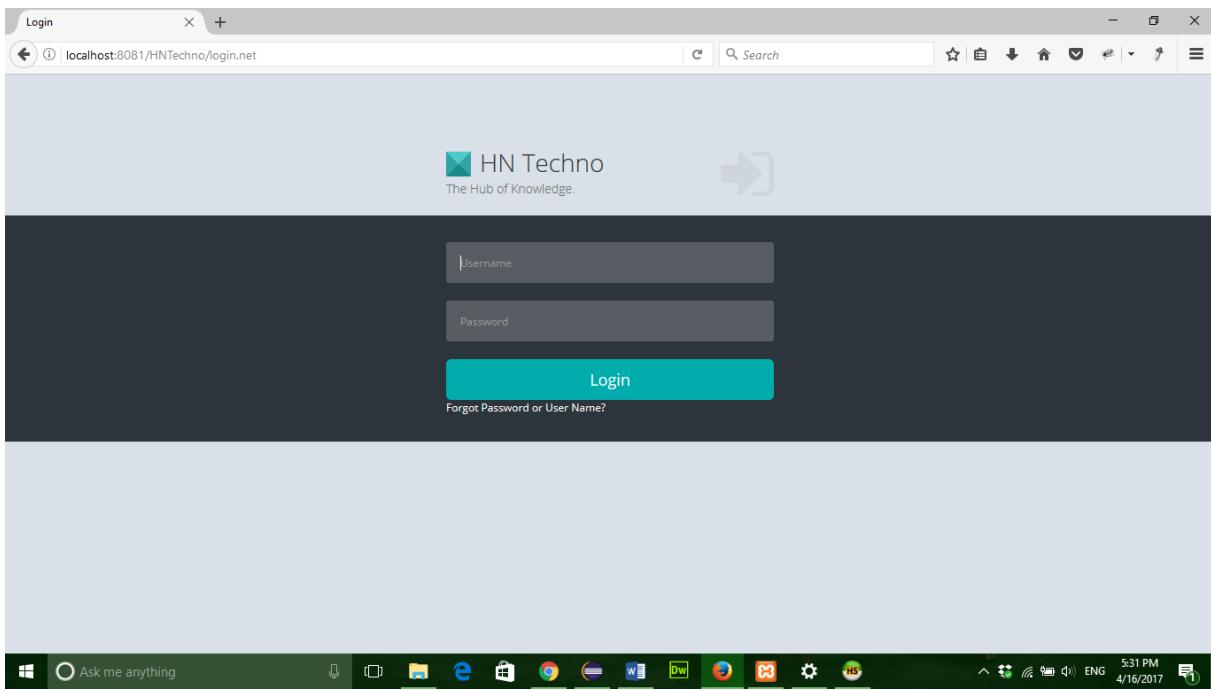


Figure 8.5 Login Page

The screenshot shows a web browser window titled 'HNTechno' with the URL 'localhost:8081/HNTechno/AddLang.net'. The page displays a table titled 'Language' with six entries. The table has columns for 'NO', 'LANGUAGE', 'TITLE', and 'ACTION'. Each row contains a set of edit icons. The entries are:

| NO | LANGUAGE | TITLE | ACTION |
|----|----------|---------------------------|--------|
| 1 | Java | Technology | |
| 2 | HTML | Hypertext Markup Language | |
| 3 | CSS | Cascading Style Sheet | |
| 4 | Php | HyperText Preprocessor | |
| 5 | C | Programming | |
| 6 | Python | Python | |

The left sidebar shows a navigation menu with items like 'Course', 'Material', 'Location', etc., under 'ROLE_ADMIN'. The status bar at the bottom shows the date and time as '4/16/2017 5:25 PM'.

Figure 8.6 Language Page

Batch Details

| NO | BATCH NAME | LANGUAGE | SUB LANGUAGE | FACULTY NAME | START DATE & END DATE | START TIME & END TIME | ACTION |
|----|------------|----------|--------------|--------------|--|-----------------------|--------|
| 1 | pqr | Java | Advance Java | xyz | 2017-04-12 00:00:00.0 to 2017-06-20 00:00:00.0 | 11:30 to 12:30 | |

Showing 1 to 1 of 1 entries

Figure 8.7 Batch Details

Student Profile

Student Basic Information

| | | |
|---|-------------------------------------|---------------------------|
| First Name ankit | Middle Name a | Last Name lad |
| Mobile Number 9724748406 | Email Id ankitlad.4267@gmail.com | Country Select Country |
| State Select State | City Select City | Address [empty] |
| Gender <input checked="" type="radio"/> Male <input type="radio"/> Female | | |

Student Information **Course Details** **Authentication** **College Information** **Project Information**

proceed

Figure 8.8 Student Registration

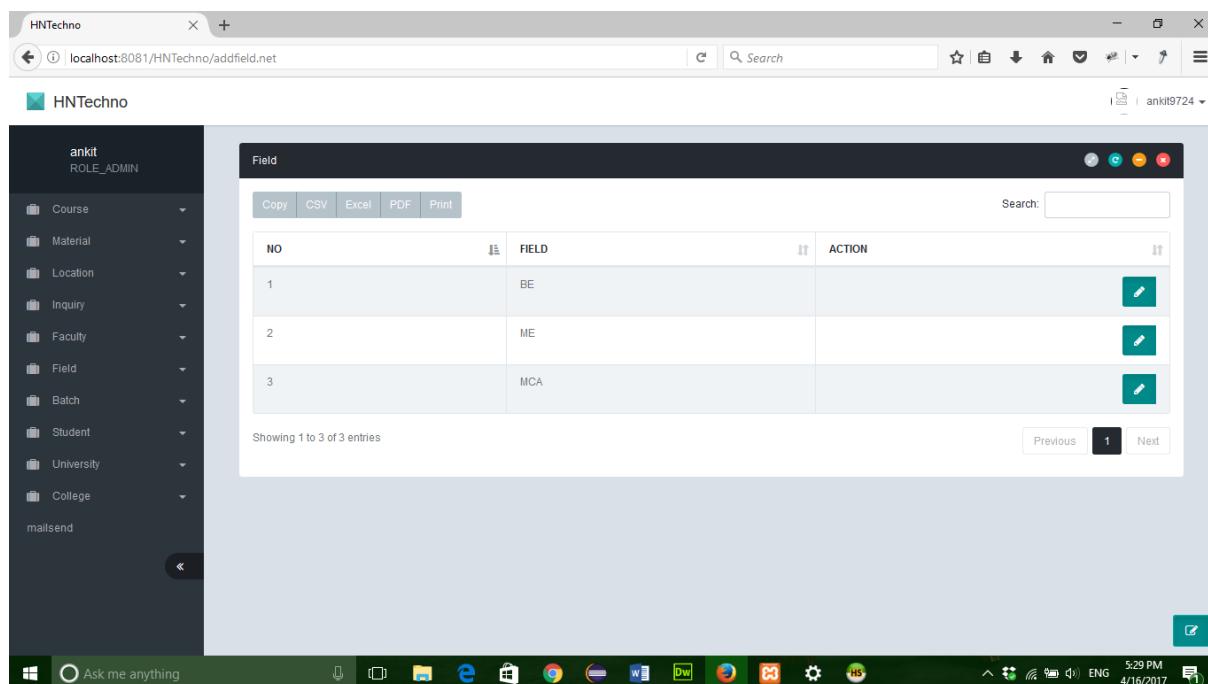


Figure 8.9 Field Page

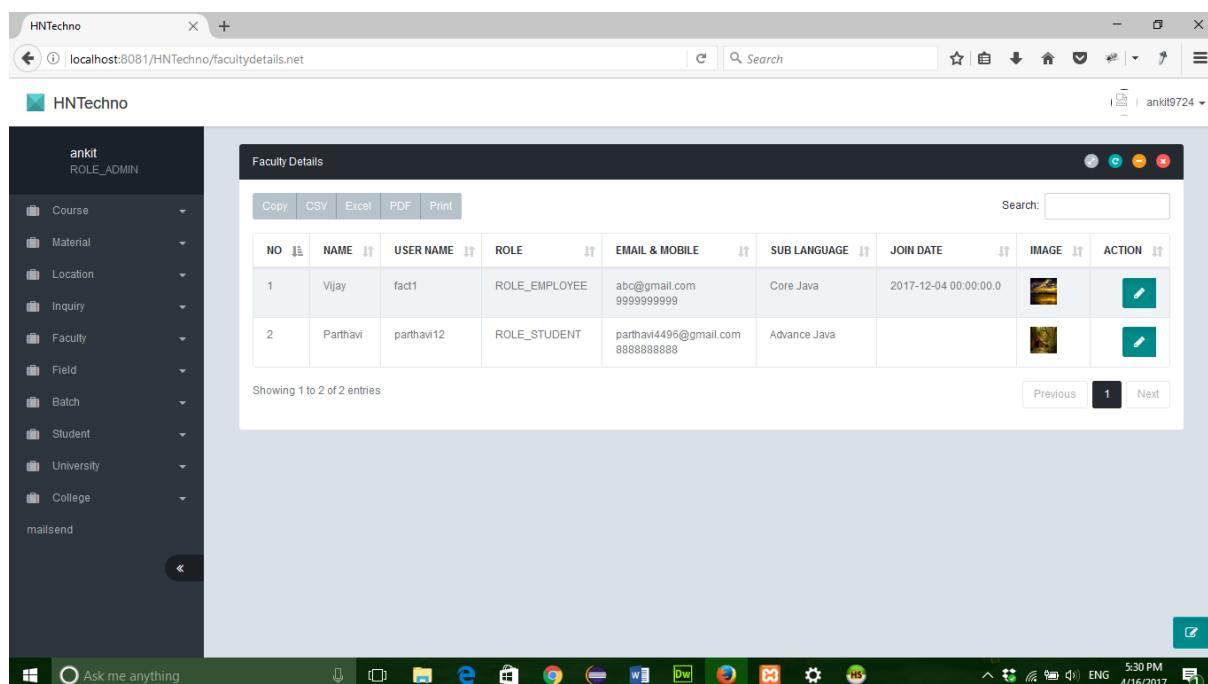


Figure 8.10 Faculty and student details

CHAPTER: 9

LIMITATION AND FUTURE ENHANCEMENT

9.1 LIMITATIONS

9.2 FUTURE ENHANCEMENTS

9.1 LIMITATION

- There is a basic need of internet to use our website.
- The smooth functionality of website mainly depends on the speed of the hardware and then on the speed of the internet.
- Some basic functionalities are given priority first and if all those are basic functionalities are implemented then advanced functionality will be served.
- Only website is under construction so right now mobile application is not made by us.

9.2 FUTURE ENHANCEMENT

After completion of some functionalities other functionalities given below will be served:

- User authorization can be done by using different ways.
- Student's profile generated based on task completion
- Compiler for task performing
- Online exams
- Attendance

CHAPTER: 10

CONCLUSION

10.1 CONCLUSION

10.1 CONCLUSION

HN Techno provides easy maintenance and easy way to handle and manage all contents as well as resources at one place. Many users connected together and get training according to their requirements and thrust to become professionals. Its provide platforms at which all are connected with each other to communicate and make a network.

REFERENCES

- <http://hntechno.com/>
- <https://www.tutorialspoint.com/>
- <http://www.javatpoint.com/>
- <https://stackoverflow.com/>
- <https://www.quora.com/>
- https://en.wikipedia.org/wiki/Face_detection
- https://en.wikipedia.org/wiki/Facial_recognition_system
- <http://www.w3schools.com/angular/default.asp>
- <http://www.w3schools.com/bootstrap/default.asp>

APPENDIX

**PERIODIC PROGRESS REPORT (PPR)
DESIGN ENGINEERING CANVAS (DE)
BUSINESS MODEL CANVAS (BMC)
PATENT DRAFTING EXERCISE (PDE)**

PPR 1

| | | | |
|------------------------|--------------------------|---------------------|---|
| Enrollment No : | 130770107085 | College : | Silver Oak College Of Engineering & Technology, Ahmedabad |
| Student Name : | Patel Ayushi Pankajbhai | Department : | Computer Engineering |
| Mobile No : | 9727402900 | Discipline : | BE |
| Email : | ayushipatel126@gmail.com | Semester : | Semester 8 |

PPR Details

Time Interval : -

Periodic Progress Report : First PPR

Project : Hntechno Followup

Status : Reviewed (Freeze)

1. What Progress you have made in the Project ?

I developed all logical concept for system, it includes all various pages like servlet, jsp, xml file to provide functionality of the system. I learnt different topics like insertion of data in MVC structure.

2. What challenge you have faced ?

I found difficulty in work flow of all pages. And managing of data in database.

3. What support you need ?

I need support for creating a structure to move data from one Servlet page to another including data security. And data must be available to Servlet pages when it is required.

4. Which literature you have referred ?

I have referred content for preparing query of insert, update and delete in database from javatpoint and other websites.

Document : [No document uploaded](#)

Comments

Comment by Internal Guide :

None

Comment by External Guide :

None

Comment by HOD :

None

Comment by Principal :

None

Comment by University Admin :

None

PPR 2

| | | | |
|------------------------|--------------------------|---------------------|---|
| Enrollment No : | 130770107085 | College : | Silver Oak College Of Engineering & Technology, Ahmedabad |
| Student Name : | Patel Ayushi Pankajbhai | Department : | Computer Engineering |
| Mobile No : | 9727402900 | Discipline : | BE |
| Email : | ayushipatel126@gmail.com | Semester : | Semester 8 |

PPR Details

Time Interval : 10 days, 11 hours

Periodic Progress Report : Second PPR

Project : Hntechno Followup

Status : Reviewed (Freeze)

1. What Progress you have made in the Project ?

I have started working on an admin panel by performing various operation such as insert, delete, update. I also managed handling of a user. I started learning concepts such as login, forgot password.

2. What challenge you have faced ?

I found difficulty in managing data as data is supposed to be handled from various pages. I also found it tricky to understand concepts of login.

3. What support you need ?

I need support for data management. I would like to know how to handle data efficiently.

4. Which literature you have referred ?

I have referred content from javatpoint and other websites and videos.

Document : No document uploaded

Comments

Comment by Internal Guide :

None

Comment by External Guide :

None

Comment by HOD :

None

Comment by Principal :

None

Comment by University Admin :

None

PPR 3

| | | | |
|------------------------|--------------------------|---------------------|---|
| Enrollment No : | 130770107085 | College : | Silver Oak College Of Engineering & Technology, Ahmedabad |
| Student Name : | Patel Ayushi Pankajbhai | Department : | Computer Engineering |
| Mobile No : | 9727402900 | Discipline : | BE |
| Email : | ayushipatel126@gmail.com | Semester : | Semester 8 |

PPR Details

Time Interval : 10 days, 14 hours

Periodic Progress Report : Third PPR

Project : Hntechno Followup

Status : Reviewed (Freeze)

1. What Progress you have made in the Project ?

I have created admin panel where user can send queries to admin of our system. I have learned to upload image, video, file. I also learned more about list which became helpful to transfer data from one page to another. I have also started working on user panel of the system.

2. What challenge you have faced ?

Various challenges faced while uploading of image, video, file. I also faced challenge while working with list and other collection elements, which are used to transfer data from one page to another.

3. What support you need ?

I need support for understanding concepts of collection which is used to manipulate data from one page to another.

4. Which literature you have referred ?

I have referred to w3schools, TutorialsPoint and JavaTpoint for various concepts of Java and Javascript.

Document : No document uploaded

Comments

Comment by Internal Guide :

None

Comment by External Guide :

None

Comment by HOD :

None

Comment by Principal :

None

Comment by University Admin :

None

PPR 4

| | | | |
|------------------------|--------------------------|---------------------|---|
| Enrollment No : | 130770107085 | College : | Silver Oak College Of Engineering & Technology, Ahmedabad |
| Student Name : | Patel Ayushi Pankajbhai | Department : | Computer Engineering |
| Mobile No : | 9727402900 | Discipline : | BE |
| Email : | ayushipatel126@gmail.com | Semester : | Semester 8 |

PPR Details

Time Interval : 11 days, 1 hours

Periodic Progress Report : Forth PPR

Project : Hntechno Followup

Status : Reviewed (Freeze)

1. What Progress you have made in the Project ?

I have completed all main modules of the system like admin, registered users, end users etc and I have completed all my systems manipulation operations which includes all insert, update and delete operation and all the logical operations are also completed which includes usage of session for user currently logged in, data transfer using cookies, session etc. I have completed all validations in my pages. I have completed all my code for login, registration, feedback, forgot password etc.

2. What challenge you have faced ?

I faced challenge while making certain pages which required complicated logic for fetching data. I faced challenge learning both JavaScript and Ajax are new and learning them is time consuming and so it was critical for me to learn JavaScript and Ajax and then implement code but I learned and completed the code with successful working code.

3. What support you need ?

I need support to adjust display of several page. I need support to solve some doubts in JavaScript, Ajax and also in validation.

4. Which literature you have referred ?

I have referred to w3schools, Tutorialspoint, JavaTpoint and other web pages from google for completion of several pages of the system for learning ajax and javascript and also for learning code of forgot password and other logical code.

Document : [No document uploaded](#)

Comments

Comment by Internal Guide :

None

Comment by External Guide :

None

Comment by HOD :

None

Comment by Principal :

None

Comment by University Admin :

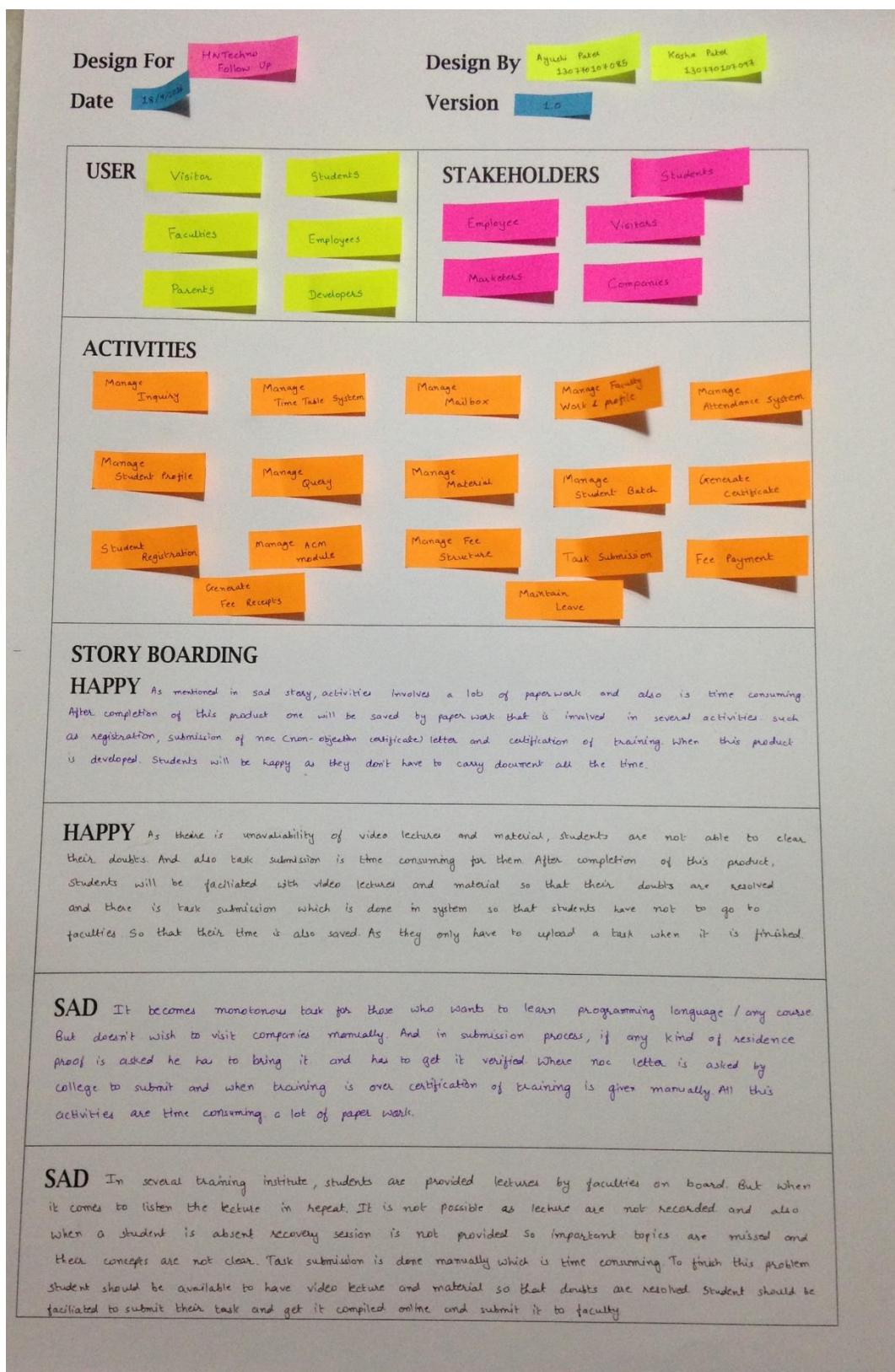
None

Design Engineering Canvas

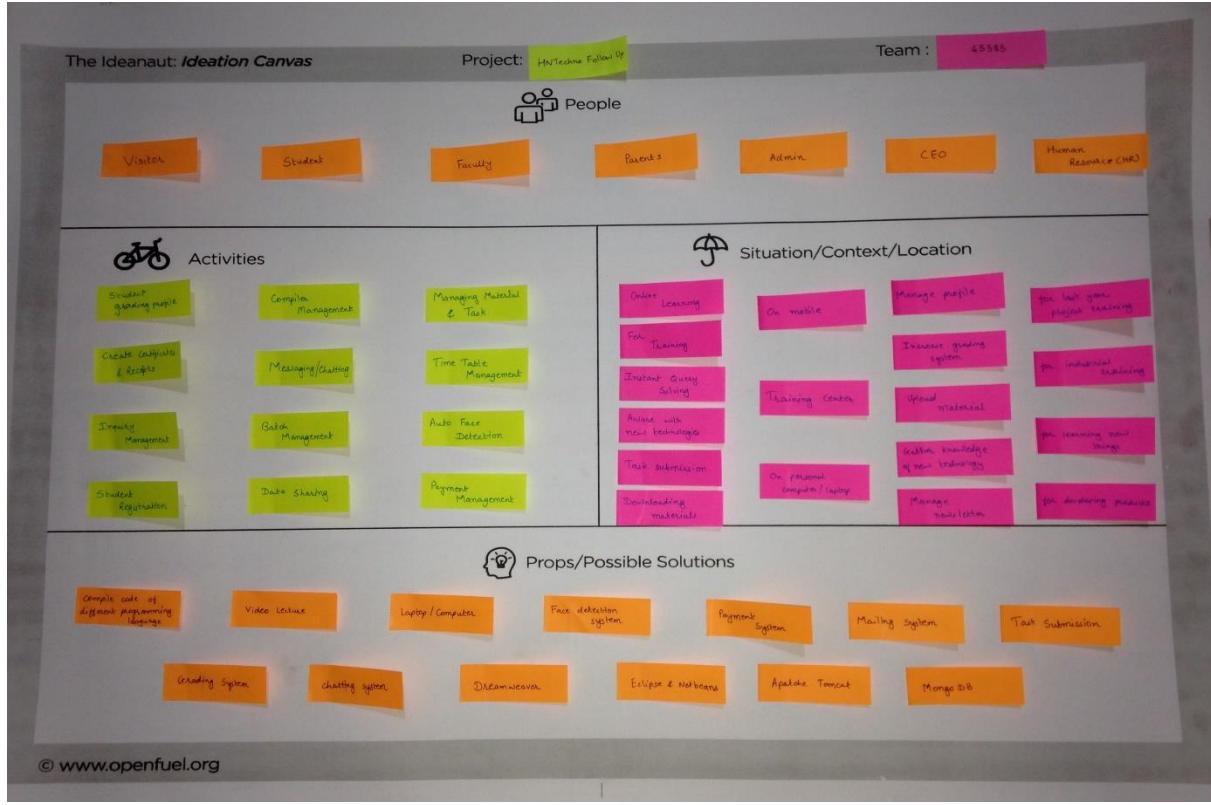
AEIOU Summary

| AEIOU Summary | | Group ID: 65595 Domain Name: Education | Date: 14/3/2016 | Version: 1.0 |
|---|--|---|-----------------|--------------|
| Environment: <ul style="list-style-type: none"> - Formal educational institutions (Classroom, Library, Auditorium, Hall, etc.) - Informal educational institutions (Seminars, Workshops, Conferences, etc.) - Social media platforms (Facebook, LinkedIn, YouTube, etc.) - Physical environment (Lighting, Temperature, Air Quality, etc.) | Interactions: <ul style="list-style-type: none"> - Students interacting with teachers, parents, and each other during lessons. - Faculty members interacting with students, parents, and each other during meetings. - Students interacting with each other during group discussions. - Faculty members interacting with students during one-on-one sessions. | Objects: <ul style="list-style-type: none"> - Laptops / Computers: Used for coding, reading, managing assignments, and taking notes. - Smart phones: Used for communication, research, and sharing data. - Notebooks: Used for writing, note-taking, and drawing. - Books: Used for reading, studying, and teaching. - Mouse: Used for navigating through documents and special notes. | | |
| Activities: <ul style="list-style-type: none"> - Teaching, learning, studying, research, inquiry, problem-solving, and assessment. - Collaboration, communication, presentation, and feedback. - Research, experiments, and practical activities. - Group discussions, presentations, and special events. | | Users: <ul style="list-style-type: none"> - Faculty: Teaching, managing, assessing, evaluating, and inquiry. - Students: Learning, submitting assignments, getting feedback, and inquiry. - Parents: Supporting, encouraging, and inquiry. - Administrators: Managing, monitoring, and inquiry. - Management: Planning, developing, and inquiry. - Developers: Coding, developing, and inquiry. - Resource Owners: Providing, maintaining, and inquiry. - Interviewees: Answering questions, providing information, and inquiry. | | |

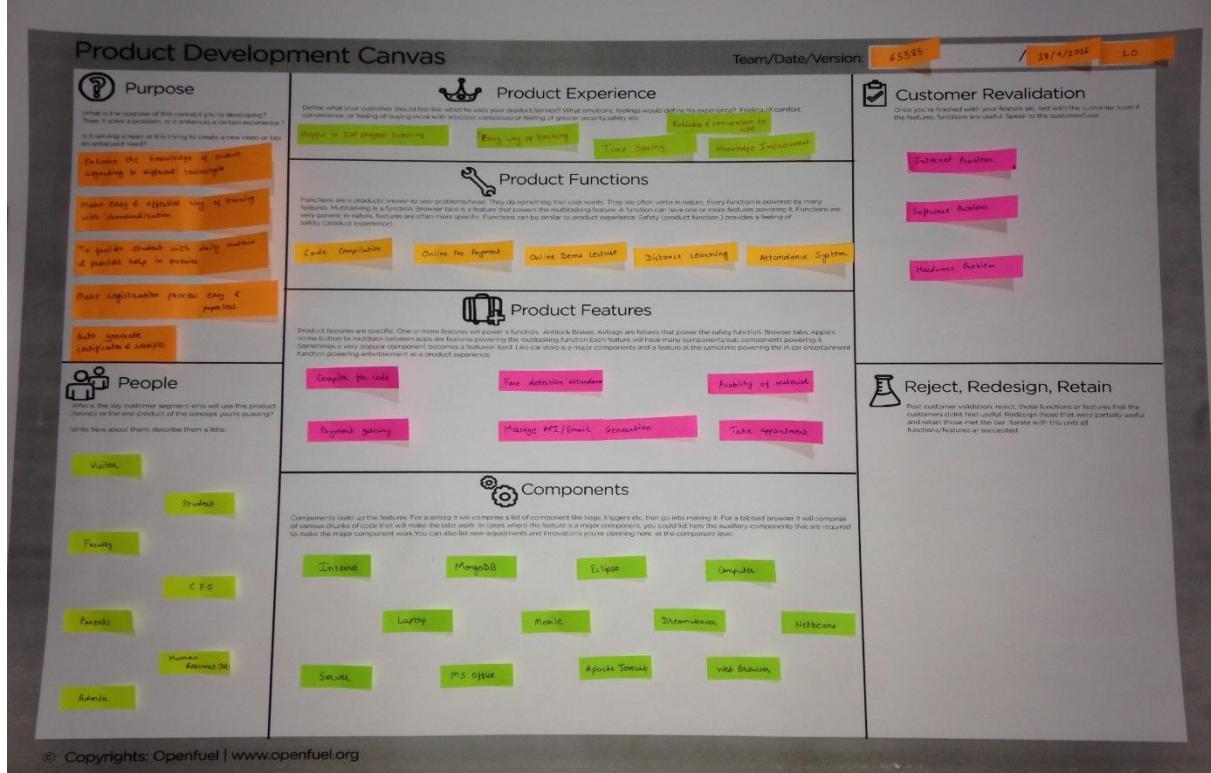
Empathy Canvas



Ideation Canvas



Product Development Canvas



BMC

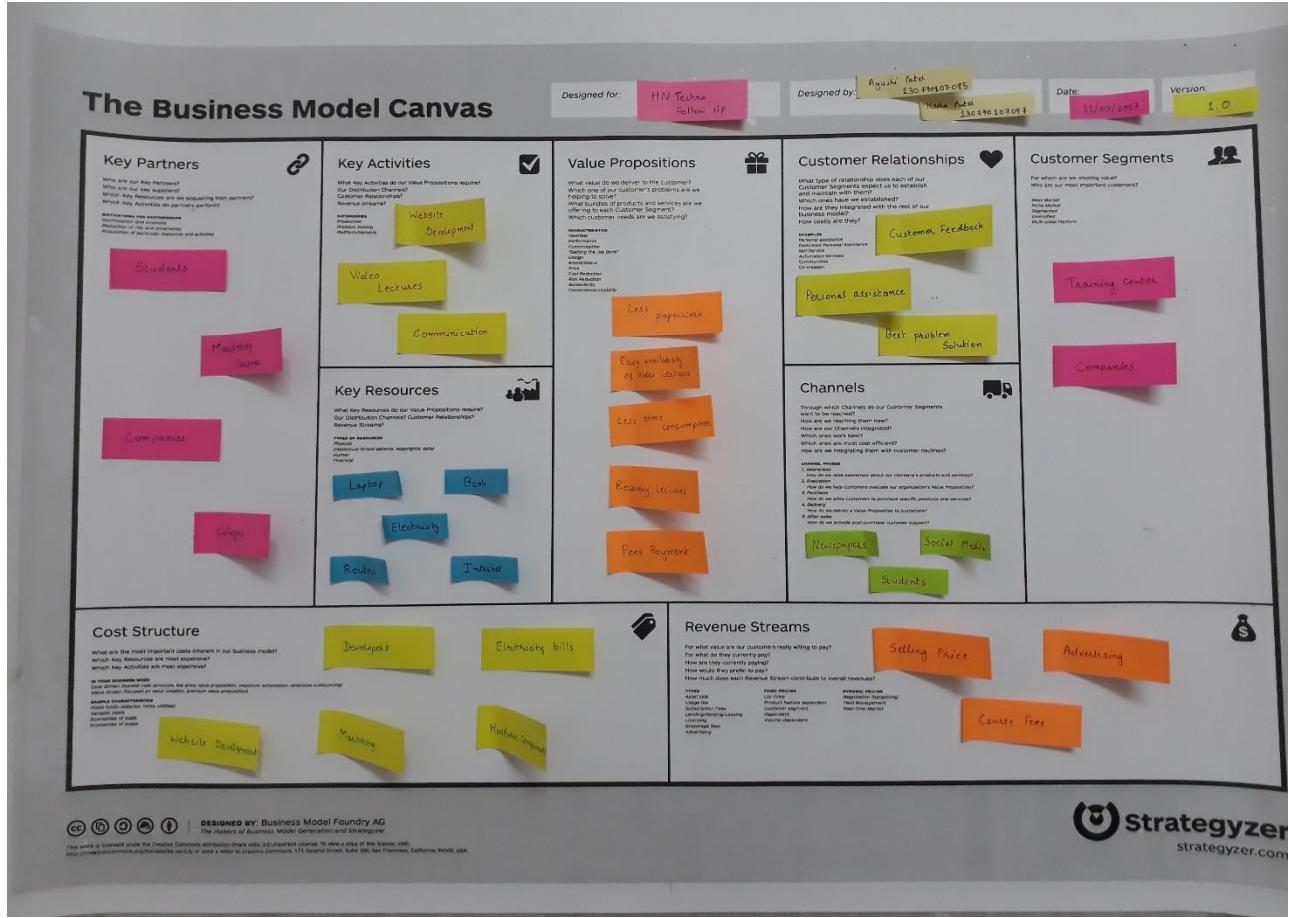
Business Model Canvas Report

Project Name
HNTechno Follow Up

Project ID: - 65585

Group Members

| Name | Enrollment No. |
|------------------------|-----------------------|
| Patel Ayushi P. | 130770107085 |
| Patel Kosha K. | 130770107097 |



1. Customer Segment

- Training Center**

Most important customer of this project is training centers.

- Students**

Another customer is students for whom this product is being made. The purpose of this product is to satisfy students.

2. Value Proposition

- Less paperwork**

There is a lot of paperwork involved in certification, registration process, fees receipts, address proof verification. We will reduce all kind of paperwork for student and company.

- Easy availability of video lecture**

Students' problem is that they have to get video of lecture via pen drive which is monotonous. We will provide video of lecture to every batch.

- Less Time Consumption**

Some basic questions that is asked has to be answered without wasting time. By developing a website that has some basic answers and responds to theirs query indeed

saves a lot time of users. And by having theirs appointment confirmed online will truly save their time.

- **Recovery lectures**

If any student is absent and wishes to attend lectures can ask for a recovery lectures.

- **Fees payment**

Students can pay their fees according to theirs convenience as fees schedule will be decided by them.

3. Channels

- **Website**

We reach to customers and Customers reach to us using website contact module.

- **Social media**

We are going to show our ads on television, newspaper, social web pages etc.

- **Students**

We can reach to more students because user of this product will promote and will recommend our company to others.

4. Customers Relationship

- **Personal assistance**

We are going to provide Personal assistance for those customers who are about to buy our product and if they need new functionalities we will provide them if it is possible.

- **Customer Feedback**

We are going to take feedback from students whether they liked the product or not.

5. Revenue Streams

- **Sell Price**

Its cost is decided by management people to sell a product including profit.

- **Advertising**

We can advertise other products.

- **Course fees**

We get revenues from course.

6. Key activities

- **Video Lectures**

This is the major activity that is done by the admin to make videos available for students.

- **Communication**

In this activity communication takes place between users and admin in which doubts can be solved, appointment can be taken and many more.

- **Website Management**

In this activity all the process related to portal management are covered.

7. Key Resources

- **Laptop**

We provide best website that can be used with laptop that provides many functionalities such as video lecture availability and task submission and many more.

- **Router**
One the part of system that connect different node to the server.
- **Internet**
One the part of system that used by customer to communicate with server and provide order via this and notify by this.
- **Bank**
If we need financial support we can get it from the bank.

8. Key Partners

- **Training center**
Training center takes decisions regarding functionalities of the system.
- **Marketing partner**
Who plays main role to advertise to the customer via different channel.
- **Students**
Who plays important role for providing constant feedback that helps us for the betterment of the product.
- **Colleges**
Who plays important role for suggesting to take training in this particular center.

9. Cost Structure

- **Electricity Bill**
It costs to electrify bill of training center.
- **Hardware Component**
It contains all nodes like laptop, router and server establishment and maintenance cost.
- **Marketing**
It costs to marketing and advertisement cost.
- **Developers**
It costs to developers for making this product.

PDE:**Form 1**

| GIC Patent Drafting Exercise | | Team ID: 72409 | | | | | | | | | | | | | | | | | | |
|--|-----------------------------|---|--|------------|--------------------------|---------|------------|-------|---|-------------------------|--------|--|------------|--------------------------|---|-----------------------------|--------|--|------------|--------------------------|
| <h1 style="text-align: center;">GTU Innovation Council</h1> <h2 style="text-align: center;">Patent Drafting Exercise (PDE)</h2> | | | | | | | | | | | | | | | | | | | | |
| FORM 1 THE PATENTS ACT 1970 (39 OF 1970) & THE PATENTS RULES, 2003 APPLICATION FOR GRANT OF PATENT | | (FOR OFFICE USE ONLY) Application No: Filing Date: Amount of Fee paid: CBR No: _____ | | | | | | | | | | | | | | | | | | |
| 1. Applicant(s) : <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th>ID</th> <th>Name</th> <th>Nationality</th> <th>Address</th> <th>Mobile No.</th> <th>Email</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>Patel Ayushi Pankajbhai</td> <td>Indian</td> <td>Computer Engineering , Silver Oak College Of Engineering & Technology, Ahmedabad , Gujarat Technological University.</td> <td>9727402900</td> <td>ayushipatel128@gmail.com</td> </tr> <tr> <td>2</td> <td>Patel Koshaben Kamleshkumar</td> <td>Indian</td> <td>Computer Engineering , Silver Oak College Of Engineering & Technology, Ahmedabad , Gujarat Technological University.</td> <td>9725586195</td> <td>koshapatel1045@gmail.com</td> </tr> </tbody> </table> | | | ID | Name | Nationality | Address | Mobile No. | Email | 1 | Patel Ayushi Pankajbhai | Indian | Computer Engineering , Silver Oak College Of Engineering & Technology, Ahmedabad , Gujarat Technological University. | 9727402900 | ayushipatel128@gmail.com | 2 | Patel Koshaben Kamleshkumar | Indian | Computer Engineering , Silver Oak College Of Engineering & Technology, Ahmedabad , Gujarat Technological University. | 9725586195 | koshapatel1045@gmail.com |
| ID | Name | Nationality | Address | Mobile No. | Email | | | | | | | | | | | | | | | |
| 1 | Patel Ayushi Pankajbhai | Indian | Computer Engineering , Silver Oak College Of Engineering & Technology, Ahmedabad , Gujarat Technological University. | 9727402900 | ayushipatel128@gmail.com | | | | | | | | | | | | | | | |
| 2 | Patel Koshaben Kamleshkumar | Indian | Computer Engineering , Silver Oak College Of Engineering & Technology, Ahmedabad , Gujarat Technological University. | 9725586195 | koshapatel1045@gmail.com | | | | | | | | | | | | | | | |
| 2. Inventor(s): <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th>ID</th> <th>Name</th> <th>Nationality</th> <th>Address</th> <th>Mobile No.</th> <th>Email</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>Patel Ayushi Pankajbhai</td> <td>Indian</td> <td>Computer Engineering , Silver Oak College Of Engineering & Technology, Ahmedabad , Gujarat Technological University.</td> <td>9727402900</td> <td>ayushipatel128@gmail.com</td> </tr> <tr> <td>2</td> <td>Patel Koshaben Kamleshkumar</td> <td>Indian</td> <td>Computer Engineering , Silver Oak College Of Engineering & Technology, Ahmedabad , Gujarat Technological University.</td> <td>9725586195</td> <td>koshapatel1045@gmail.com</td> </tr> </tbody> </table> | | | ID | Name | Nationality | Address | Mobile No. | Email | 1 | Patel Ayushi Pankajbhai | Indian | Computer Engineering , Silver Oak College Of Engineering & Technology, Ahmedabad , Gujarat Technological University. | 9727402900 | ayushipatel128@gmail.com | 2 | Patel Koshaben Kamleshkumar | Indian | Computer Engineering , Silver Oak College Of Engineering & Technology, Ahmedabad , Gujarat Technological University. | 9725586195 | koshapatel1045@gmail.com |
| ID | Name | Nationality | Address | Mobile No. | Email | | | | | | | | | | | | | | | |
| 1 | Patel Ayushi Pankajbhai | Indian | Computer Engineering , Silver Oak College Of Engineering & Technology, Ahmedabad , Gujarat Technological University. | 9727402900 | ayushipatel128@gmail.com | | | | | | | | | | | | | | | |
| 2 | Patel Koshaben Kamleshkumar | Indian | Computer Engineering , Silver Oak College Of Engineering & Technology, Ahmedabad , Gujarat Technological University. | 9725586195 | koshapatel1045@gmail.com | | | | | | | | | | | | | | | |
| 3. Title of Invention/Project: Hntechno Followup | | | | | | | | | | | | | | | | | | | | |
| 4. Address for correspondence of applicant/authorized patent agent in india <p>Name: Patel Ayushi Pankajbhai Address: Computer Engineering , Silver Oak College Of Engineering & Technology, Ahmedabad , Gujarat Technological University. Mobile: 9727402900</p> | | | | | | | | | | | | | | | | | | | | |
| Note : This is just a mock Patent Drafting Exercise (PDE) for semester 8, BE students of GTU.These documents are not to be submitted with any patent office. | | | | | | | | | | | | | | | | | | | | |

Email ID: ayushipatel128@gmail.com

5. Priority particulars of the application(S) filed in convention country

| Country | Application No. | Filing Date | Name of the Applicant | Title of the Invention |
|---------|-----------------|-------------|-----------------------|------------------------|
| N/A | N/A | N/A | N/A | N/A |

6. Particulars for filing patent co-operation treaty (pct) national phase Application

| International application number | International filing date as allotted by the receiving office |
|----------------------------------|---|
| N/A | N/A |

7. Particulars for filing divisional application

| Original(First) Application Number | Date of filing of Original (first) application |
|------------------------------------|--|
| N/A | N/A |

8. Particulars for filing patent of addition

| Original(First) Application Number | Date of filing of Original (first) application |
|------------------------------------|--|
| N/A | N/A |

9. DECLARATIONS:

(i) Declaration by the inventor(s)

I/We, the above named inventor(s) is/are true & first inventor(s) for this invention and declare that the applicant(s). herein is/are my/our assignee or legal representative.

Date : 21 - April - 2017

Name

Signature & Date

1 Patel Ayushi Pankajbhai _____

2 Patel Koshaben Kamleshkumar _____

(ii) Declaration by the applicant(s) in the convention country

I/We, the applicant (s) in the convention country declare that the applicant(s) herein is/are my/our assignee or legal representative.applicant(s)

(iii) Declaration by the applicant(s)

I/We, the applicant(s) hereby declare(s) that:-



I am/We in possession of the above mentioned invention.



The provisional/complete specification relating to the invention is filed with this application.



The invention as disclosed in the specification uses the biological material from India and the necessary permission from the competent authority shall be submitted by me/us before the grant of patent to me/us.



There is no lawful ground of objection to the grant of the patent to me/us.

Note : This is just a mock Patent Drafting Exercise (PDE) for semester 8, BE students of GTU.These documents are not to be submitted with any patent office.

Page 2

I am/we are the assignee or the legal representative of true & first inventors.

The application or each of the application/particulars of each are given in the para 5 was the first application in the convention country/countries in respect of my/our invention.

I/we claim the priority from the above mentioned applications(s) filed in the convention country/countries & state that no application for protection in respect of invention had been made in a convention country before that date by me/us or by any person.

My/Our application in India is based on international application under Patent Cooperation Treaty (PCT) as mentioned in para 6.

The application is divided out of my/our application(s) particulars of which are given in para 7 and pray that this application may be treated as deemed to have been filed on _____ under section 16 of the Act.

The said invention is an improvement in or modification of the invention particulars of which are given in para 8.

10. Following are the attachments with the application:

(a) Provisional specification/Complete specification

(b) Complete specification (In confirmation with the international application) / as amended before the International Preliminary Examination Authority (IPEA), as applicable (2 copies). No. of pages.....No. of claims.....

(c) Drawings (In confirmation with the international application)/as amended before the International Preliminary Examination Authority(IPEA),as applicable(2 copies),No.of sheets....

(d) Priority documents

(e) Translations of priority documents/specification/international search reports

(f) Statement and undertaking on Form 3

(g) Power of Authority

(h) Declaration of inventorship on Form 5

(i) Sequence listing in electronic Form

(j) Fees Rs.XXX in Cash /Cheque/Bank Draft bearing No.XXX Date: XXX on XXX Bank.

I/We hereby declare that to the best of my /our knowledge, information and belief the fact and matters stated herein are correct and I/We request that a patent may be granted to me/us for the said invention.
Dated this 21 day of April , 2017

| | |
|-------------------------------|------------------|
| Name | Signature & Date |
| 1 Patel Ayushi Pankajbhai | _____ लान् |
| 2 Patel Koshaben Kamleshkumar | _____ |

Note : This is just a mock Patent Drafting Exercise (PDE) for semester 8, BE students of GTU. These documents are not to be submitted with any patent office.

Page 3

Form 2

| | | |
|---|--|----------------|
| GIC Patent Drafting Exercise | | Team ID: 72409 |
| FORM 2 THE PATENTS ACT, 1970 (39 OF 1970) & THE PATENTS RULES, 2003 PROVISIONAL SPECIFICATION | | |
| <p>1. Title of the project/invention : Hitechno Followup</p> <p>2. Applicant(s) :</p> <p>1) (a) NAME : Patel Ayushi Pankajbhai (b) NATIONALITY : Indian (c) ADDRESS : Computer Engineering , Silver Oak College Of Engineering & Technology, Ahmedabad , Gujarat Technological University.</p> <p>2) (a) NAME : Patel Koshaben Kamleshkumar (b) NATIONALITY : Indian (c) ADDRESS : Computer Engineering , Silver Oak College Of Engineering & Technology, Ahmedabad , Gujarat Technological University.</p> <p>3. Preamble to the description : The following specification describes the invention.</p> | | |
| <p>Note : This is just a mock Patent Drafting Exercise (PDE) for semester 8, BE students of GTU.These documents are not to be submitted with any patent office.</p> <p style="text-align: right;">Page 1</p> | | |

4. Description :**a. Field of Application / Project / Invention :**

This web application is for an IT company that provides functionalities such as asking for an appointment, downloading of video lectures, fees schedule, doubt solving, receipt generation, profile generation, certificate generation, recovery session etc.

b. Prior Art / Background of the Invention / References :

Generally system includes some activities such as certificate generation, receipt generation a lot of paper work which is difficult to manage for each and every student. Unavailability of video lectures is caused to some students.

c. Summary of the Invention/Project :

This invention solves students facing problem in getting appointment, video lectures, doubt solving, certificates etc.

d. Objects of the Invention/Project :

This system can be used to solve problems of users such as visiting, asking for an appointment to resolve doubts in their mind. Any user who wishes to visit company website can get basic information, demo lecture, demo material, faculty list. User gets himself registered into the system to interact with it.

e. Drawing(s) :**f. Description of the Invention :**

The project is developed to create convenient and easy way to use system for users of the system, students, and faculties. Our web application provides basic information to user, to facilitate them to get an appointment, to provide them list of faculties, to get payment done, to make registration process easy. Student will be facilitated to request for leave, to view fees schedule, to chat with other users, to download material and videos, to submit a task, to get certificate.

g. Examples :**h. Unique Features of the Project :**

Availability of video lectures, fees scheduling, receipt generation, profile generation, certificate generation, recovery session.

5. Date & Signature :

Date :21 - April - 2017

Sign and Date
Patel Ayushi Pankajbhai

Sign and Date
Patel Koshaben
Kamaleshkumar

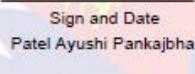
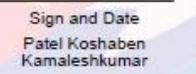
6. Abstract of the project / invention :

This web application of HNTechno Company will include basic and advanced functionality any IT company can desire for. There are features like viewing demo lectures, viewing faculty list, taking appointment, task submission, attendance, getting certificate, downloading material etc. in the system. The main aim of our project is to create convenient and easy to use system for users of the system, students, and faculties. User will register themselves in HNTechno to get involved in the system. Student will be able to download lectures and submitting task and getting certificate in the end. System will be developed in such a way that it is useful and easy in use.

Note : This is just a mock Patent Drafting Exercise (PDE) for semester 8, BE students of GTU. These documents are not to be submitted with any patent office.

Page 2

Form 3

| GIC Patent Drafting Exercise | | Team ID: 72409 | | | | | | | | | | | | | | | |
|---|---------------------|--------------------|---------------------------|---------------------|---------------|---------------------|---------------------|--------------------|---------------------------|---------------------|---------------|-----|-----|-----|-----|-----|-----|
| FORM 3 THE PATENTS ACT, 1970 (39 OF 1970) & THE PATENTS RULES, 2003 STATEMENT AND UNDERTAKING UNDER SECTION 8 | | | | | | | | | | | | | | | | | |
| <p>1. Declaration :</p> <p>I/We, Patel Ayushi Pankajbhai , Patel Koshaben Kamleshkumar</p> <p>2. Name, Address and Nationality of the joint Applicant :</p> <p>Patel Ayushi Pankajbhai (Indian) Address :Computer Engineering , Silver Oak College Of Engineering & Technology, Ahmedabad , Gujarat Technological University.</p> <p>Patel Koshaben Kamleshkumar (Indian) Address :Computer Engineering , Silver Oak College Of Engineering & Technology, Ahmedabad , Gujarat Technological University.</p> <p>Here by declare :</p> <p>(i) that I/We have not made any application for the same/substantially the same invention outside India. (ii) that the right in the application(s) has/have been assigned to,</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th>Name of the Country</th> <th>Date of Application</th> <th>Application Number</th> <th>Status of the Application</th> <th>Date of Publication</th> <th>Date of Grant</th> </tr> </thead> <tbody> <tr> <td>N/A</td> <td>N/A</td> <td>N/A</td> <td>N/A</td> <td>N/A</td> <td>N/A</td> </tr> </tbody> </table> <p>(iii) that I/We undertake that up to the date of grant of patent by the Controller , I/We would keep him inform in writing the details regarding corresponding application(s) for patents filed outside India within 3 months from the date of filing of such application.</p> <p>Dated this <u>21</u> day of <u>April</u> , <u>2017</u></p> <p>3. Signature of Applicants :</p> <div style="display: flex; justify-content: space-around; align-items: flex-end;"> <div style="text-align: center;">  <p>Sign and Date Patel Ayushi Pankajbhai</p> </div> <div style="text-align: center;">  <p>Sign and Date Patel Koshaben Kamleshkumar</p> </div> </div> <p>To The Controller of Patent The Patent Office, at Mumbai.</p> | | | | | | Name of the Country | Date of Application | Application Number | Status of the Application | Date of Publication | Date of Grant | N/A | N/A | N/A | N/A | N/A | N/A |
| Name of the Country | Date of Application | Application Number | Status of the Application | Date of Publication | Date of Grant | | | | | | | | | | | | |
| N/A | N/A | N/A | N/A | N/A | N/A | | | | | | | | | | | | |
| <p>Note : This is just a mock Patent Drafting Exercise (PDE) for semester 8, BE students of GTU.These documents are not to be submitted with any patent office.</p> | | | | | | Page 1 | | | | | | | | | | | |