Hiring Buddy

A PROJECT REPORT

Submitted by

JOSHI MAYANK [19BECE30109]

HADVANI LAV [19BECE30102]

KARAVADRA ANKITA [19BECE30014]

ODEDRA BHAVISHA [19BECE30507]

In fulfilment of the award of the degree

of

BACHELOR OF ENGINEERING

in

COMPUTER ENGINEERING

LDRP Institute of Technology and Research, Gandhinagar

Kadi Sarva Vishwavidyalaya

April 2022-23

LDRP INSTITUTE OF TECHNOLOGY AND RESEARCH GANDHINAGAR

CE Department

CERTIFICATE

This is to certify that the Project Work entitled **"HIRING BUDDY"** has been carried out by **JOSHI MAYANK (19BECE30109)** under my guidance in fulfilling the degree of Bachelor of Engineering in Computer Engineering Semester-8 of Kadi Sarva Vishwavidyalaya University during the academic year 2022-2023.

Prof. Vishal Nayak	Dr. Sandip Modha
Internal Guide	**Head of the Department**
LDRP ITR	**LDRP ITR**

LDRP INSTITUTE OF TECHNOLOGY AND RESEARCH GANDHINAGAR

CE Department

CERTIFICATE

This is to certify that the Project Work entitled **"HIRING BUDDY"** has been carried out by **HADVANI LAV (19BECE30102)** under my guidance in fulfilling the degree of Bachelor of Engineering in Computer Engineering Semester-8 of Kadi Sarva Vishwavidyalaya University during the academic year 2022-2023.

Prof. Vishal Nayak	Dr. Sandip Modha
Internal Guide	**Head of the Department**
LDRP ITR	**LDRP ITR**

LDRP INSTITUTE OF TECHNOLOGY AND RESEARCH GANDHINAGAR

CE Department

CERTIFICATE

This is to certify that the Project Work entitled **"HIRING BUDDY"** has been carried out by **ODEDRA BHAVISHA (19BECE30507)** under my guidance in fulfilling the degree of Bachelor of Engineering in Computer Engineering Semester-8 of Kadi Sarva Vishwavidyalaya University during the academic year 2022-2023.

Prof. Vishal Nayak	Dr. Sandip Modha
Internal Guide	**Head of the Department**
LDRP ITR	**LDRP ITR**

LDRP INSTITUTE OF TECHNOLOGY AND RESEARCH GANDHINAGAR

CE Department

CERTIFICATE

This is to certify that the Project Work entitled **"HIRING BUDDY"** has been carried out by **KARAVADRA ANKITA (19BECE30014)** under my guidance in fulfilling the degree of Bachelor of Engineering in Computer Engineering Semester-8 of Kadi Sarva Vishwavidyalaya University during the academic year 2022-2023.

Prof. Vishal Nayak	Dr. Sandip Modha
Internal Guide	**Head of the Department**
LDRP ITR	**LDRP ITR**

Presentation-I for Project-III

1. Name & Signature of Internal Guide	
2. Comments from Panel Members	
3. Name & Signature of Panel Members	

Presentation-2 for Project-III

1. Name & Signature of Internal Guide	
2. Comments from Panel Members	
3. Name & Signature of Panel Members	

Acknowledgement

We take this opportunity to express our profound gratitude and deep regard to our guide for his exemplary guidance, monitoring and constant encouragement throughout the course of this project. The blessing, help and guidance given by his time to time shall carry us a long way in the journey of life on which we are about to embark. We also take this opportunity to express a deep sense of gratitude to the university for cordial support, valuable information and guidance, which help us in completing this task throughout the various stages.

Mayank Joshi (19BECE30109)

Lav Hadvani (19BECE30102)

Ankita Karavadra (19BECE30014)

Bhavisha Odedra (19BECE30507)

ABSTRACT

Hiring Buddy aimed at developing an application for the “PLACEMENT MANAGEMENT SYSTEM” of the college. The system is an application that can be accessed and effectively used throughout the organization with proper login enabled. This system can be used as an application for the Placement Officers in the college to manage student information with regard to placement. Student logging should be able to upload their personal and educational information in the form of a resume. The key feature of this project is that it is one-time registration enabled. Our project provides the facility of maintaining the details of the students. It reduces manual work and consumes less paperwork to reduce time. The Placement Management System is developed as an attempt to take a record of the company and students by restricting such a large database to that of a particular class of students or company.

List Of Figure

List Of Tables

Table Of Contents

1. INTRODUCTION

1.1 Introduction:

- The software system is a placement management system that helps in managing students' data in the college with regard to placement.
- It intends to help fast in fast access procedures in placement-related activities and ensures to maintain the details of the student.
- Students logging in should be able to upload their personal and educational information.
- The key feature of this project is that it is one-time registration enabled.
- The placement cell calls the companies to select their students for jobs via the campus interview and also allows the companies to view the student resumes in a selective manner.
- They can filter the student's profiles as per their requirement and job details of the placed students will be provided by the administrator.
- Our project provides the facility of maintaining the details of the students and gets the requested list of candidates for the company who would like to recruit the students based on the given query.

1.2 Scope:

Our project has a big scope to do:

- Students can access previous information about placement.
- We can stores information about all students.
- Various companies can access their information.
- Notifications are sent to students about the companies.
- Students can maintain their information and can update it.

1.3 Project summary and Purpose:

Project summary:

- Placement Management System manages student information in the college with regard to placement.
- It improves existing system.
- It has the facility of maintaining the details of the student, thereby reducing the manual work.
- It will save time and energy which are spending in making reports and collecting data.

Project Purpose:

- The purpose of the project is to build an application program to reduce the manual work for managing the data of students for placement.
- For the purpose of placement of the student in colleges, The placement officer has to collect the information and resume of students and manages them manually and arranges them according to various streams.
- If any modification is required that is to be also done manually.
- To reduce the job required to manage resume and the information of various recruiters, a new system is proposed which is processed through computers.

1.4 Overview of the project:

- This system is developed in a manner that is easily manageable, time saving and reliving one from manual work.
- The requirements and various posts of the company has to be updated and the placement admin cell send the mail to the students who are under requirement.
- The placement officer can update the status of students whether those are getting a job or not.

1.5 Problem definition:

- This project is aimed at developing an online application for the Placement Dept. of the college.
- The system is an online application that can be accessed throughout the organization and outside as well with proper login provided.
- This system can be used as an application for the Placement Officer of the college to manage the student information with regards to placement.
- Students logging should be able to upload their information in the form of a CV.
- Company representatives logging in may also access/search any information put up by Students.

2. TECHNOLOGY AND LITERATURE REVIEW

2.1 Tools and Technology:

2.1.1 PHP XAMPP:

XAMPP is a software distribution which provides the Apache web server, MySQL database (actually MariaDB), Php and Perl all in one package.

It is available for Windows, MAC and Linux systems. No configuration is necessary to integrate Php with MySQL.

It is a great fit for this course and provides a relatively painless installation and way to manage the configuration changes. Also provided is PhpMyadmin which gives a GUI tool for managing your MySQL databases.

I would highly recommend installing this for Windows or MAC. It doesn't exclude you from other competing software installations, it just gives an easy way to get going. For Ubuntu Linux systems, I still would recommend installing Apache/MySQL/Php/PhpMyadmin through Ubuntu packages.

2.1.1 BOOTSTRAP:

Our project uses bootstrap for the frontend as Bootstrap is a free and open-source CSS framework directed at responsive, mobile-first front-end web development. It contains CSS- and JavaScript-based design templates for typography, forms, buttons, navigation, and other interface components.

Why we are using bootstrap?

One framework, Every device:

Bootstrap easily and efficiently scales your websites and applications with a single code base, from phones to tablets to desktops with CSS media queries

Pre-processor:

Bootstrap ships with vanilla CSS, but its source code utilize the CSS pre-processors, less and Sass. Quickly get started with precompiled CSS or build on the source.

Full of features:

With Bootstrap, you get extensive and beautiful documentation for common HTML elements, dozens of custom HTML and CSS components, and awesome jQuery plugins.

2.1.2 Oracle Database:

We are going to use oracle database to save all the product and its detail and other data.
As Oracle Database 19c offers market-leading performance, scalability, reliability, and security both on-premises and in the cloud. And since it's the Long Term Release of the Oracle Database 12c and 18c family of products, it provides the highest level of release stability and longest time frame for support and bug fixes.

Why we are using oracle database?

Mixed workloads with one database:

Free application developers from complex transformations and redundant data with a single-purpose database.

Increase database security:

Secure all your data and avoid the fragmentation and risk common with single-purpose databases.

Simplify database management:

Easily manage, tune, and back up your data to ensure the best overall performance and availability.

2.2 Brief history of work done:

2.2.1 Project Development Approach:

The model that is referred for the development of the project is INCREMENTAL model. It combines elements of the waterfall model applied in an iterative fashion. In this process the phases are same as waterfall but the advantage is that when first phase is done it is incremented and then the other phases are carried with the same cycle. Here in this add ones on each phase can be added according to the need of the client and the project.
Phases are as follows:

- Communication
- Planning
- Modelling: Includes Designing
- Construction
- Deployment: Feedback, Delivery

Each phase is iteratively carried out. The main reason for using this than any other is waterfall has the drawback of iterations, if there is any other requirement added later on then this is not possible to add up in it, Spiral model has the disadvantage that it needs more manpower and even it is for multiple transactions or multiple tasks handling projects and so does the time consumption is more in it for those projects.

Planning is essential because multiple software teams work in parallel on different system functions. Scalability should be obtained in any of the projects selected but it is not available in waterfall cause of a few drawbacks.

2.2.2 Milestones and Deliverables:

- **Month 1: Milestones & Deliverables:**

Milestones	Deliverables
Study about our web application requirement, planning	Analysis Report
Understand project definitions and basic terms and logic for Parameter Evaluation.	
Gathering the requirements of the project using different fact-finding techniques.	Analysis Report
Still Continue with Requirement's study.	

- **Month 2-4: Milestones & Deliverables:**

Milestones	Deliverables
System Analysis	Analysis Report
System Design including various diagrams	SRS

- **Month 5-7: Milestones & Deliverables:**

Milestones	Deliverables
Integrating techniques of bootstrap and xampp	Designing/Coding
Database creation and Procedures	Designing/Coding
Admin Module of Client Support System	Designing/Coding
Consultant Module of Client Support System	Designing/Coding
Accountant Module of Client Support System	Designing/Coding

- **Month 8: Milestones & Deliverables:**

Milestones	Deliverables
Client Module of Client Support System	Designing/Coding
Website Testing	Testing
Required changes after testing	Designing/Coding

2.3 Cost Estimation:

Several models are used to find the estimate costing of the software but we are using the well-known model known as COCOMO model.
Cocomo (Constructive Cost Model) is a regression model based on LOC, i.e. number of Lines of Code. It is a procedural cost estimate model for software projects and often used as a process of reliably predicting the various parameters associated with making a project such as size, effort, cost, time and quality. The key parameters which define the quality of any software products, which are also an outcome of the Cocomo are primarily Effort & Schedule:

- **Effort:** Amount of labour that will be required to complete a task. It is measured in person- months units.
- **Schedule:** Simply means the amount of time required for the completion of the job, which is, of course, proportional to the effort put in. It is measured in the units of time such as weeks, and months.

Different models of Cocomo have been proposed to predict the cost estimation at different levels, based on the amount of accuracy and correctness required. All of these models can be applied to a variety of projects, whose characteristics determine the value of the constant to be used in subsequent calculations. These characteristics pertaining to different system types are mentioned below.

- Organic
- Semi-detached
- Embedded

SOFTWARE PROJECTS	A	B	C	D
Organic	2.4	1.05	2.5	0.38
Semi Detached	3.0	1.12	2.5	0.35
Embedded	3.6	1.20	2.5	0.32

Since we are unfamiliar with some aspects of the system like server handling and some more aspects so out project come under semi-detached model. Therefore,

Effort = a*(KLOC) ^b PM
= (3.0)*(12.500) ^ (1.12)
= 50.78 person-month approx.

Schedule = c*(Effort) ^d
= 2.5*(50.78) ^0.35
= 9.80 months approx.

Here as we see 5months estimate time is taken by the team of 2 people to make this website if we take average salary of each person as 20,000rs/month hence the website will cost 2*20,000*5 = 2,00,000rs approx. where Transportation, hardware and other cost have not been included.

3. System Requirements Study

3.1 User Characteristics:

Analysing user characteristics is an important aspect of any project. It allows us to clearly define and focus on who the end users are for the project. Also, it allows checking the progress of the project to ensure that we are still developing the system for the end users.

The user must have the following characteristics:

- User must have basic knowledge of Computers.
- User should understand the use of all modules.
- User can easily interact with the proposed system.
- User must know the technical terms used in the company for performing different tasks especially related to call logs, payment details, transportation details and report retrieval.
- User should be also being aware of the running process of the system.

3.2 Software and Hardware Requirements and Recommendation:

3.2.1 Software Requirements:

- Language HTML, PHP, CSS, BOOTSTRAP, JAVASCRIPT
- Software VISUAL STUDIO, XAMPP SERVER
- database ORACLE, MYSQL, PHPMYADMIN
- windows edition windows 8 enterprise
- Processor Intel(R) core(TM) i5 – 6200U CPU @ 2.30GHz
- System type 64 bit operating system

3.2.2 Hardware Requirements:

- Operating system Windows 8 64-bit, Windows 7 Service Pack 1 64-bit, Windows Vista Service Pack 2 64-bit
- CPU Core 2 Quad Q6600 at 2.4 GHz or AMD Phenom 9850 at 2.5 GHz
- Memory 2GB RAM
- Free space 65 GB of free space
- Graphics hardware DirectX 10-compatible GPU: GeForce 9800GT 1GB or ATI Radeon HD 4870 1GB
- Sound hardware DirectX 10 compatible sound card

3.3 Assumptions and Dependencies:

3.3.1 Assumptions:

- Database transactions are assumed to be secure and reliable.
- User is the person having enough knowledge for the traversing operation.
- We will provide a user-friendly interface so that any user can easily navigate through the system, but he/she should be capable of providing valid credentials for successful login.
- The server used for data storage is always secured.

3.3.2 Dependencies:

- The system is dependent upon the user's valid credentials. If user inputs wrong username or password, he/she will not be allowed to login to the system.
- This application depends on the server and internet as all the information is collected and then stored in the server through secure internet connection.
- All the users of the system will be assigned a specific role. According to these roles each and every user will be allowed to access predefined set of features.

4. SYSTEM ANALYSIS

4.1 Class Diagram:

- A class diagram is a static diagram. It represents the static view of an application. The class diagram is not only used for visualizing, describing, and documenting different aspects of a system but also for constructing executable code of the software application.

- A class diagram describes the attributes and operations of a class and also the constraints imposed on the system. Class diagrams are widely used in the modelling of object-oriented systems because they are the only UML diagrams, which can be mapped directly with object-oriented languages.

- The class diagram shows a collection of classes, interfaces, associations, collaborations, and constraints. It is also known as a structural diagram.

- The purpose of the class diagram can be summarized as follow;

  - Analysis and design of the static view of an application.
  - Describe the responsibilities of a system.
  - The base for component and deployment diagrams.
  - Forward and reverse engineering.

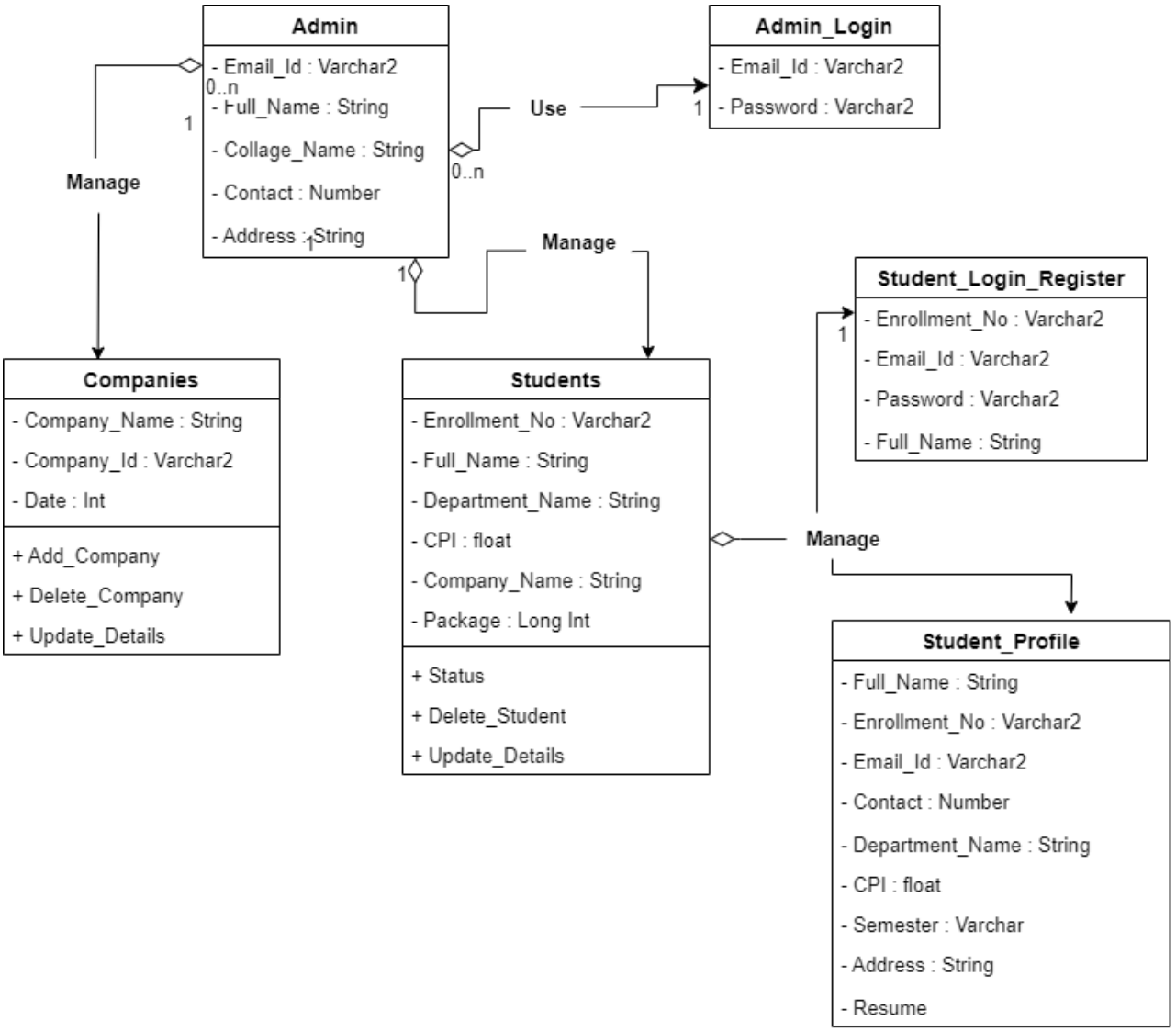


Figure 4.1. Class Diagram

4.2 Use Case Diagram:

- To model a system, the most important aspect is to capture the dynamic behaviour. Dynamic behaviour means the behaviour of the system when it is running/operating.

- Only static behaviour is not sufficient to model a system rather dynamic behaviour is more important than static behaviour. In UML, there are five diagrams available to model the dynamic nature and a use case diagram is one of them. Now as we have to discuss that the use case diagram is dynamic in nature, there should be some internal or external factors for making the interaction.

- These internal and external agents are known as actors. Use case diagrams consist of actors, use cases and their relationships. The diagram is used to model the system/subsystem of an application. A single-use case diagram captures a particular functionality of a system.

- In brief, the purposes of use case diagrams can be said to be as follows −

• Used to gather the requirements of a system.

• Used to get an outside view of a system.

• Identify the external and internal factors influencing the system.

• Show the interaction among the requirements actors.

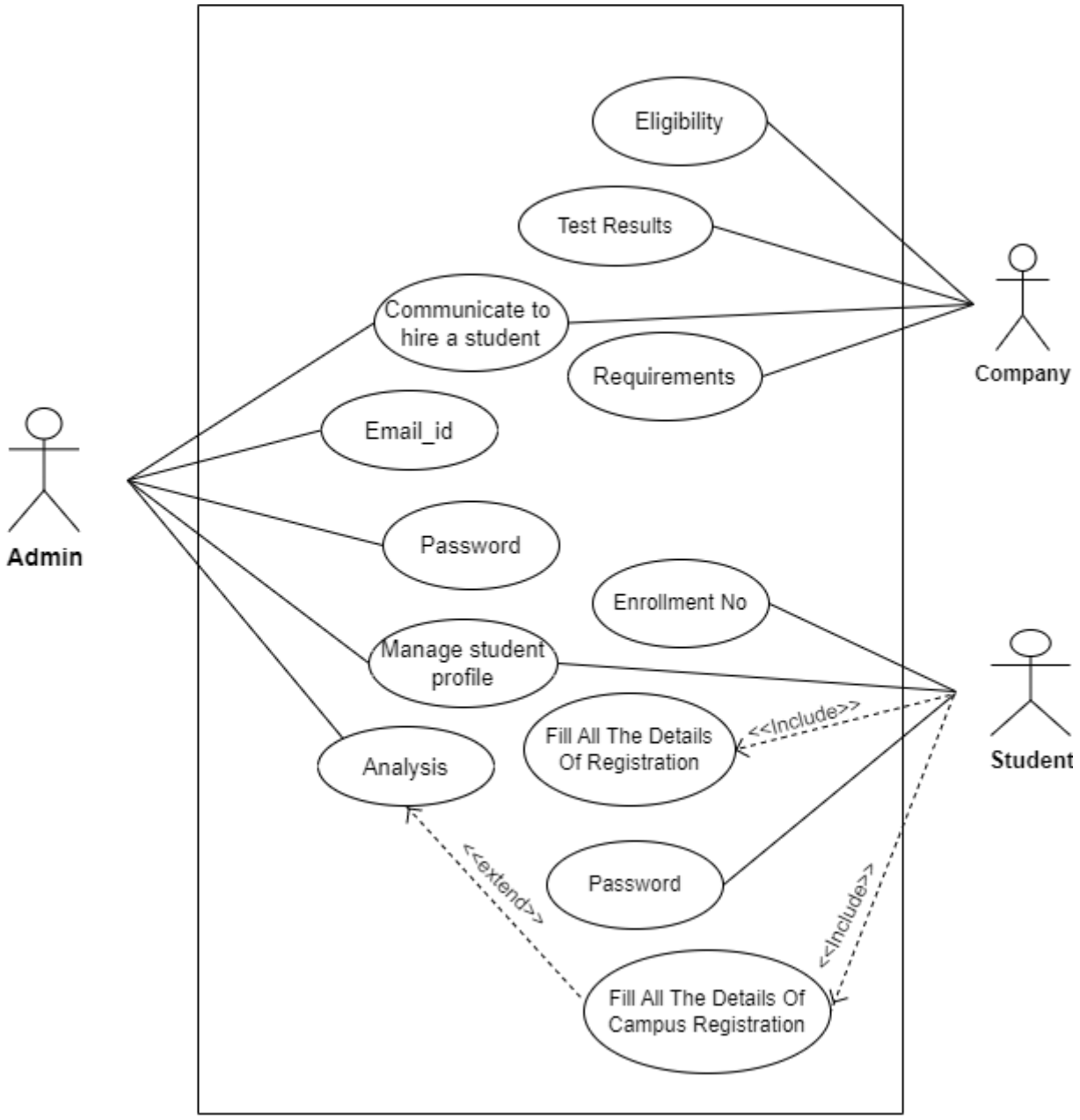


Figure 4.2. Use Case Diagram

4.3 Sequence Diagram:

- UML Sequence Diagrams are interaction diagrams that detail how operations are carried out. They capture the interaction between objects in the context of a collaboration.

- Sequence Diagrams are time focus and they show the order of the interaction visually by using the vertical axis of the diagram to represent the time when messages are sent and when.

- Sequence Diagrams captures:

- The interaction that takes place in a collaboration that either realizes a use case or an operation (instance diagrams or generic diagrams)

- High-level interactions between the user of the system and the system, between the system and other systems or between subsystems (sometimes known as system sequence diagrams)

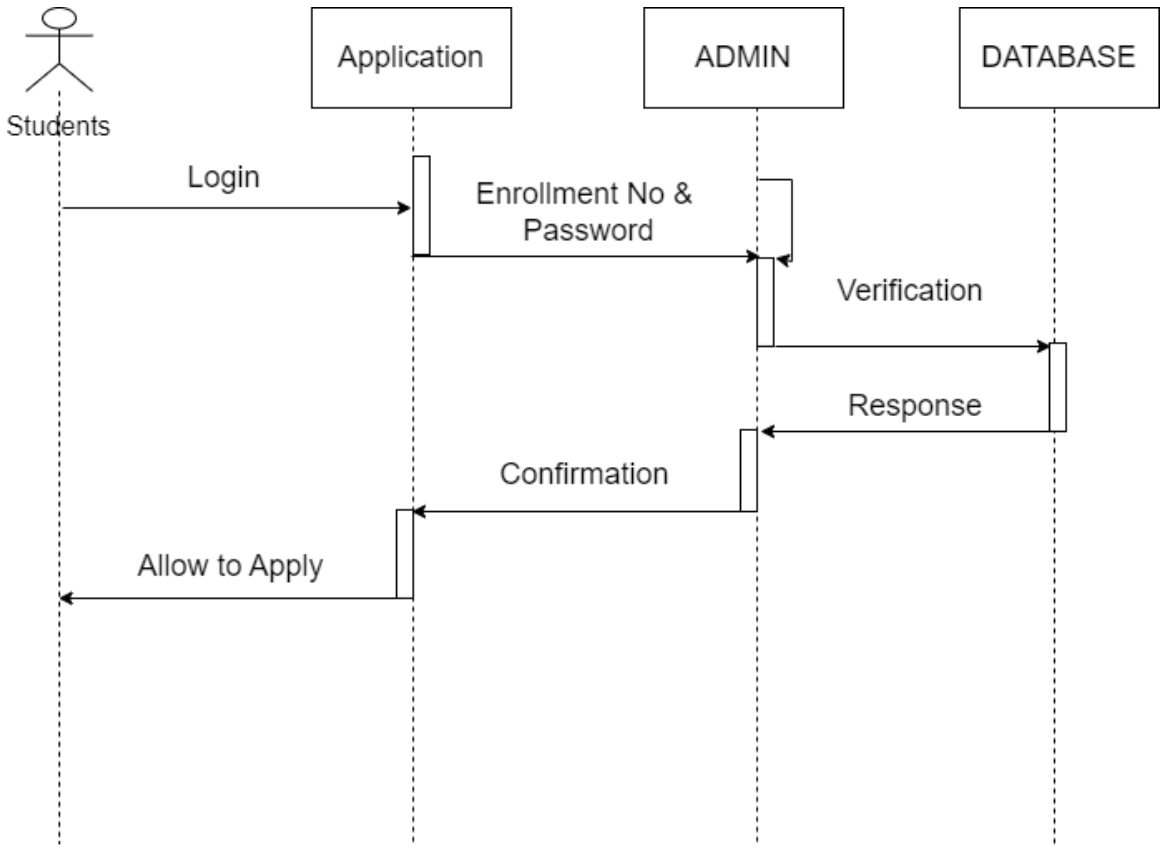


Figure 4.3.1 Sequence Diagram for Students

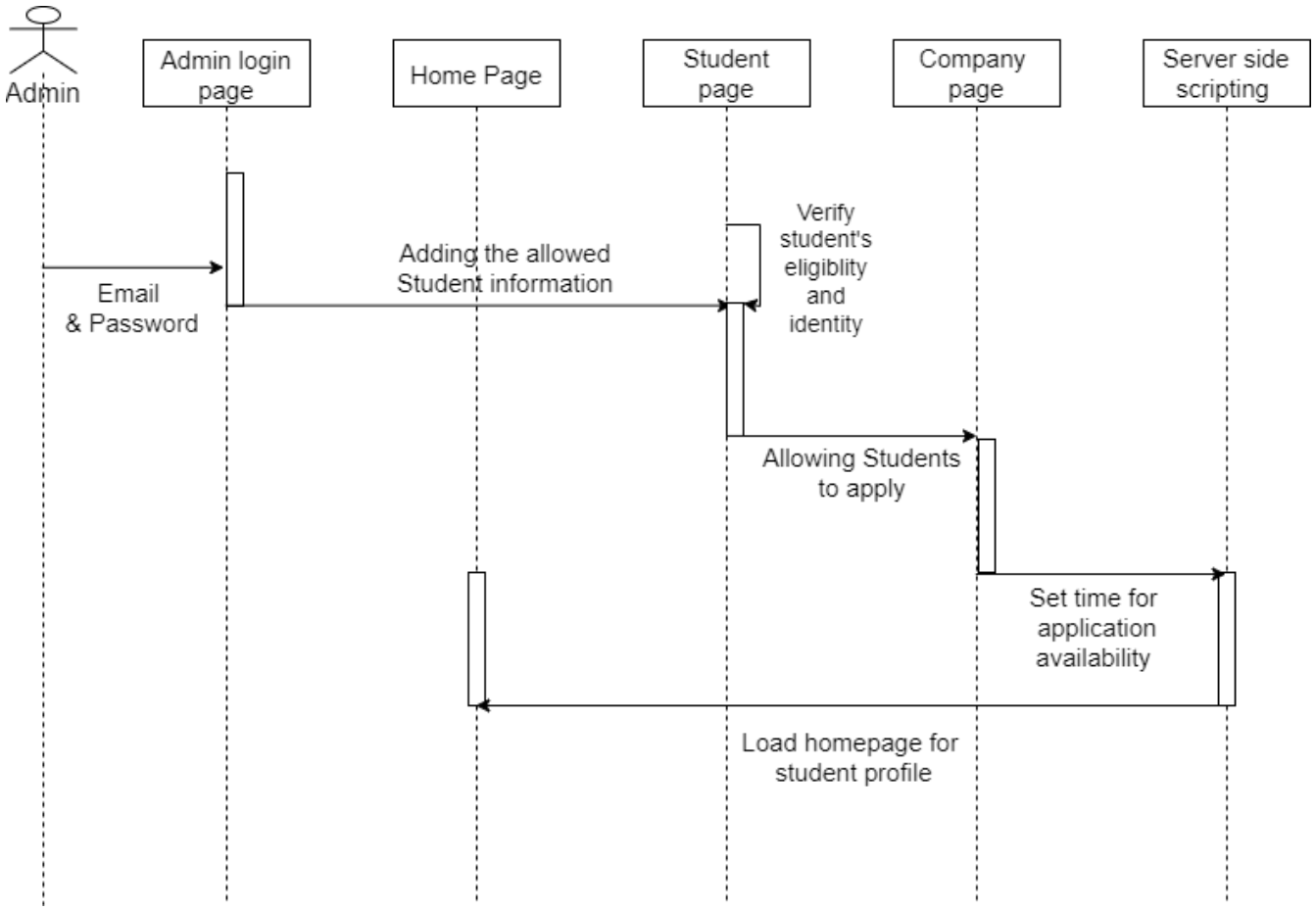


Figure 4.3.2 Sequence Diagram for Admin

4.4 Data Flow Diagram:

DFD graphically represents the functions, or processes, which capture, manipulate, store, and distribute data between a system and its environment and between components of a system.

The visual representation makes it a good communication tool between User and System designer.

Structure of DFD allows starting from a broad overview and expand it to a hierarchy of detailed diagrams. DFD has often been used due to the following reasons:

- Logical information flow of the system
- Determination of physical system construction requirements
- Simplicity of notation
- Establishment of manual and automated systems requirements

Context Level:

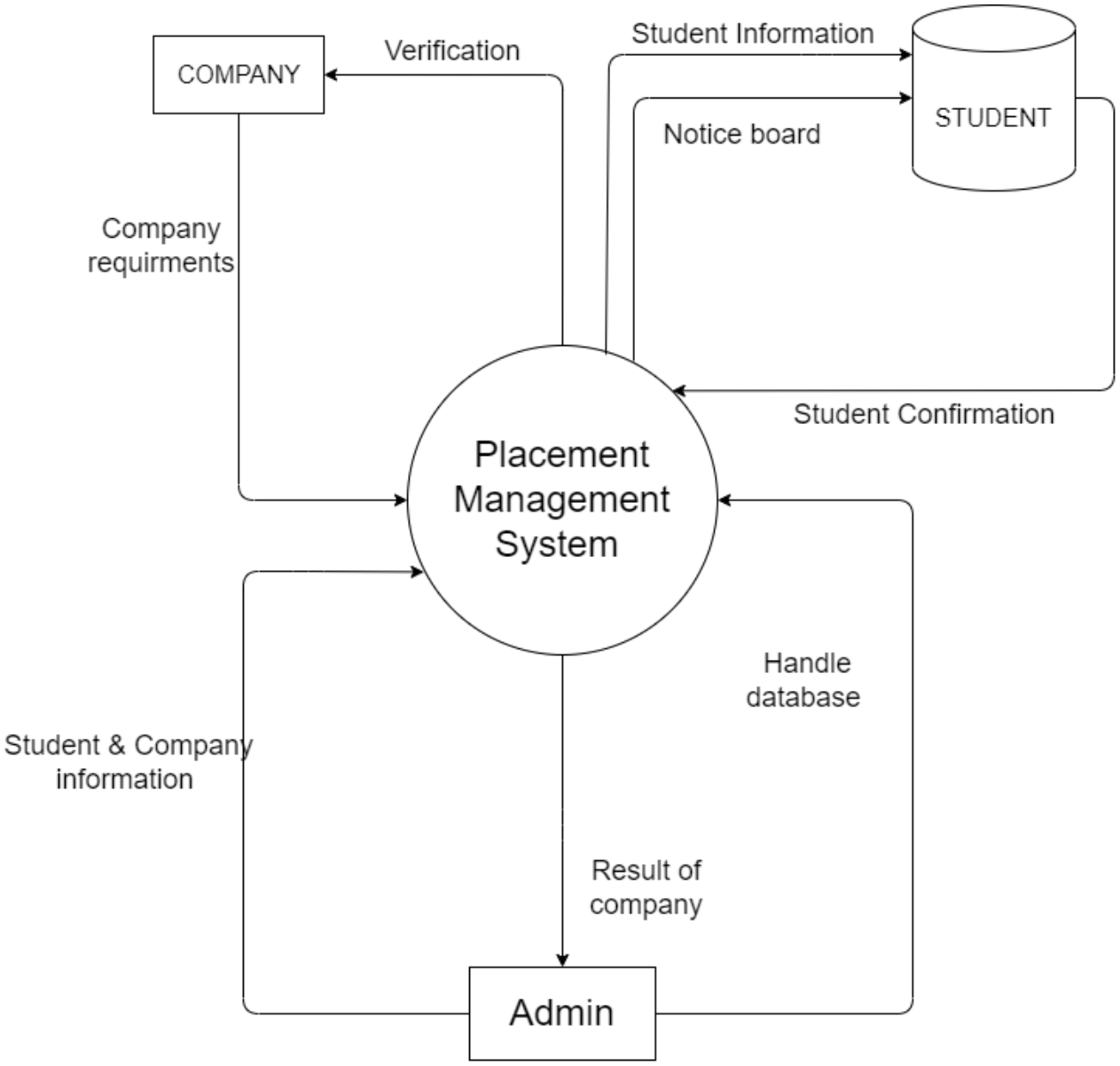


Figure 4.4.1 0-Level Data Flow Diagram

1-Level DFD:

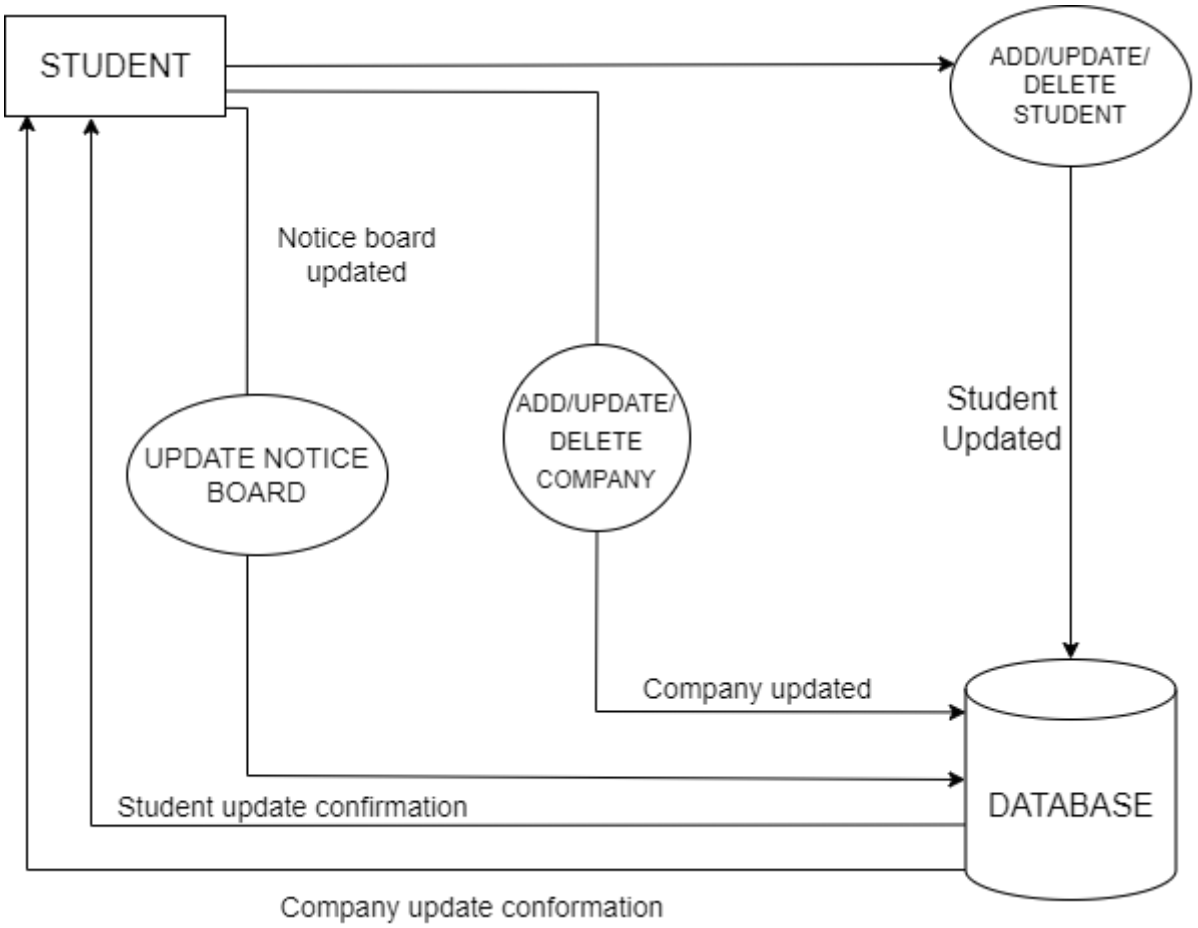


Figure 4.4.2 1-Level Data Flow Diagram

2-Level DFD:

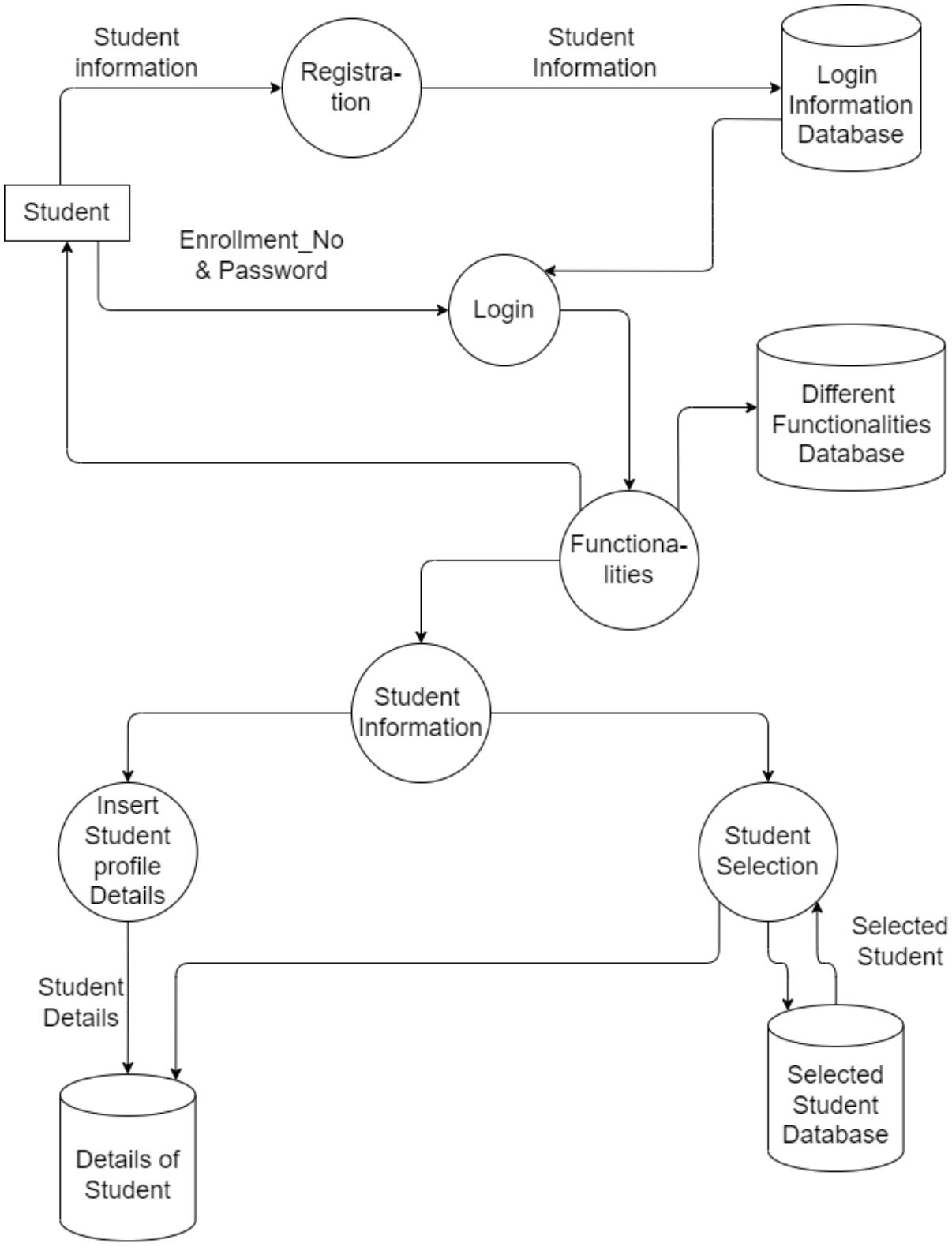


Figure 4.4.3 2-Level Data Flow Diagram

5. SYSTEM DESIGN

5.1 Entity Relationship Diagram:

An entity relationship diagram (ERD) shows the relationships of entity sets stored in a database.
An entity in this context is an object, a component of data.

An entity set is a collection of similar entities. These entities can have attributes that define its properties by defining the entities, and their attributes, and showing the relationships between them, an ER diagram illustrates the logical structure of databases.

ER diagrams are used to sketch out the design of a database.

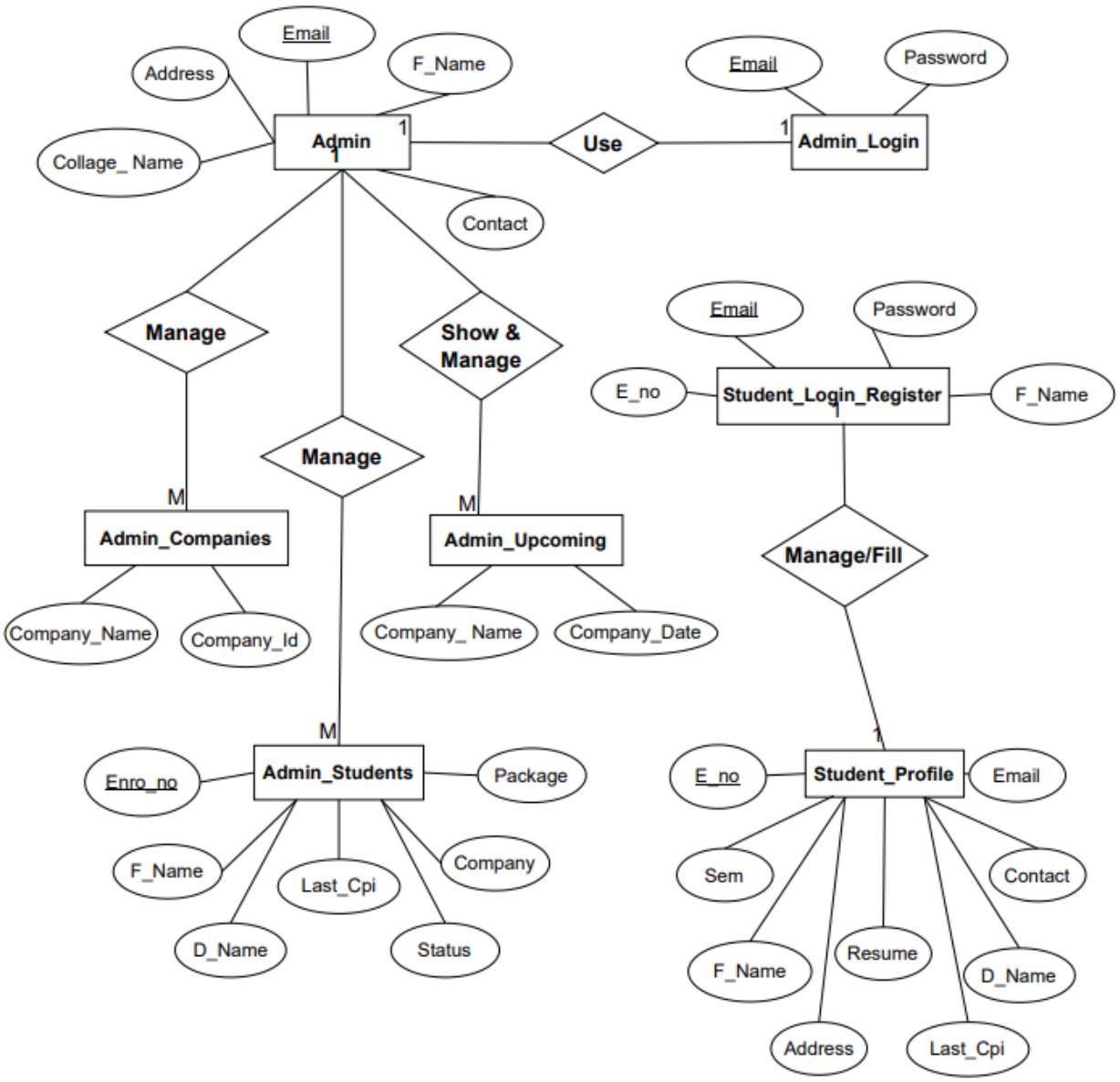


Figure 5.1 Entity Relationship Diagram

5.2 Data Dictionary Diagram:

A data dictionary contains metadata i.e., data about the database. The data dictionary is very important as it contains information such as what is in the database, who is allowed to access it, where is the database physically stored etc. The users of the database normally don't interact with the data dictionary, it is only handled by the database administrators.

The data dictionary in general contains information about the following;

- Names of all the database tables and their schemas.
- Details about all the tables in the database, such as their owners, their security constraints, when they were created etc.
- Physical information about the tables such as where they are stored and how.
- Table constraints such as primary key attributes, foreign key information etc.
- Information about the database views that are visible.

admin:

Column_Name	Data_Type	Size	Constraint
f_name	Varchar	255	Not Null
email	Varchar	255	Primary key
contact	big int	10	Not null
college_name	Varchar	255	Not null
address	Varchar	255	Not null

admin_login:

Column_Name	Data_Type	Size	Constraint
email	Varchar	255	Primary key
password	varchar	255	Not null

admin_students:

Column_Name	Data_Type	Size	Constraint
enro_no	varchar	30	Unique Key
f_name	Varchar	255	Not Null
d_name	Varchar	255	Not Null
last_cpi	float	10	Not Null
status	varchar	30	Not Null
company	Varchar	255	Not Null
package	Int	255	Not Null

admin_companies:

column_Name	Data_Type	Size	Constraint
company_id	varchar	255	Unique Key
company_name	varchar	255	Not Null
company_package	float	100	Not Null
comapny_desc	varchar	255	Not Null

admin_upcoming:

Column_Name	Data_Type	Size	Constraint
company_name	varchar	100	Not Null
company_date	Int	10	Not Null
comapny_desc	varchar	255	Not Null

Student_login_register:

Column_Name	Data_Type	Size	Constraint
f_name	varchar	50	Not Null
email	Varchar	50	Primary key
e_no	Varchar	20	Not Null
password	varchar	30	Not Null

Student_profile:

Column_Name	Data_Type	Size	Constraint
f_name	varchar	50	Not Null
Enrollment_no	Varchar	255	Primary key
Department_name	Varchar	255	Not Null
semester	int	8	Not Null
last cpi	float	10	Not Null
address	Varchar	255	Not Null
contact	bigint	255	Not Null
email	varchar	255	Not Null
resume	Varchar	255	Not Null

5.3 Activity Diagram:

Activity diagram is another important diagram in UML to describe the dynamic aspects of the system.

An activity diagram is basically a flowchart to represent the flow from one activity to another activity. The activity can be described as an operation of the system.

The control flow is drawn from one operation to another. This flow can be sequential, branched, or concurrent. Activity diagrams deal with all types of flow control by using different elements such as fork, join, etc.

The purpose of an activity diagram can be described as −

- Draw the activity flow of a system.
- Describe the sequence from one activity to another.
- Describe the parallel, branched and concurrent flow of the system.

 There are 2 activity diagrams for this system one for the user side and another for the administration side.

5.3.1 Activity Diagram For Admin:

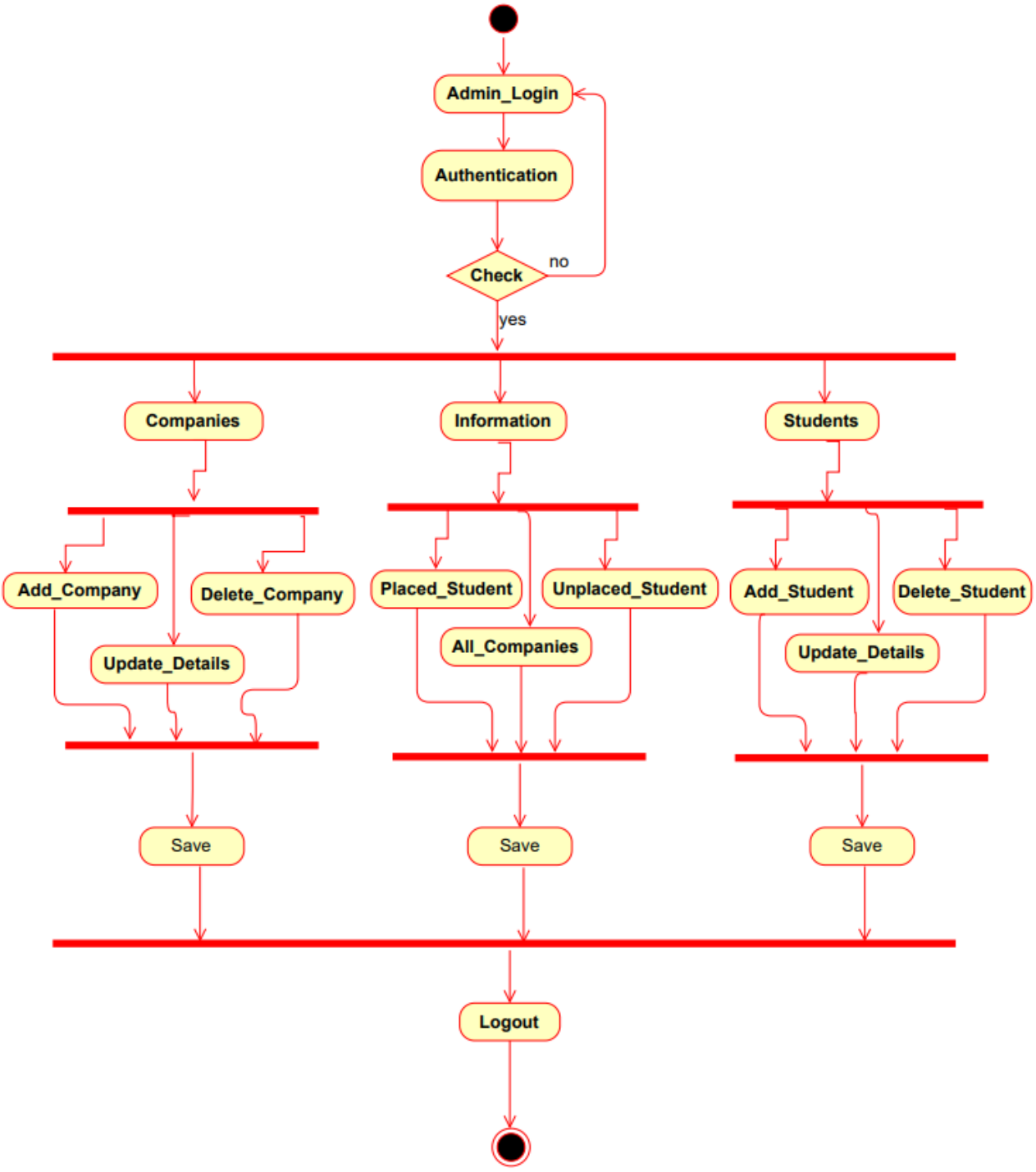


Figure 5.2 Activity Diagram for Admin

5.3 Input/Output and Surface design:

5.4.1 Sample of forms, reports and interface:

- Admin Login Page:

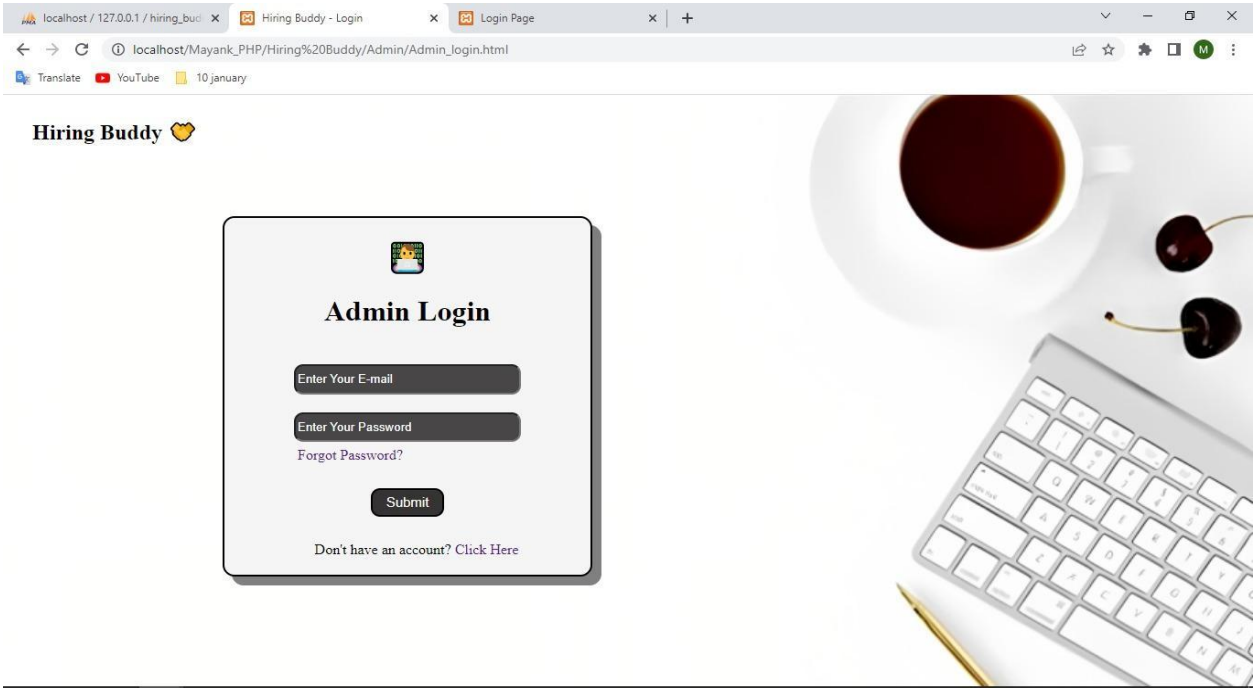


Figure 5.3.1 Admin Login Page

- Admin Home Page:

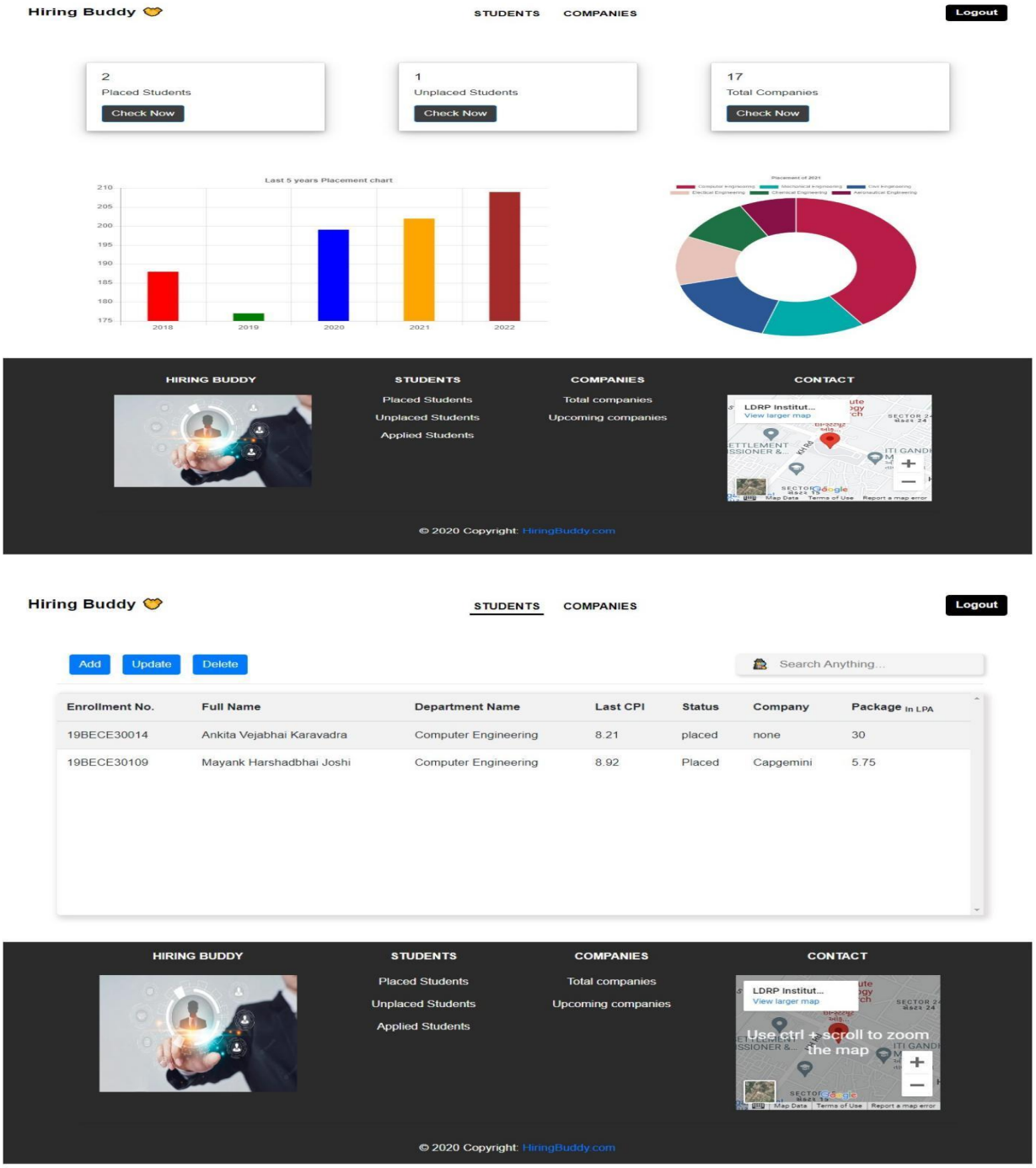
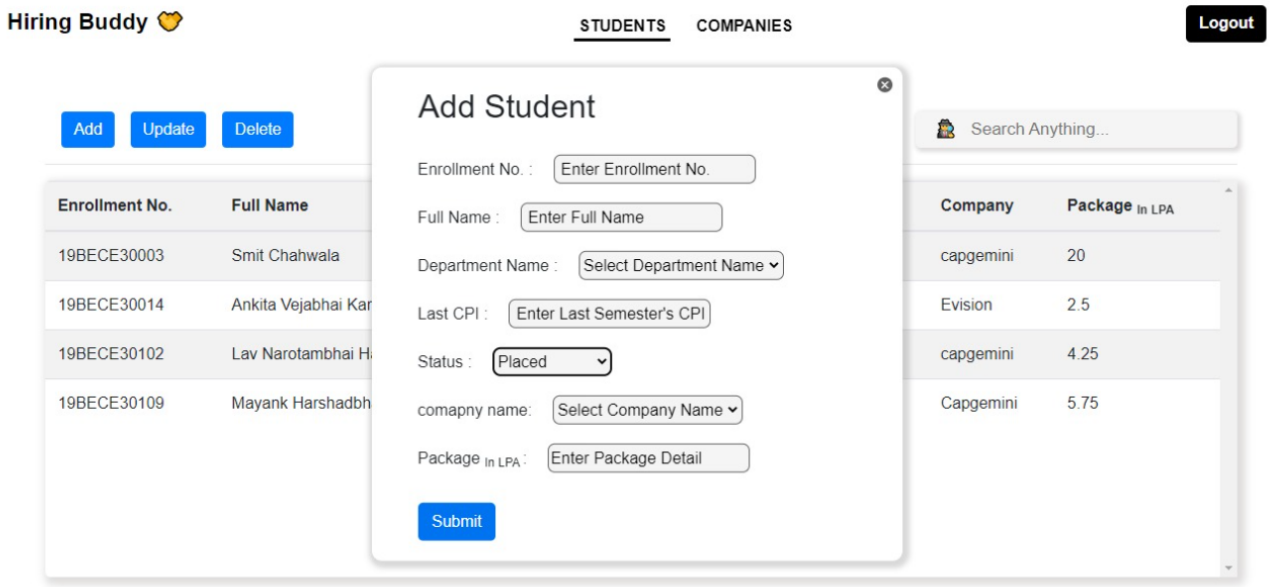


Figure 5.3.2 Home Page Admin



- Company Home Page

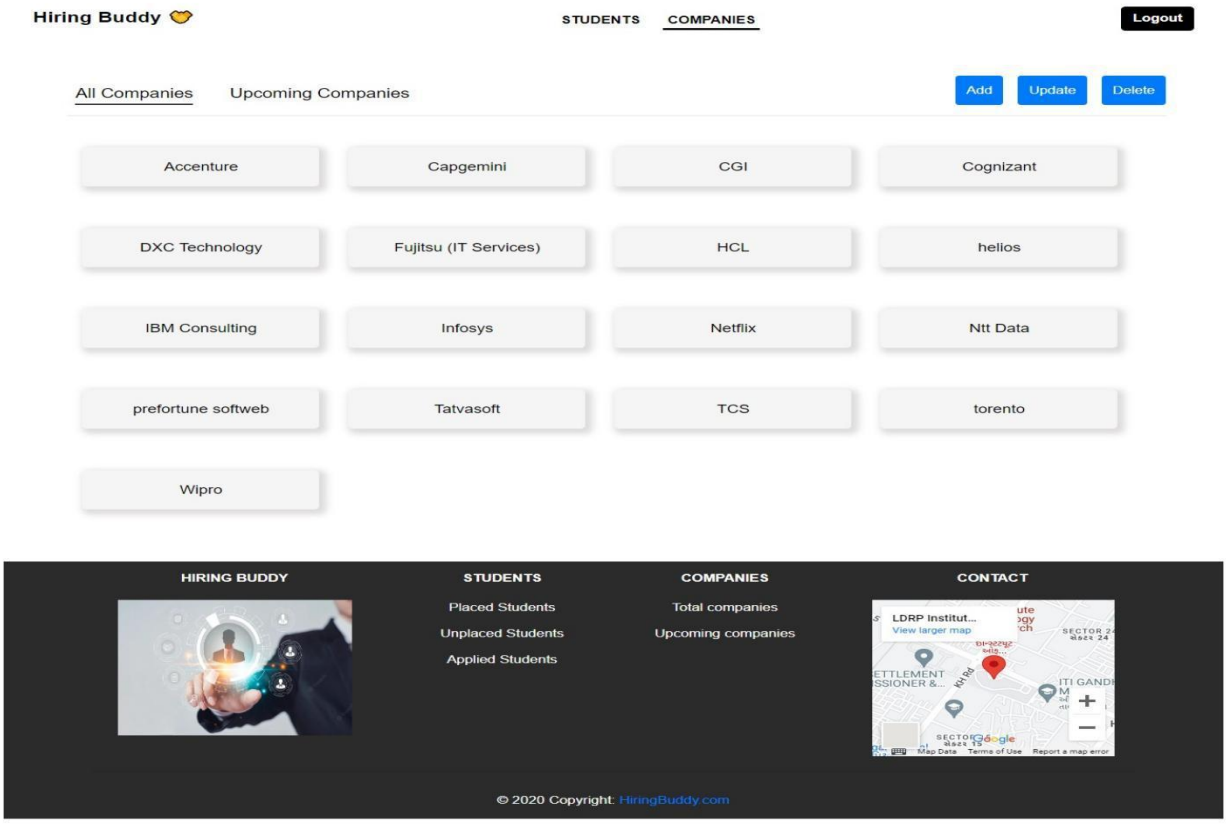


Figure 5.3.3 Company Home Page

- Upcoming Company Page

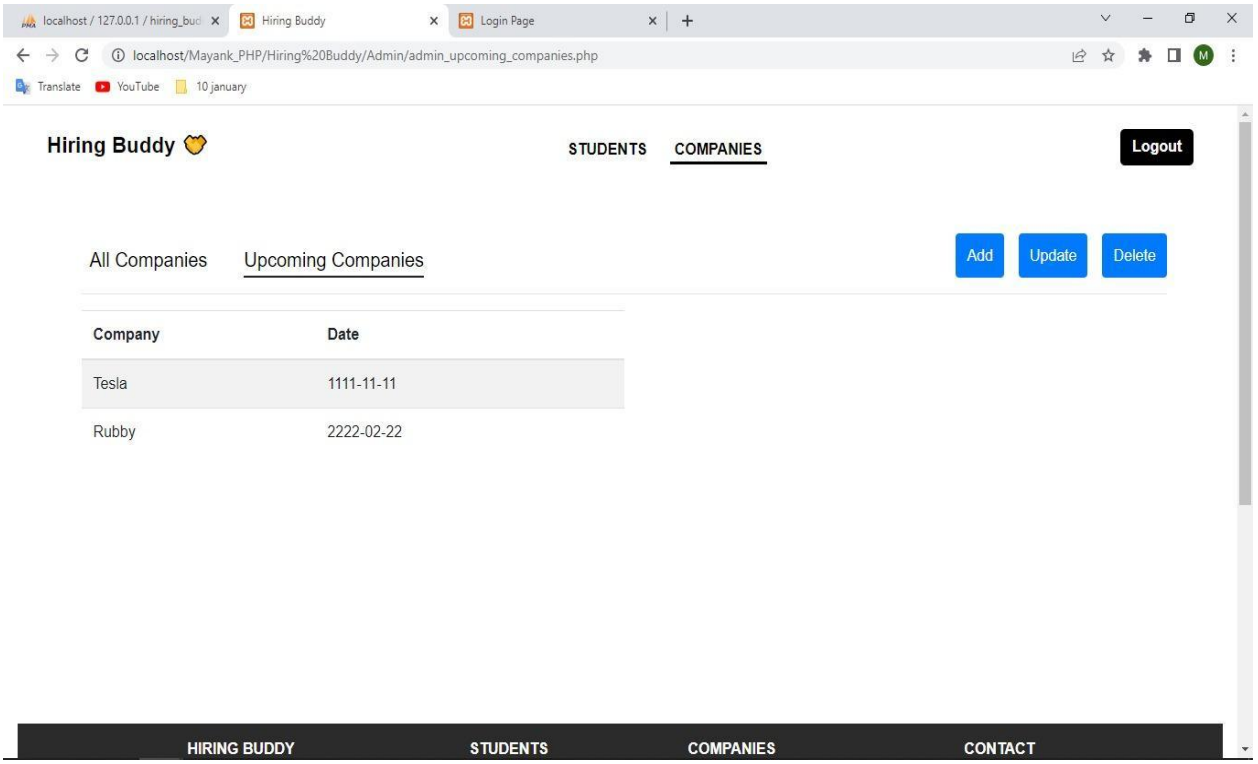


Figure 5.3.4 Upcoming Company Page

- Student Registration Page

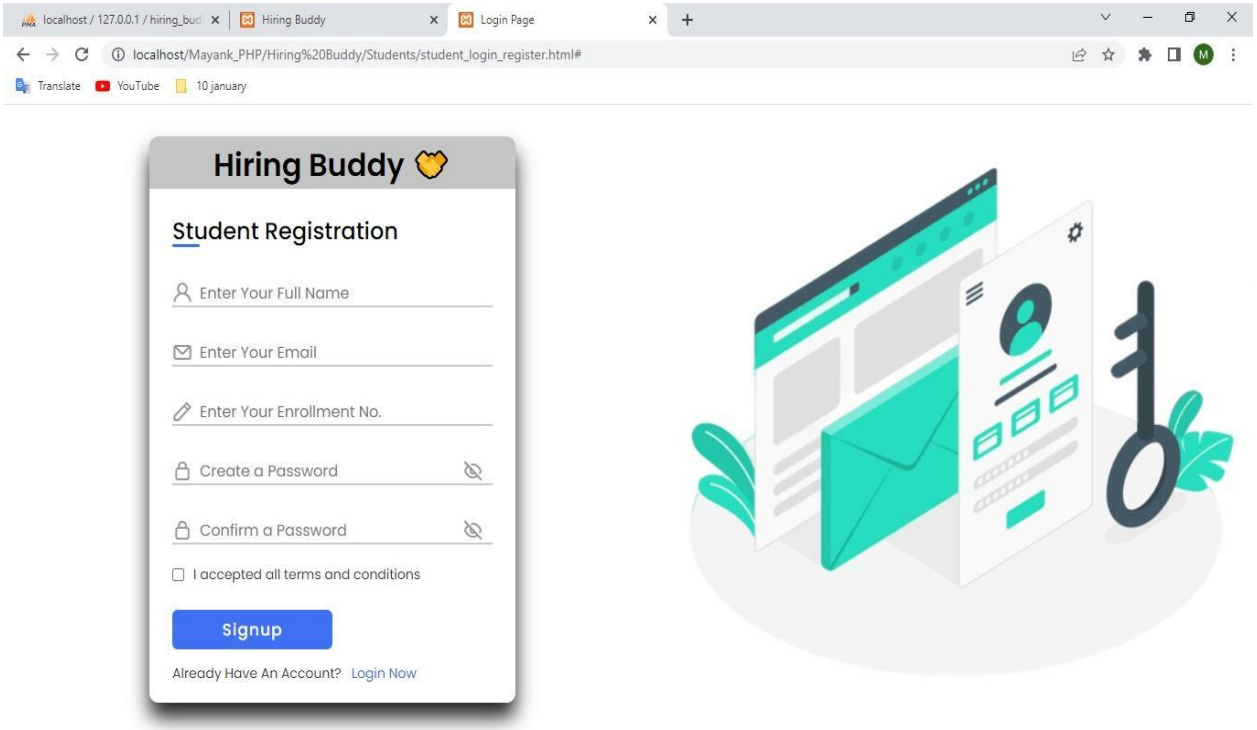


Figure 5.3.5 Student Registration Page

- Student Login Page

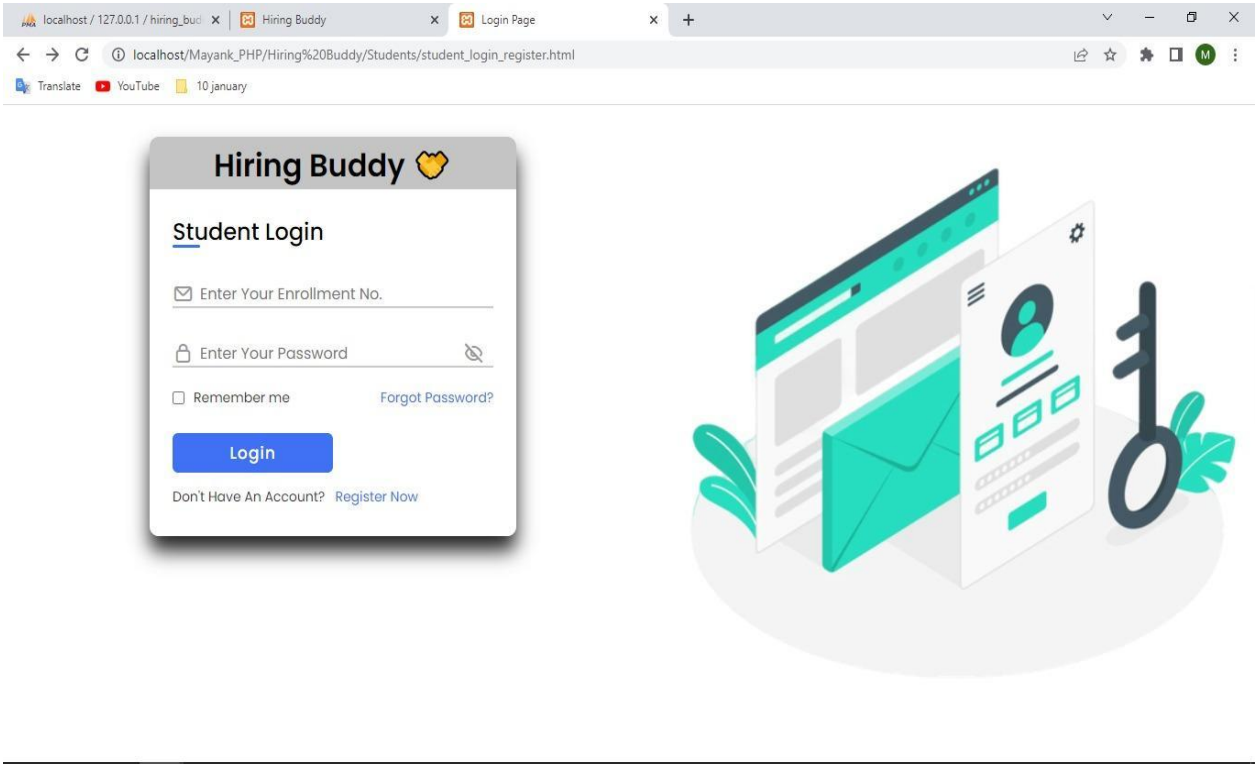


Figure 5.3.6 Student Login Page

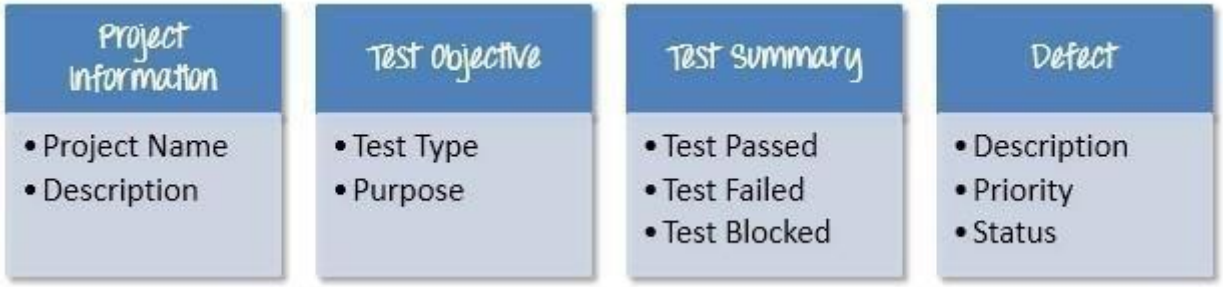
6 TESTING

6.1 Test Report:

A test Report is a document which contains a summary of all test activities and final test results of a testing project. The test report is an assessment of how well the Testing is performed. Based on the test report, stakeholders can evaluate the quality of the tested product and make a decision on the software release.

For example, if the test report informs that there are many defects remaining in the product, stakeholders can delay the release until all the defects are fixed.

- What does a test report contain?



All information about the project such as the project name, product name, and version should be described in the test report.

Test Objective:

As mentioned in the Test Planning tutorial, Test Report should include the objective of each round of testing, such as Unit Test, Performance Test, System Test …etc.

Test Summary:

This section includes the summary of testing activity in general. The information detailed here includes.

6.2 Testing Planning Steps:

6.2.1 Functionality Testing:

Test for – all the links in web pages, database connection, forms used in the web pages for submitting or getting information from users, and Cookie testing. Check all the links:

- Test the outgoing links from all the pages from the specific domain under test.
- Test all internal links.
- Test links jumping on the same pages.
- Test links used to send the email to admin or other users from web pages.
- Test to check if there are any orphan pages.
- Lastly in link checking, check for broken links in all the above-mentioned links.

Test forms in all pages: Forms are an integral part of any website. Forms are used to get information from users and to keep interacting with them. So, what should be checked on these forms?

- First check all the validations on each field.
- Check for the default values of fields.
- Wrong inputs to the fields in the forms.
- Options to create forms if any, form delete, view or modify the forms. Let's take the example of the search engine project currently I am working on, in this project, we have advertiser and affiliate signup steps. Each signup step is different but dependent on other steps. So, sign-up flow should get executed correctly.

There are different field validations like email Ids, User financial info validations.

All these validations should get checked in manual or automated web testing.

Cookies testing: Cookies are small files stored on the user's machine. There are basically used to maintain the session mainly login sessions. Test the application by enabling or disabling the cookies in your browser options. Test if the cookies are encrypted before writing to the user's machine. If you are testing the session cookies (i.e. cookies expire after the session ends) check for login sessions and user stats after the session end. Check the effect on application security by deleting the cookies.

Validate your HTML/CSS: If you are optimizing your site for Search engines then HTML/CSS validation is very important. Mainly validate the site for HTML syntax errors. Check if the site is crawlable to different search engines.

- **Database testing**: Data consistency is very important in web applications. Check for data integrity and errors while you edit, delete, modify the forms or do any DB-related functionality. Check, if all the database queries are executing correctly, data is retrieved correctly and also updated correctly. More database testing could be loaded on DB, we will address this in web load or performance testing below.

6.2.2 Usability Testing:

- **Test for navigation:** Navigation means how the user surfs the web pages, different controls like buttons, boxes or how the user uses the links on the pages to surf different pages. Usability testing includes: Web site should be easy to use. Instructions should be provided clearly. Check if the provided instructions are correct means whether they satisfy the purpose. The main menu should be provided on each page. It should be consistent.

- **Content:** Content should be logical and easy to understand. Check for spelling errors. The use of dark colours annoys users and should not be used in the site theme. You can follow some standards that are used for web page and content building. These are commonly accepted standards as I mentioned above about annoying colours, fonts, frames etc.

- Content should be meaningful. All the anchor text links should be working properly. Images should be placed properly with proper sizes. These are some basic standards that should be followed in web development. Your task is to validate all for UI testing.

- Other user information for user help: Like search option, sitemap, help files etc. Sitemap should be present with all the links in websites with a proper tree view of navigation. Check for all links on the sitemap. The “Search in the site” option will help users to find the content pages they are looking for easily and quickly. These are all optional items and if present should be validated.

6.2.3 Interface testing:

The main interfaces are:

- Web server and application server interface
- Application server and Database server interface.

Check if all the interactions between these servers are executed properly. Errors are handled properly. If the database or web server returns any error message for any query by the application server then the application server should catch and display these error messages appropriately to users.

Check what happens if the user interrupts any transaction in between. Check what happens if the connection to the web server is reset in between.

6.2.4 Performance testing:

Web applications should sustain the heavy load. Web performance testing should include

- Web Load Testing
- Web Stress Testing

Test application performance on different internet connection speeds. In the web load testing test if many users are accessing or requesting the same page. Can the system sustain peak load times? The site should handle many simultaneous user requests, large input data from users, Simultaneous connection to DB, heavy load on specific pages etc.

Stress testing: Generally, stress means stretching the system beyond its specification limits. Web stress testing is performed to break the site by giving stress and checking how the system reacts to stress and how the system recovers from crashes.

Stress is generally given on input fields, login and sign-up areas. In web performance, testing website functionality on different operating systems, and different hardware platforms are checked for software and hardware memory leakage errors.

6.2.5 Security testing:

Following are some test cases for web security testing:

- Test by pasting the internal URL directly into the browser address bar without login. Internal pages should not open.
- If you are logged in using a username and password and browsing internal pages then try changing URL options directly. I.e. If you are checking some publisher site statistics with publisher site ID= 123. Try directly changing the URL site ID parameter to a different site ID which is not related to the log of the user. Access should be denied for this user to view others' stats.
- Try some invalid inputs in input fields like login username, password, and input text boxes. Check the system reaction on all invalid inputs.
- Web directories or files should not be accessible directly unless given a download option.
- Test if SSL is used for security measures. If used proper message should get displayed when the user switch from non-secure http:// pages to secure https:// pages and vice versa.

All transactions, error messages, security breach attempts should get logged in log files somewhere on a web server.

6.3 Testing Strategies:

6.3.1 White Box Testing:

If we go by the definition, “White box testing” (also known as clear, glass box or structural testing) is a testing technique which evaluates the code and the internal structure of a program.

White box testing involves looking at the structure of the code. When you know the internal structure of a product, tests can be conducted to ensure that the internal operations are performed according to the specification. And all internal components have been adequately exercised.

White box testing coverage specifications:

1. Code coverage

2. **Segment coverage:** Ensure that each code statement is executed once.

3. **Branch Coverage or Node Testing:** Coverage of each code branch in all possible ways.

4. **Compound Condition Coverage:** For multiple conditions test each condition with multiple paths and a combination of the different paths to reach that condition.

5. **Basis Path Testing:** Each independent path in the code is taken for testing.

6. **Data Flow Testing (DFT):** In this approach you track the specific variables through each possible calculation, thus defining the set of intermediate paths through the code. DFT tends to reflect dependencies but it is mainly through sequences of data manipulation. In short, each data variable is tracked and its use is verified. This approach tends to uncover bugs like variables used but not initialised or declared but not used, and so on.

7. **Path Testing:** Path testing is where all possible paths through the code are defined and covered. It's a time-consuming task.

8. **Loop Testing:** These strategies relate to testing single loops, concatenated loops, and nested loops. Independent and dependent code loops and values are tested by this approach.

Limitations:
Not possible for testing each and every path of the loops in the program. This means exhaustive testing is impossible for large systems.

This does not mean that WBT is not effective. Selecting important logical paths and data structures for testing is practically possible and effective.

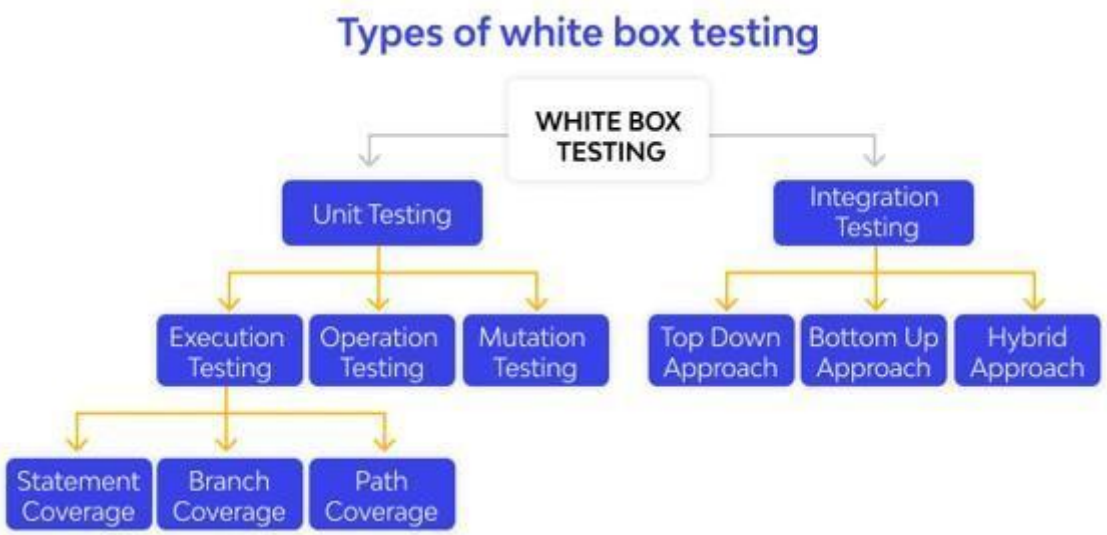
Steps to Perform WBT:

Step 1 – Understand the functionality of an application through its source code. This means that a tester must be well-versed with the programming language and the other tools as well as techniques used to develop the software.

Step 2– Create the tests and execute them.

When we discuss the concept of testing, "coverage" is considered to be the most important factor. Here I will explain how to have maximum coverage from the context of White box testing.

- **Type Of White Box Testing:**



Today, we are going to focus mainly on the execution testing types of the 'Unit testing white box technique'.

3 Main White Box Testing Techniques:

1. Statement Coverage
2. Branch Coverage
3. Path Coverage

Note that the statement, branch or path coverage does not identify any bug or defect that needs to be fixed. It only identifies those lines of code which are either never executed or remain untouched. Based on this further testing can be focused on.

Let's understand these techniques one by one with a simple example.

1) Statement coverage:

In a programming language, a statement is nothing but a line of code or instruction for the computer to understand and act accordingly. A statement becomes an executable statement when it gets compiled and converted into the object code and performs the action when the program is in running mode. Hence, *"Statement Coverage"*, as the name itself suggests, is the method of validating whether each and every line of the code is executed at least once.

2) Branch Coverage:

"Branch" in a programming language is like the "IF statements". An IF statement has two branches: True and False.
So, in Branch coverage (also called Decision coverage), we validate whether each branch is executed at least once.

In the case of an "IF statement", there will be two test conditions: One to validate the true branch and, the other to validate the false branch.

Hence, in theory, Branch Coverage is a testing method which is when executed ensures that each and every branch from each decision point is executed.

3) Path Coverage:

Path coverage tests all the paths of the program. This is a comprehensive technique which ensures that all the paths of the program are traversed at least once. Path Coverage is even more powerful than Branch coverage. This technique is useful for testing complex programs.

6.3.2 Black Box Testing:

Black Box Testing is also known as behavioural, opaque-box, closed-box, specification-based or eye-to-eye testing.

It is a Software Testing method that analyses the functionality of a software/application without knowing much about the internal structure/design of the item that is being tested and compares the input value with the output value.

The main focus in Black Box Testing is on the functionality of the system as a whole. The term 'Behavioural Testing' is also used for Black Box Testing. Behavioural test design is slightly different from the black-box test design because the use of internal knowledge isn't strictly forbidden, but it's still discouraged.

Each testing method has its own advantages and disadvantages. There are some bugs that cannot be found using the only black box or only white box technique.

The majority of the applications are tested by the Black Box method. We need to cover the majority of test cases so that most of the bugs will get discovered by a Black Box method.

This testing occurs throughout the software development and Testing Life Cycle i.e., in the Unit, Integration, System, Acceptance, and Regression Testing stages. This can be both Functional and Non-Functional.

Types of Black Box Testing:

1) Functional Testing:
This type deals with the functional requirements or specifications of an application. Here, different actions or functions of the system are being tested by providing the input and comparing the actual output with the expected output.

For Example, when we test a Dropdown list, we click on it and verify that it expands and all the expected values are showing in the list.

A few major types of Functional Testing are:

- Smoke Testing
- Sanity Testing
- Integration Testing
- System Testing
- Regression Testing
- User Acceptance Testing

2) Non-Functional Testing:

Apart from the functionalities of the requirements, there are several non-functional aspects as well that are required to be tested to improve the quality and performance of the application.

A few major types of Non-Functional Testing include

- Usability Testing
- Load Testing
- Performance Testing
- Compatibility Testing
- Stress Testing
- Scalability Testing

Advantages:

- The tester need not have a technical background. It is important to test by being in the user's shoes and thinking from the user's point of view.
- Testing can be started once the development of the project/application is done. Both the testers and developers work independently without interfering in each other's space.
- It is more effective for large and complex applications.
- Defects and inconsistencies can be identified at the early stage of testing.

Disadvantages:

- Without any technical or programming knowledge, there are chances of ignoring possible conditions of the scenario to be tested.
- In a stipulated time, there are possibilities of testing less and skipping all possible inputs and their output testing.
- A Complete Test Coverage is not possible for large and complex projects

6.4 Test Case Scenario:

6.4.1 Test Case for Inconsistent Database:

TEST CASE ID: TC001	**NAME: Database Inconsistency**
PURPOSE	Checking out the consistency of database in oracle server through SQL Developer
INPUT	Inserting new data through form
EXPECTED OUTPUT	Reflection/addition of data in the database.
STEPS:	

1. Open the user login form.
2. Fill in data and click submit it to add to the database this should be reflected in database tables.

6.4.2 Test Case for new donation functioning well:

TEST CASE ID: TC002	**NAME: Add New Student**
PURPOSE	To assure that a new Student is added successfully.
INPUT	Enrollment No, Full Name, CPI, Department Name, Status.
EXPECTED OUTPUT	New Student is successfully Added.
STEPS:	

1. First of all log in with proper credentials.
2. Select add button and fill in the details. click save.

Figure 6.1 Test Case Diagram

7. CONCLUTION

▯ CONCLUTION:

- Hiring Buddy manages student information in the college with regard to placement.
- It also provides facilities to companies which will hold their campus drive in colleges. They can communicate with the college placement head and student directly to avoid the time delay of the hiring process.
- Overall, the purpose of Hiring Buddy is to give some relaxation and excitement to placement officers, students and companies to make the process easy.

.

8. BIBLIOGRAPHY

- **Web Resources:**
  - www.w3cschools.com
  - https://www.javatpoint.com/software-engineering-tutorial
  - https://www.guru99.com/how-test-reports-predict-the-success-ofyour-testing-project.html
  - https://www.tutorialspoint.com/software_engineering/software_project_management.htm

- **Reference Books:**
  - Gunderloy, Jorden BPB Publications (2000) - "Mastering SQL Server"
  - Luke Welling and Laura Thomson (5th Edition) - "PHP and MySQL Web Development"
  - Roger S.Pressmen, T. Mc. GH. – Software Engineering (Theoretical Approach)
  - Thereon Willis wrox publications (2000) - "Beginning SQL Server"