CS1113: Software Engineering

“PlaceDoor”

Placement Management System

Report

PREPARED BY

Ayush Ranwa (2020BTechCSE016)

Keshaw Soni (2020BTechCSE039)

Somya Gautam (2020BTechCSE076)

Yash Agarwal (2020BTechCSE102)

FACULTY GUIDE

Prof. Amit Sinhal

Department of Computer Science and Engineering

Institute of Engineering and Technology

JK Lakshmipat University Jaipur

April 2023

CERTIFICATE

This is to certify that the project work entitled "**PlaceDoor (Placement Management System)**" submitted by **Ayush Ranwa, Keshaw Soni, Somya Gautam, and Yash Agarwal** towards the partial fulfilment of the requirements for the degree of **Bachelor of Technology in Computer Science Engineering** of JK Lakshmipat University, Jaipur is the record of work carried out by them under our supervision and guidance. In our opinion, the submitted work has reached the level required for being accepted for the final submission of the project.

Prof. Amit Sinhal

Professor

Department of Computer Science Engineering

Institute of Engineering & Technology

JK Lakshmipat University, Jaipur

ACKNOWLEDGEMENTS

We have completed this project under the guidance and supervision of **Prof. Amit Sinhal** Professor, IET, JK Lakshmipat University**.** We will be failed in our duty if we do not acknowledge the esteemed scholarly guidance, assistance, feedback, and knowledge we have received from him towards the fruitful and timely completion of this work.

We express our deepest thanks to **Dr. Dheeraj Sanghi**, Vice Chancellor, JK Lakshmipat University, and **Dr. Sanjay Goel**, Director, Institute of Engineering and Technology, JK Lakshmipat University for their constant support, encouragement, and guidance.

We also acknowledge with a deep sense of reverence, our gratitude towards our parents for their direct or indirect support during the entire course of this project.

Thanking You

Yours Sincerely,

Ayush Ranwa
Keshaw Soni
Somya Gautam
Yash Agarwal

ABSTRACT

This report presents the main objectives, features, and outcomes of our project, "PlaceDoor," which aims to address the challenges faced by students in the placement process.

The report encompasses a comprehensive set of UML diagrams, elucidating the system's architecture, design, and flow, along with project snapshots showcasing the practical implementation of the code. Furthermore, the poster project presents an in-depth overview of the project's implementation, and insights into the development and testing process.

PlaceDoor is a comprehensive and user-friendly platform that centralizes placement-related activities in universities. Its features include personalized resume customization, job application functionality, and updates on applied jobs, empowering students to enhance their chances of securing suitable placements. For companies, PlaceDoor offers an easy-to-use job posting platform with relevant candidate data, facilitating efficient recruitment processes. The report highlights the successful implementation of PlaceDoor and its positive impact on improving opportunities for successful placements and seamless placement management.

LIST OF FIGURES

TABLE OF CONTENTS

1. INTRODUCTION

1.1 Problem Statement

The problem statement revolves around the lack of a dedicated platform for placement-related activities in universities, resulting in frustration among students. The absence of a centralized platform leads to challenges such as students missing out on job postings and requirements, having to rely on multiple platforms like Outlook and WhatsApp for updates, and sending generic resumes to all companies. This scattered approach to job search and application process often results in inefficiencies, missed opportunities, and reduced chances of securing suitable placements.

To address this problem, the solution being proposed by our team is "PlaceDoor," a platform designed to streamline and centralize placement-related activities. PlaceDoor aims to provide a comprehensive platform where companies can post job openings, and students can access and apply for these jobs within the same platform. The platform will also have an administrative component that will enable monitoring and tracking of all activities, ensuring smooth functioning and efficient management of the placement process.

1.2 Objectives

The primary aims of this project are to:

- To create streamline and centralize placement-related activities platform for universities.
- To develop a user-friendly job application platform for students that simplifies the process of applying for jobs.
- To enable companies to easily post job openings and efficiently manage candidate data.
- To facilitate seamless administration and monitoring of the entire placement process.
- To improve communication and updates between students, companies, and administrators.
- To Reduce inefficiencies, missed opportunities, and frustrations associated with scattered job search and application processes.

1.3 Project Overview

1.3.1 Features and functionality

PlaceDoor intends to significantly improve the overall experience of students in their job search and application process. We aim to provide students with a user-friendly platform that offers personalized resume customization options. This will allow students to create resumes that highlight their skills, qualifications, and experiences in a way that resonates with potential employers. By tailoring their resumes to specific job openings, students can increase their chances of standing out and securing suitable placements.

Additionally, our platform will simplify the job application process for students. They will be able to easily search and apply for jobs within the platform, eliminating the need to navigate multiple websites or platforms for different job opportunities. This streamlined approach will save students time and effort in their job search, while also ensuring that their applications are submitted accurately and promptly.

Furthermore, our platform will enable students to view the jobs they have applied for and receive updates on the status of their applications. This will help students keep track of their job applications and stay informed about any changes or updates related to their placements. By providing timely and relevant information, our platform will empower students to actively manage their job search and increase their chances of success.

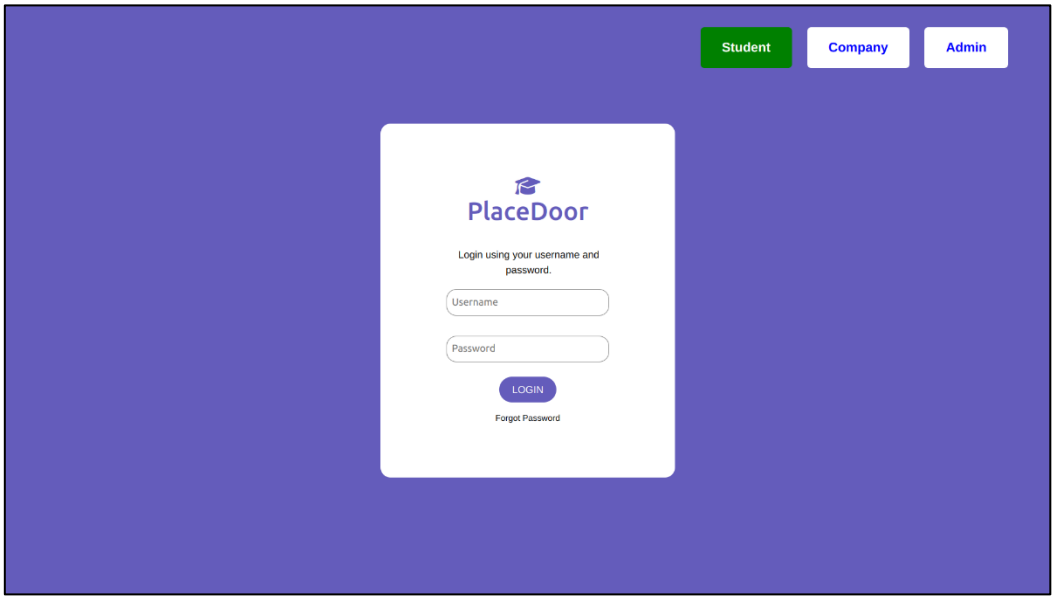
For companies, our platform will offer a straightforward and efficient process for posting jobs. Employers will be able to easily create and manage job postings, including job details, requirements, and application process. This will save companies time and effort in managing their job postings, ensuring that their openings reach a wide pool of qualified candidates.

The platform will also provide relevant data and analytics to companies, allowing them to make informed decisions about their hiring strategies and track the progress of their job postings.

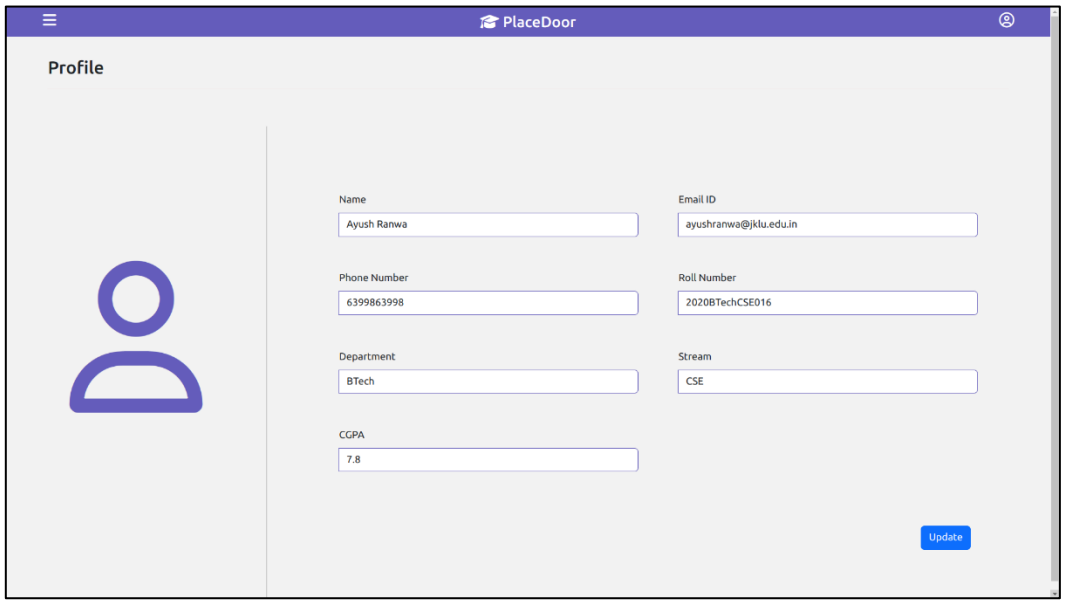
Finally, the administrative component of our platform will enable smooth management of the entire placement process. This includes features such as monitoring and tracking of job postings, applications, and placements, as well as communication and coordination among students, companies, and university officials. The administrative tools will ensure that the placement process is organized, efficient, and transparent, benefiting all stakeholders involved.

1.3.2 User Interface Design

User Interface Design is creating visually appealing, easy-to-use, and engaging interfaces for digital products like websites, apps, and games, ensuring a positive user experience through elements such as layout, navigation, visuals, and interactions.

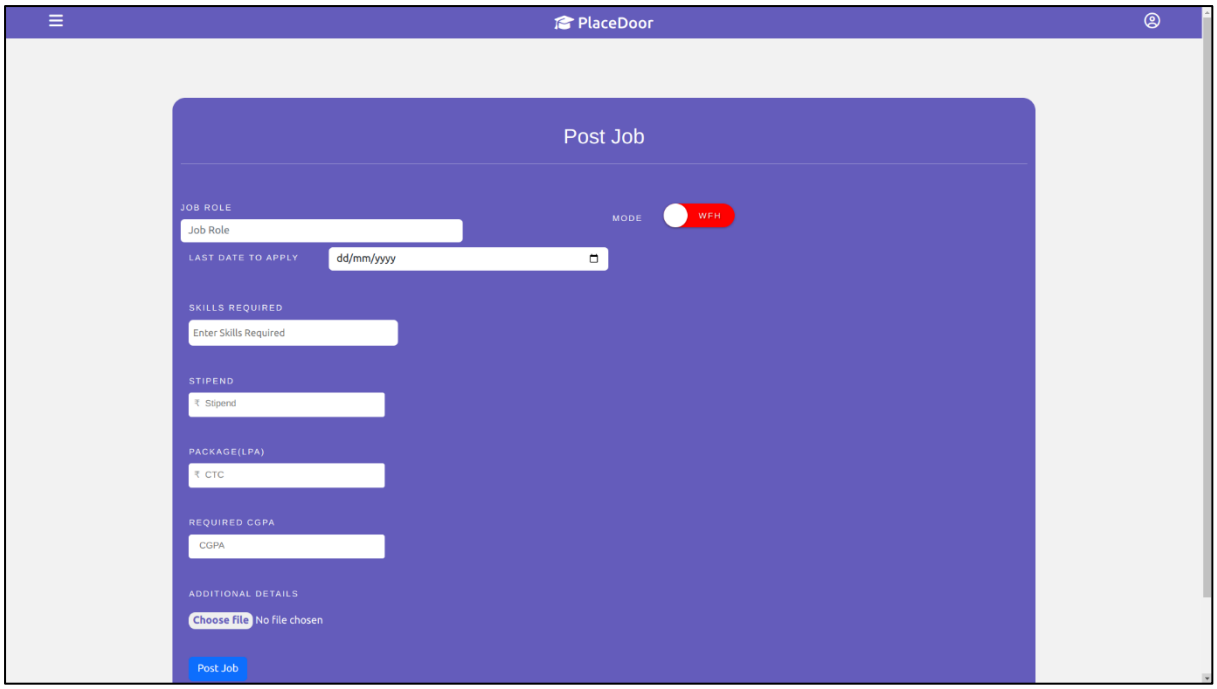


Login Page

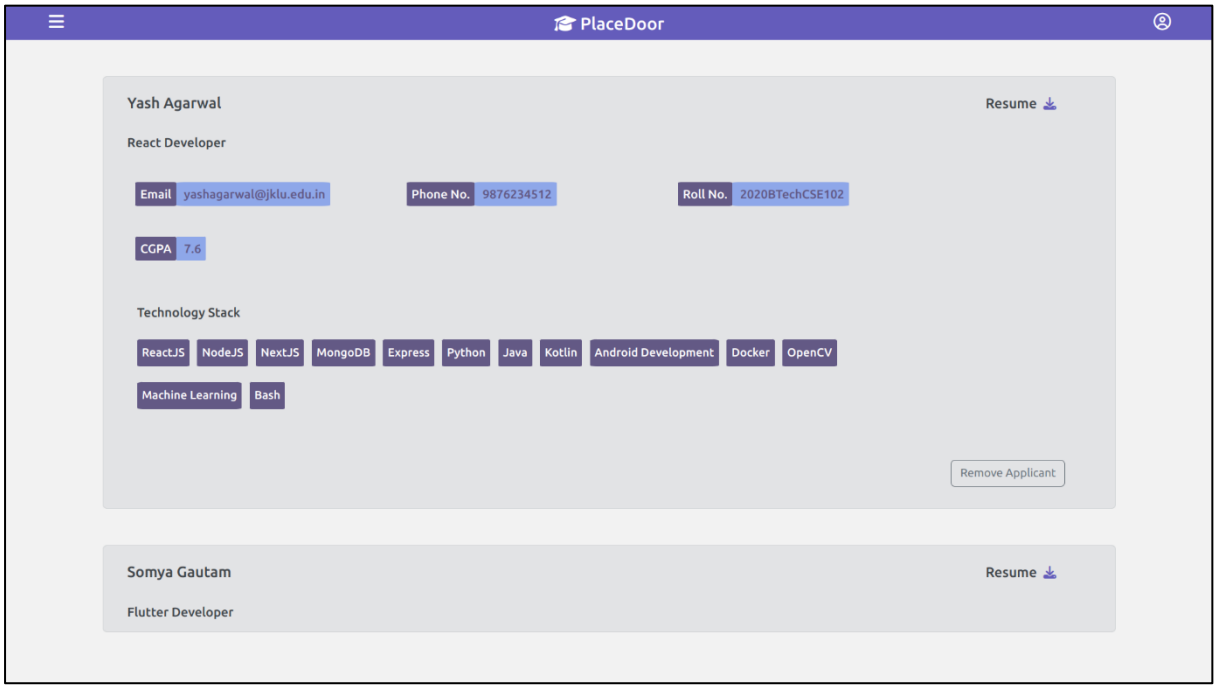


User Details Update Page

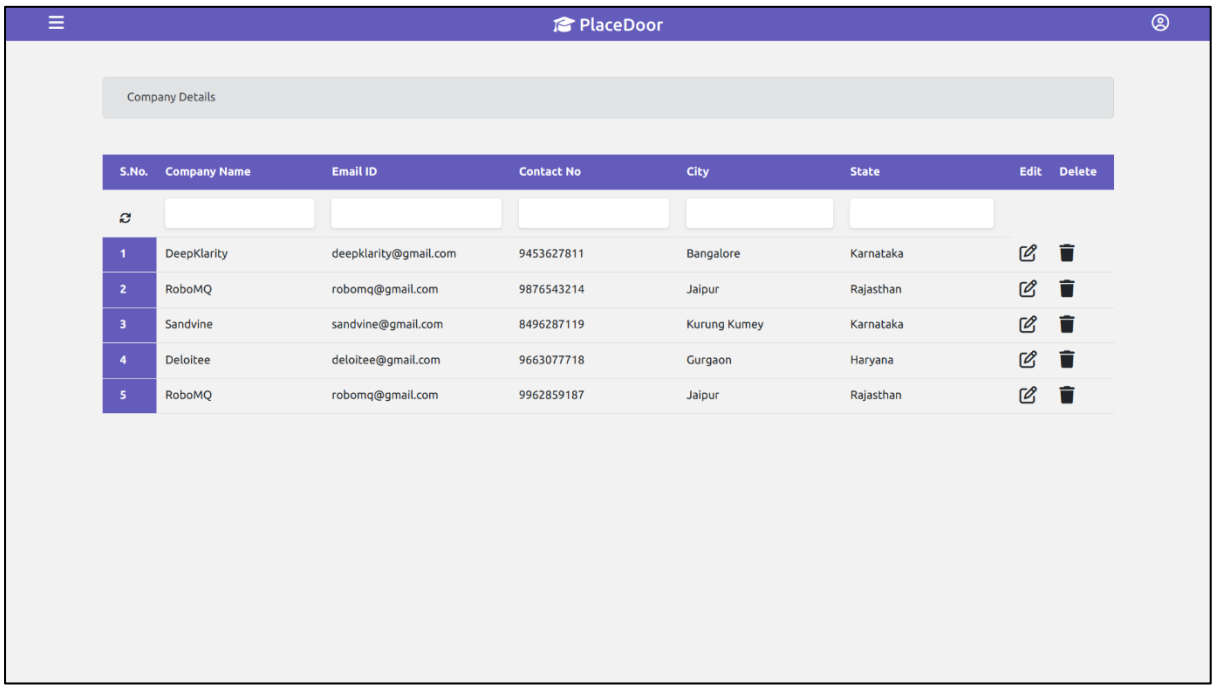
Posted Jobs page for students



Post Job form for companies



View Applicants Page for companies



View Companies Page for admin

2. UML DIAGRAMS

UML, or the "Unified Modelling Language," is a standardized graphical notation utilized in software engineering to visually depict different facets of a system. UML diagrams offer a standardized approach for visually modelling, documenting, and communicating various components of software systems, including their structure, behavior, and interactions. Examples of common UML diagrams include use case diagrams, class diagrams, activity diagrams, and sequence diagrams, each serving a specific purpose in modelling different aspects of a software system. UML diagrams are a powerful tool for visualizing, analysing, and designing complex software systems, aiding in effective communication, comprehension, and documentation of software projects.

Different UML made for this project are as follows:

2.1 ER Diagram

An entity-relationship diagram (ERD) is a data modelling technique that graphically illustrates an information system's entities and the relationships between those entities. An ERD is a conceptual and representational model of data used to represent the entity framework infrastructure. The elements of an ERD are Entities, Relationships, Attributes.

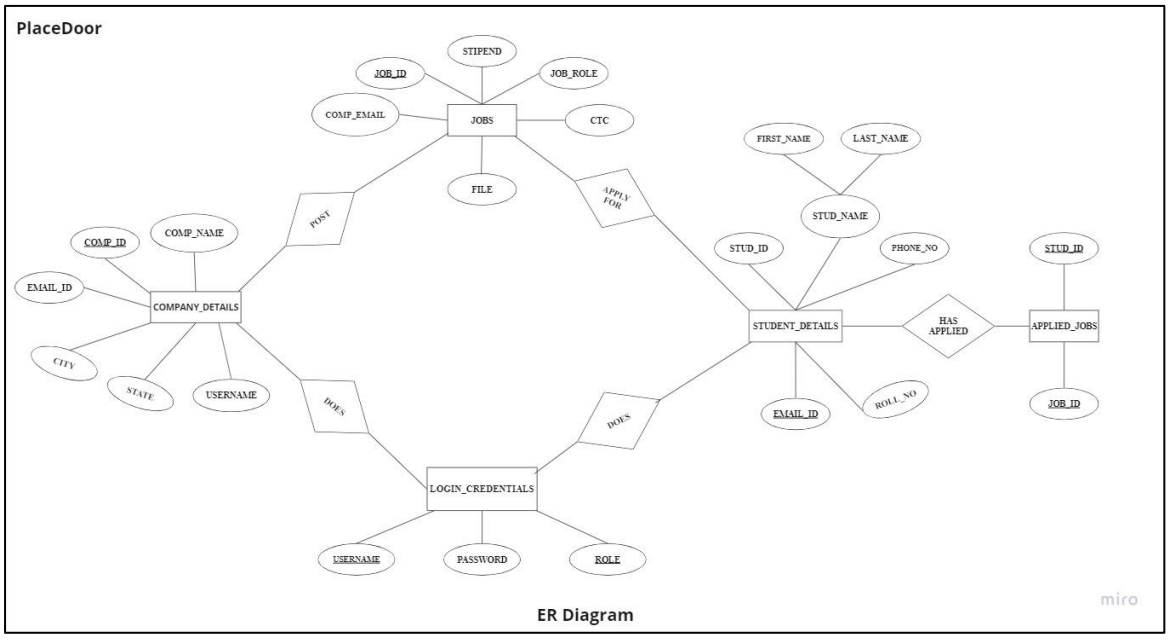


Figure 1 ER Diagram

2.2 Dataflow Diagram

A data flow diagram (DFD) maps out the flow of information for any process or system. It uses defined symbols like rectangles, circles and arrows, plus short text labels, to show data inputs, outputs, storage points and the routes between each destination. Data flowcharts can range from simple, even hand-drawn process overviews, to in-depth, multi-level DFDs that dig progressively deeper into how the data is handled. They can be used to analyse an existing system or model a new one.

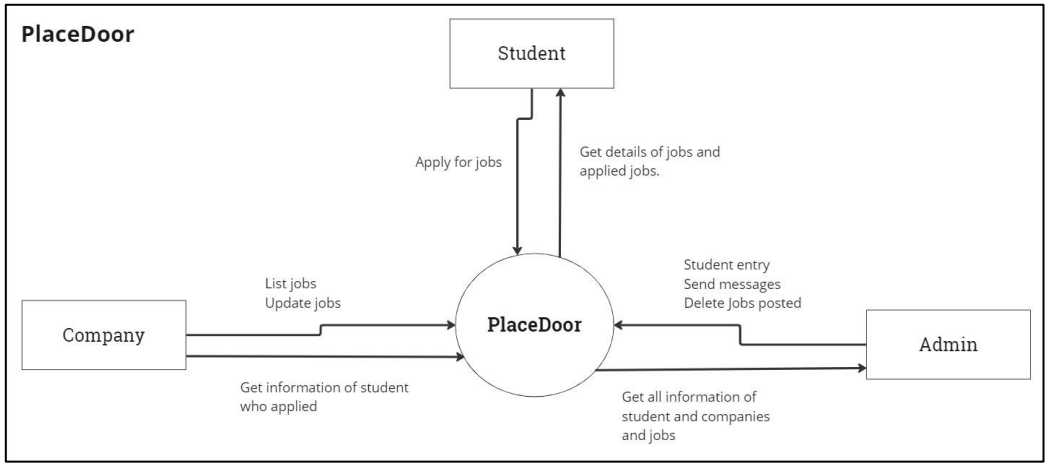


Figure 2 Level 0 Dataflow Diagram

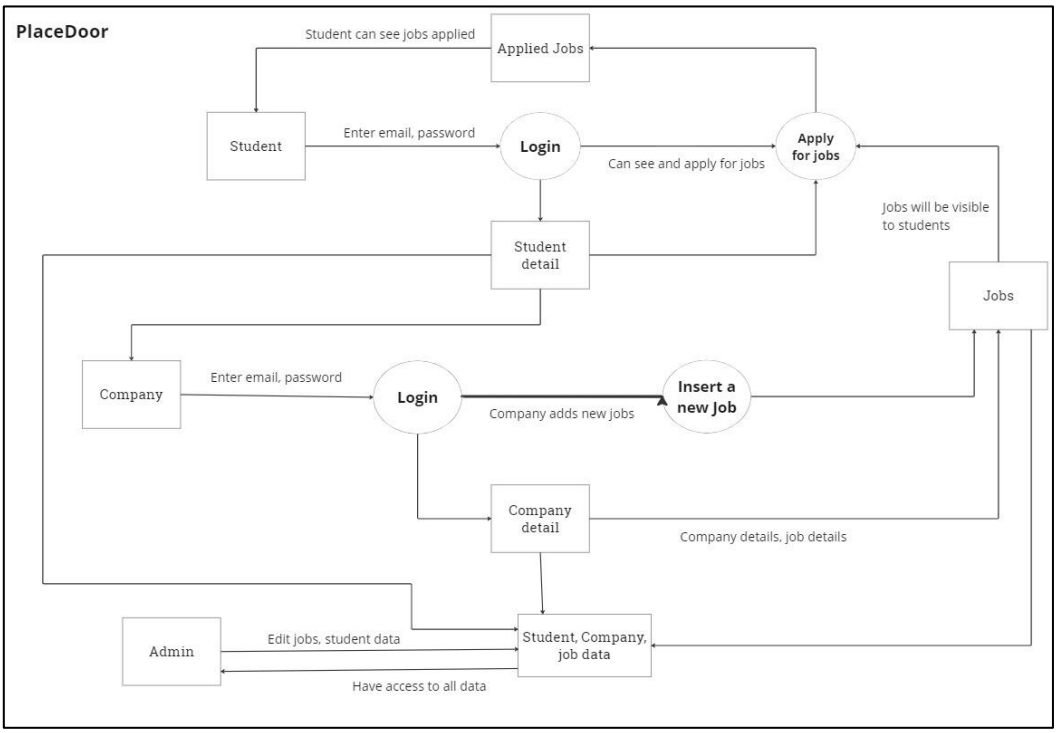


Figure 3 Level 1 Dataflow Diagram

2.3 Module Diagram

The process of breaking down a software into multiple independent modules where each module is developed separately is called Modularization. Effective modular design can be achieved if the partitioned modules are separately solvable, modifiable as well as compliable.

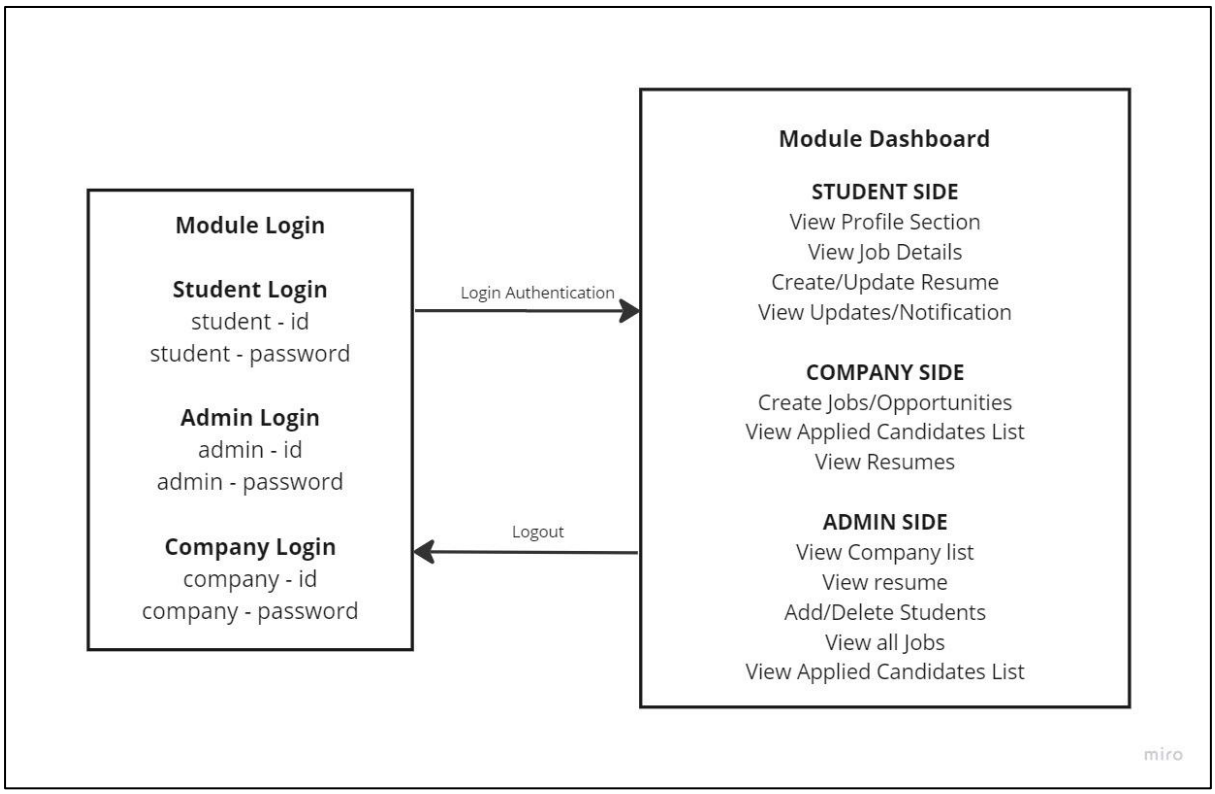


Figure 4 Module Diagram

2.4 Structure Diagram

A structure chart (SC) in software engineering and organizational theory is a chart which shows the breakdown of a system to its lowest manageable levels. They are used in structured programming to arrange program modules into a tree. Each module is represented by a box, which contains the module's name. The tree structure visualizes the relationships between modules.

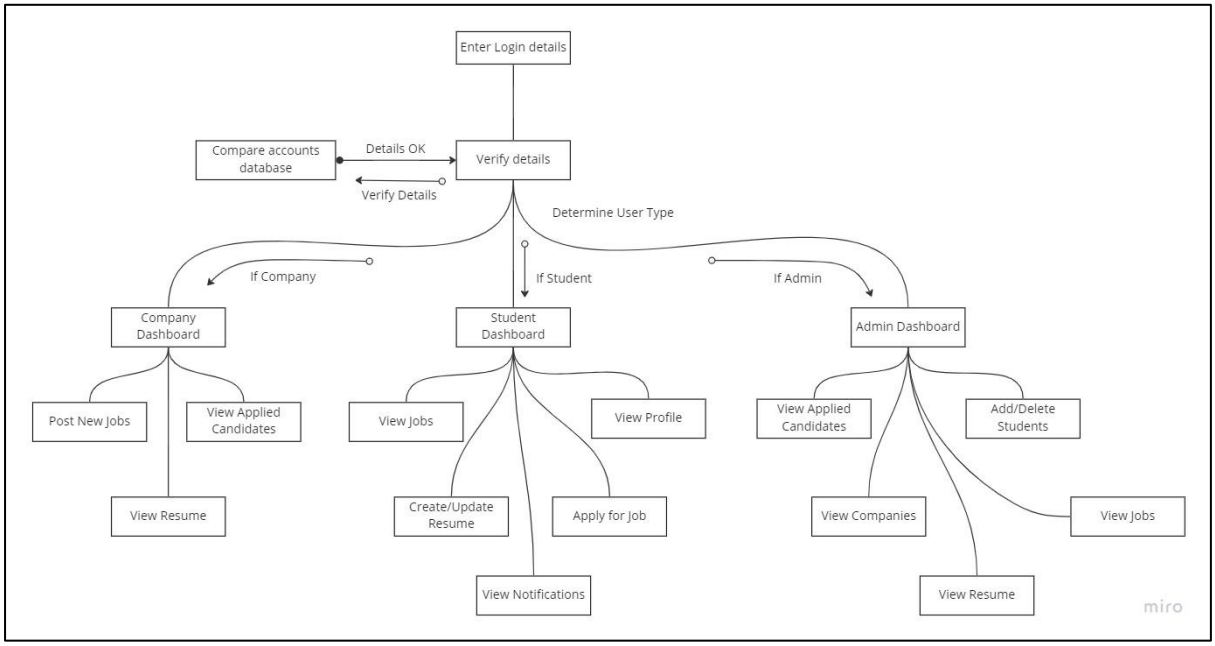


Figure 5 Structure Diagram

2.5 Flowchart

A flowchart is a type of diagram that represents a workflow or process. A flowchart can also be defined as a diagrammatic representation of an algorithm, a step-by-step approach to solving a task. The flowchart shows the steps as boxes of various kinds, and their order by connecting the boxes with arrows. This diagrammatic representation illustrates a solution model to a given problem. Flowcharts are used in analysing, designing, documenting, or managing a process or program in various fields.

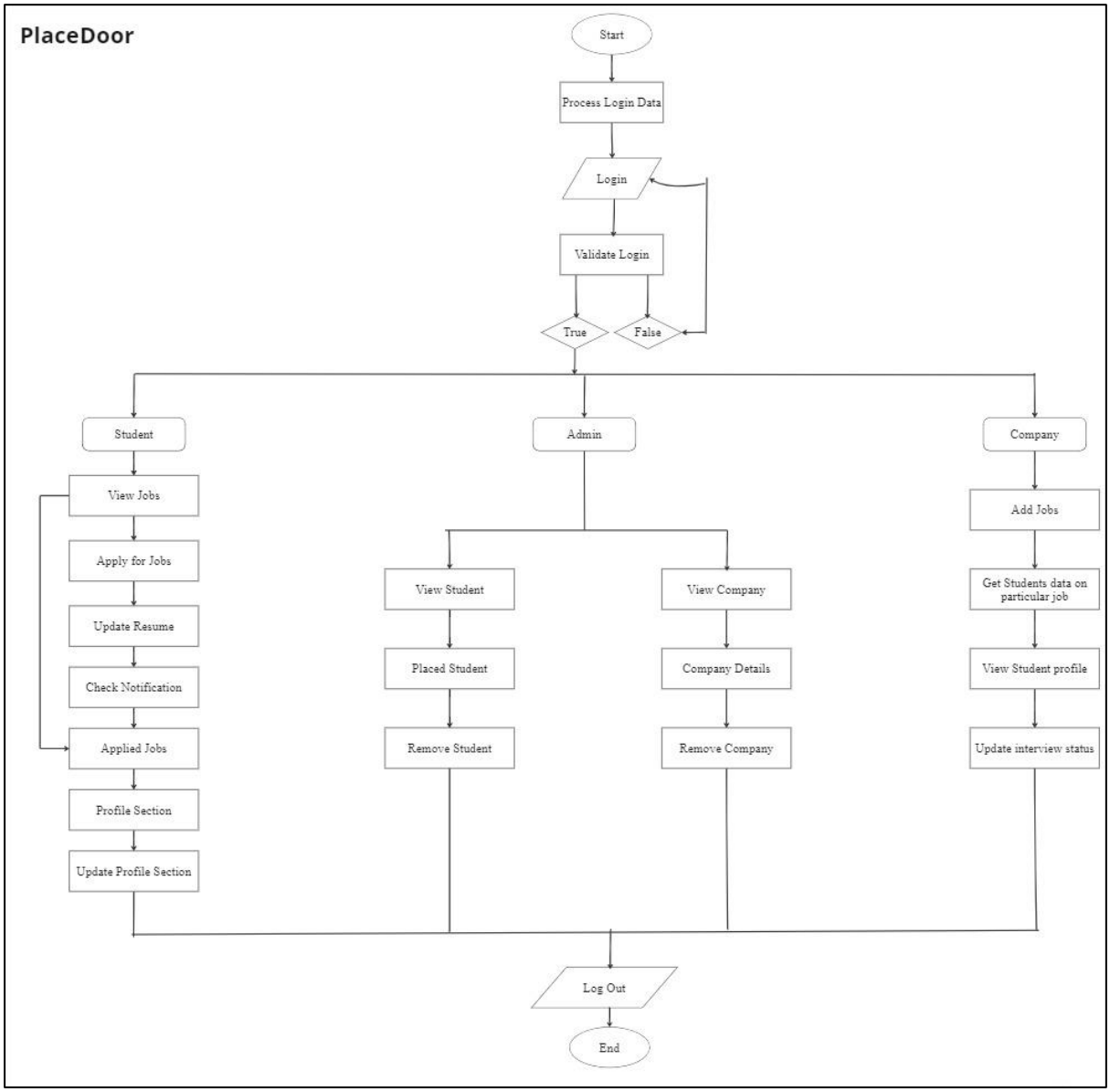


Figure 6 Flowchart

2.6 Use case Diagram

A use case diagram is a way to summarize details of a system and the users within that system. It is generally shown as a graphic depiction of interactions among different elements in a system. Use case diagrams will specify the events in a system and how those events flow, however, use case diagram does not describe how those events are implemented.

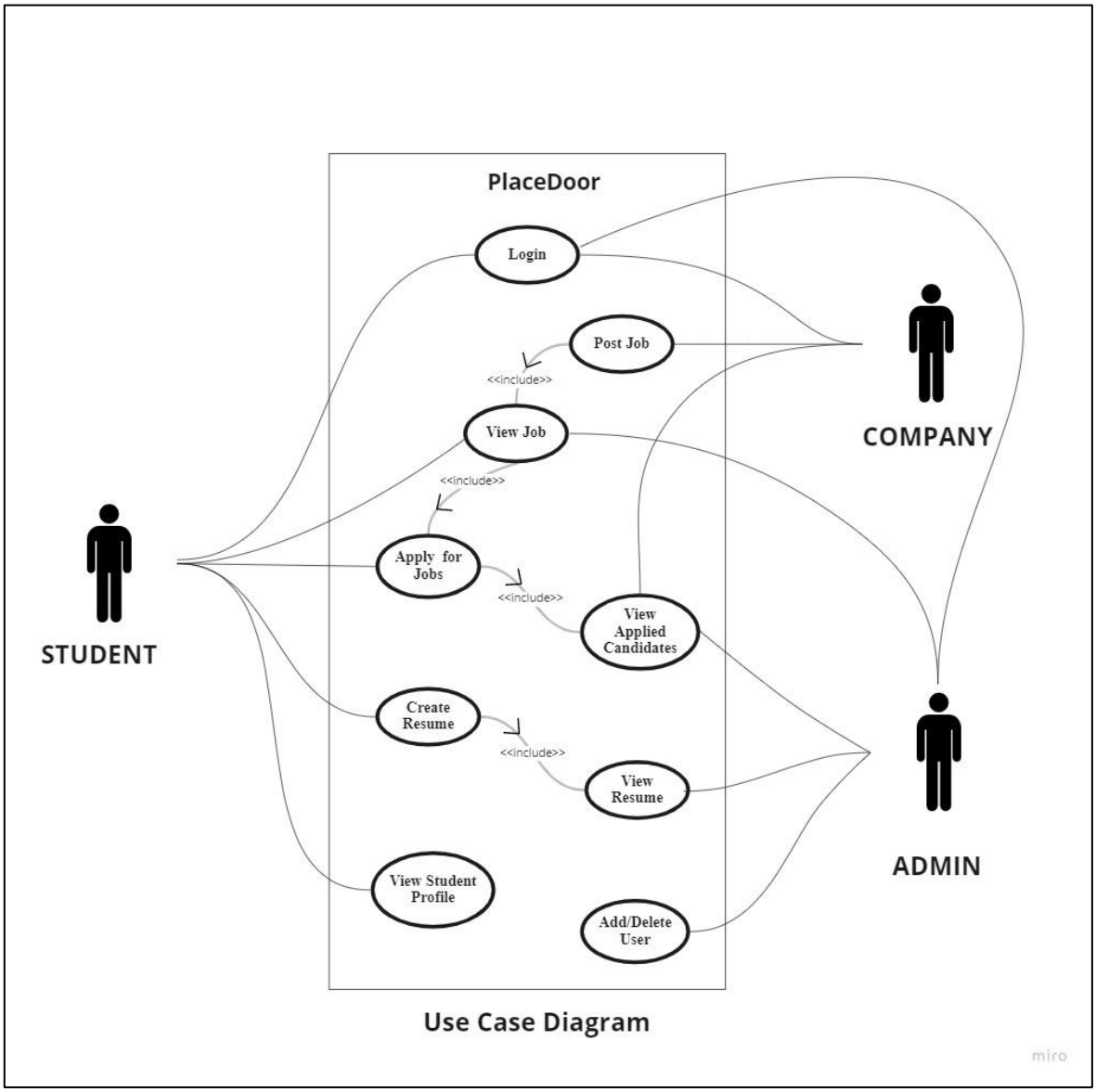


Figure 7 Use Case Diagram

2.7 Sequence Diagram

A sequence diagram is a valuable type of interaction diagram that provides insights into the collaborative behavior and order of interactions among a group of objects. This diagram serves as a valuable tool for software developers and business professionals alike, aiding in the comprehension of system requirements for new projects or documenting existing processes. Sequence diagrams, also referred to as event diagrams or event scenarios, offer a visual representation of the sequence of interactions and messages exchanged between objects or components within a system. By utilizing sequence diagrams, developers and business professionals can gain a clear understanding of how objects collaborate, the order in which interactions occur, and the flow of information among objects or components.

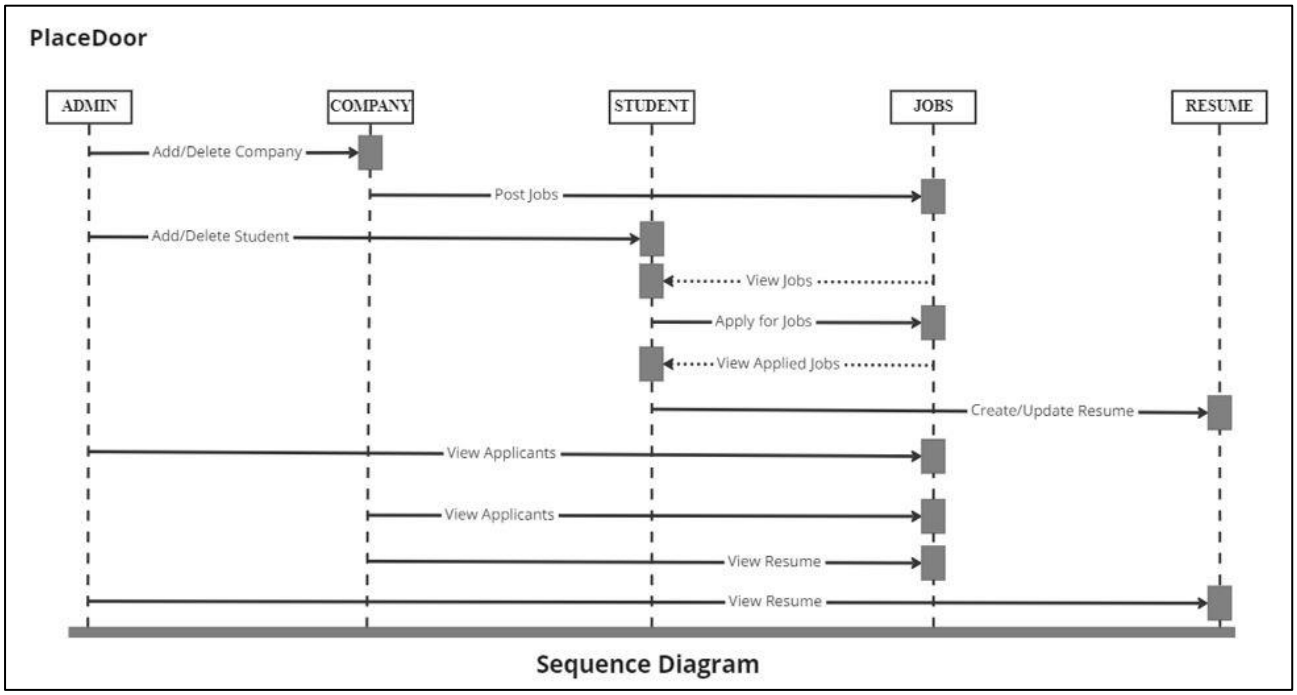


Figure 8 Sequence Diagram

2.8 Activity Diagram

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 type of flow control by using different elements such as fork, join, etc. Activity diagrams are not only used for visualizing the dynamic nature of a system, but they are also used to construct the executable system by using forward and reverse engineering techniques.

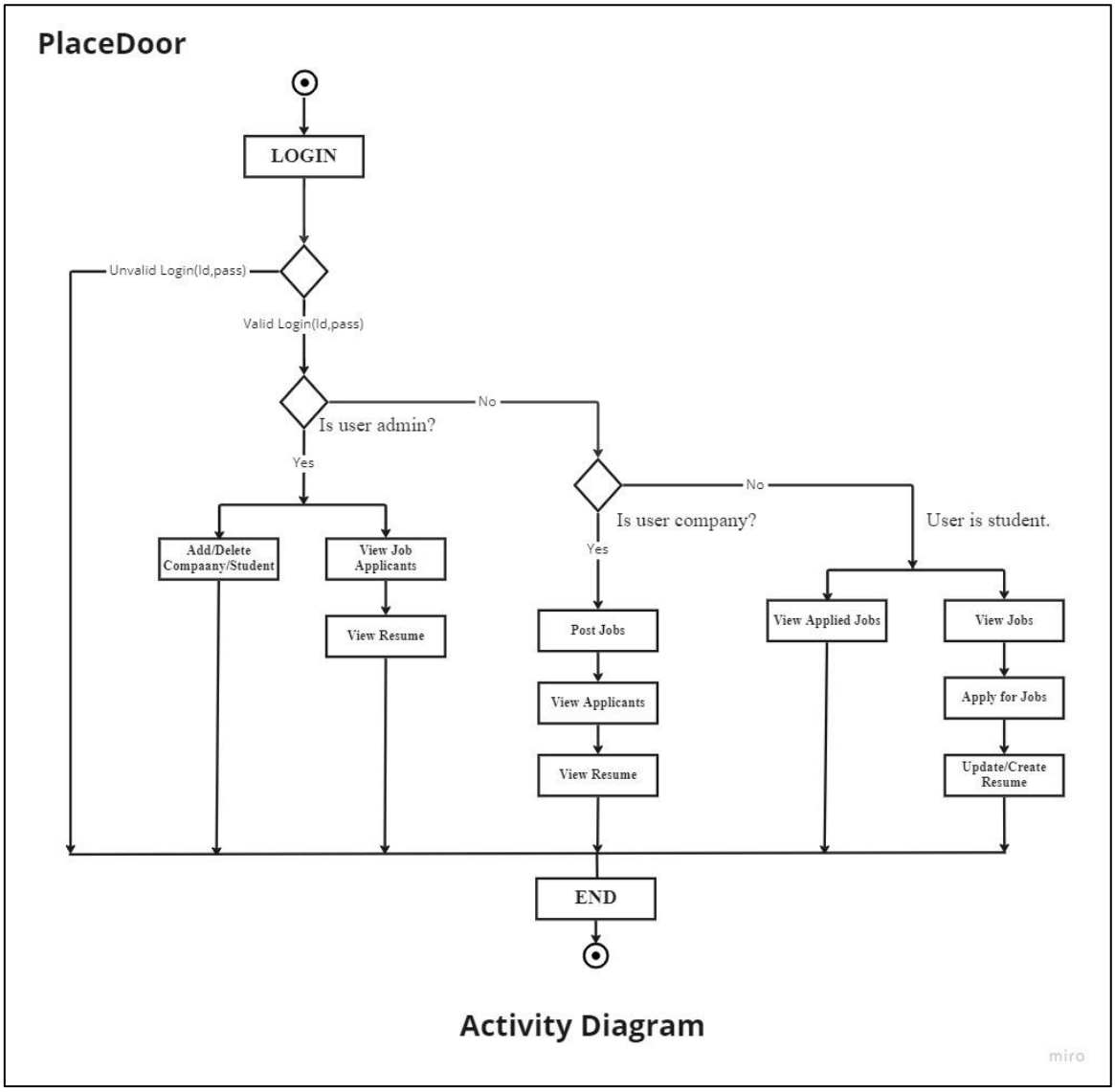


Figure 9 Activity Diagram

2.9 Class Diagram

Class diagram is a static diagram. It represents the static view of an application. 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. The 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.

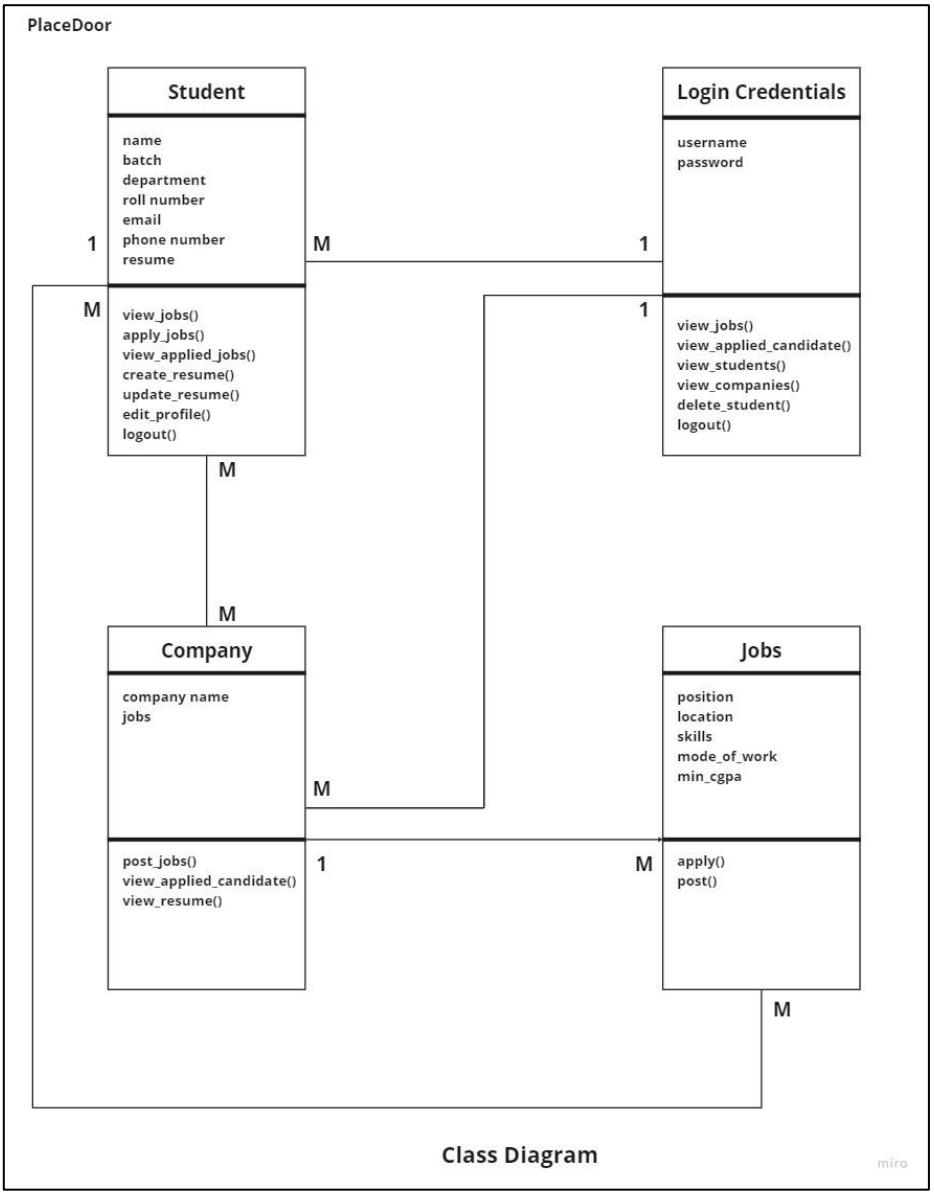


Figure 10 Class Diagram

3. SOFTWARE DEVELOPMENT

3.1 Tools and Technologies

In order to enhance our project's efficiency and coordination among team members, we employed a range of tools that facilitated our development process. These tools were carefully selected to streamline our workflow, improve communication, and optimize our productivity. The following are some of the key tools that we utilized:

GitHub

It is a widely used and highly popular platform among software developers for version control and collaborative code management. It provides a robust and intuitive interface for creating and managing code repositories, which allows teams to track changes, merge code changes, collaborate on the same codebase, and manage issues and bugs efficiently. With its comprehensive set of features, including branching, pull requests, code review, and automated workflows, GitHub enables teams to work seamlessly and cohesively on software development projects.

VS Code

Visual Studio Code is a lightweight yet feature-rich source code editor that has gained widespread popularity among developers. It offers an integrated development environment (IDE) with powerful code editing, debugging, and version control integration capabilities. VS Code supports a wide range of programming languages and provides an extensive ecosystem of extensions that can enhance the development workflow, including code snippets, IntelliSense, debugging tools, and integrated terminals. Its user-friendly interface and extensibility make it a preferred choice for many developers for coding and debugging tasks.

Notion

It is a versatile all-in-one productivity tool that offers a wide range of functionalities for teams. It serves as a unified platform for creating and organizing content in various formats, such as documents, databases, project boards, and wikis. Notion's flexibility and customizability make it a powerful tool for project documentation, task management, knowledge bases, and team collaboration. Teams can use Notion to create shared databases, assign tasks, set deadlines, collaborate on documents, and track progress, all in one centralized location.

Trello

It is a visual project management tool that uses boards, lists, and cards to help teams organize and track tasks in a visual and intuitive manner. Teams can create boards for different projects, create lists to represent stages or categories, and create cards for individual tasks. Trello's drag-and-drop interface, labels, due dates, comments, and activity tracking make it easy to manage tasks, assign responsibilities, and track progress

MongoDB Atlas

It is a cloud-based database hosting and management platform designed specifically for MongoDB, a popular NoSQL database. It provides managed hosting, automated backups, monitoring, and security features, making it a reliable and scalable option for hosting MongoDB databases in the cloud. MongoDB Atlas simplifies the setup, management, and scaling of MongoDB databases, allowing teams to focus on building their applications without worrying about database management tasks.

Canva

It is a user-friendly web-based graphic design tool that offers a wide range of templates, customization options, and stock images to create professional-looking designs. With drag-and-drop functionality, real-time collaboration, and branding tools, Canva is ideal for team projects. It supports multiple file formats, offers photo editing and transparency tools, and has a comprehensive colour palette.

Miro

It is a collaborative online whiteboard platform that enables teams to collaborate visually in real-time. Miro provides features such as sticky notes, shapes, templates, comments, and integrations with other tools, making it a versatile platform for collaborative creativity, design thinking, and visual communication.

These tools collectively play a significant role in modern software development, providing teams with powerful capabilities for collaboration, code management, project management, database hosting, and visual communication. By leveraging the rich features and functionalities offered by these tools, software development teams can enhance their productivity, efficiency, and overall project success.

We leveraged a diverse array of web development technologies to create our web application, employing a comprehensive tech stack that encompassed various tools, frameworks, and libraries. These technologies were carefully chosen to meet the unique requirements of our project and deliver a robust and feature-rich web application. The following are some of the key web development technologies that we employed:

Front-end Technologies

To develop the user interface of our web application, we leveraged widely-used front-end technologies such as HTML5, CSS3, and JavaScript. Additionally, we utilized modern frameworks like React, Angular, or Vue.js to further enhance the user experience. By employing these cutting-edge technologies, we were able to create engaging and interactive web pages that offered a seamless user experience. Our use of HTML5, CSS3, and JavaScript allowed us to design responsive web pages that adapted well to different devices and browsers, ensuring a consistent experience for users across multiple platforms. The implementation of modern frameworks like React, Angular, or Vue.js provided us with advanced tools and libraries that enabled us to build dynamic and feature-rich user interfaces. These frameworks helped us to streamline the development process, optimize performance, and create visually appealing and highly functional user interfaces for our web application. Overall, the use of these front-end technologies and frameworks played a pivotal role in creating a modern and user-friendly interface for our web application, enhancing its usability and overall user experience.

Back-end Technologies

We incorporated a variety of robust back-end technologies, such as Node.js, along with relevant frameworks like Express, to power the server-side logic of our web application. These cutting-edge technologies provided us with the capability to seamlessly implement complex server-side functionalities, efficiently manage data storage and retrieval operations, and interact with databases in a streamlined manner. By leveraging these advanced technologies, we were able to ensure a smooth and efficient flow of data within our application, enabling us to deliver a high-quality user experience to our end users. Furthermore, the utilization of these back-end technologies enabled us to effectively handle concurrent user requests, optimize performance, and enhance the overall reliability and scalability of our web application.

Database Technologies

We leveraged the widely-used and powerful database technology MongoDB as the foundation for storing and managing the data in our web application. MongoDB provided us with a robust and flexible platform to design and implement efficient database schemas that met the specific requirements of our application. With MongoDB, we were able to securely store and retrieve data, ensuring data integrity and consistency throughout the application. Additionally, MongoDB's rich set of features and capabilities allowed us to implement advanced database operations such as indexing, query optimization, and data aggregation, optimizing the performance of our application and enhancing its overall efficiency. The utilization of MongoDB as our database technology empowered us to effectively manage and manipulate data within our web application, enabling us to provide a seamless and reliable experience to our users while maintaining the highest standards of data security and integrity.

3.2 Code Inspection

In our project, we followed a rigorous and systematic approach to ensure code quality, and one of the key practices we implemented was code inspection. Code inspection, also known as code review or peer review, was an integral part of our software development process. It involved a collaborative effort where experienced developers thoroughly examined the source code to identify and correct any defects, errors, or other issues.

During the code inspection process, our team diligently reviewed the code for adherence to coding standards, best practices, design patterns, and other established guidelines. We checked for syntax errors, logic errors, performance bottlenecks, security vulnerabilities, and other coding mistakes. We also assessed the documentation, code organization, readability, and maintainability of the code.

Furthermore, we focused on ensuring that the code met the functional requirements of the software project. We carefully reviewed the code against the project's specifications and requirements, verifying that all functionalities were implemented correctly and that they met the desired outcomes. The code inspection process helped us to improve the overall quality of our codebase by identifying and fixing defects early in the development cycle. It promoted consistency and adherence to coding standards, and facilitated knowledge sharing among team members.

4. SOFTWARE TESTING

Software testing is a critical aspect of software development that plays a crucial role in ensuring the quality, reliability, and functionality of software applications. It involves the systematic and rigorous process of identifying and fixing defects, errors, and vulnerabilities in software to ensure that it performs as intended and meets the requirements of end-users. The importance of software testing cannot be overstated, as it contributes to the overall success of a software project in numerous ways.

First and foremost, software testing helps in identifying and fixing defects and errors early in the development process, preventing them from escalating into more significant issues in the later stages. This helps in reducing the overall cost of development, as fixing issues early is less time-consuming and less expensive compared to fixing them after the software has been deployed. Software testing also helps in enhancing the user experience by ensuring that the software meets the expectations and requirements of end-users. Through thorough testing, usability issues, user interface glitches, and other user-centric concerns can be identified and addressed, resulting in a more user-friendly and intuitive software application.

Additionally, software testing is essential in ensuring the security and robustness of software applications. Testing helps in identifying vulnerabilities, loopholes, and other security risks, which can be addressed before the software is deployed to production, thereby reducing the risk of security breaches, data leaks, and other security incidents.

As part of our project, we conducted a comprehensive and meticulous set of test cases, including crucial edge cases, to ensure the robustness and optimal functionality of the Placedoor application, while providing a seamless and satisfying user experience. We carefully designed and executed various test cases to thoroughly evaluate the system and validate its performance. Some examples of the test cases we performed include:

ID	Test Case Objective	Pre-requisite	Steps	Input	Expected Result	Actual Result	Status
TC-01	Check whether the website is browser independent.	Multiple browsers installed in the testing machine.	Open the website URL in multiple browsers.	-	Website opens in all the browsers tried.	Website opens in all the tested browsers.	Pass
TC-02	Check login functionality of the website.	The user should have a valid email ID.	Click on the “Sign in”.	Enter valid credentials and verify successful login	Successful login and redirection to the appropriate page.	The user was authenticated successfully.	Pass
TC-03	User data is sent to the Mongodb database.	Project should have added to Mongodb project.	Establish a connection to the MongoDB database.	MongoDB connection details (host, port, credentials)	User data shows up on the Mongodb console.	Successful insertion of the test user data into the MongoDB collection	Pass
TC-04	Create and post job with accurate job details.	Access to the website for job posting	Navigate to the job posting page.	Fill out the required job details accurately	Successful creation and posting of the job,	The job should be searchable and visible on the website	Pass
TC-05	User can easily withdraw application from applied job	User account with applied job.	Sign in and then go to applied jobs section from menu.	Click withdraw button in applied job card.	User application should be removed from that job	User application is removed and that job card is no longer visible.	Pass
TC-06	User can update their profile details.	The user must be able to edit and save changes to their profile details.	Update the required fields and any other desired information.	The desired updates or changes to the profile information, which may include adding, removing, or modifying existing information.	The updated profile details should be saved successfully to the database or storage system.	The updated profile information should be displayed correctly in the user's account settings or profile page.	Pass
TC-07	Job seeker can view available jobs.	The website or application has a section or page dedicated to available jobs.	Click on a job to view its details.	Entering a search query for a specific job title, filtering job listings by education, or other criteria.	View a list of all available jobs, with each job listing displaying the job title, company name, location, and a brief description.	Job seeker should be able to view a list of available jobs that match their search criteria.	Pass

5. CONCLUSION

Placedoor has effectively addressed the challenges of manual processes and administrative overhead, providing real-time updates, improved communication channels, and advanced features for students, employers, and administrator.

Our team has successfully developed a comprehensive placement management system that empowers students to apply for jobs, allows companies to add job listings, and enables administrators to oversee the entire process seamlessly while making necessary changes as needed. The system provides students with a user-friendly platform to manage their applied jobs, while companies can easily track the number of students who have applied in an organized manner. The system also offers administrative capabilities that allow administrators to efficiently monitor and manage the entire placement process. With our Placedoor, students can easily browse and apply for job opportunities offered by companies, making the job application process more convenient and efficient.

Companies can add and manage job listings, ensuring accurate and up-to-date information for potential applicants. Administrators have a comprehensive view of the entire process, allowing them to monitor job applications, review job listings, and make necessary changes or updates as required. The system has demonstrated its user-friendliness, scalability, and potential for further enhancements, making it a comprehensive solution for modern placement management in our college. The project has also contributed to the technological advancement of our college and has set a benchmark for future projects in the field of placement management.

As we move forward, regular updates, maintenance, and continuous improvements can further enhance the performance and functionality of the placement management system. Additionally, seeking feedback from users and incorporating their suggestions can help in refining the system and ensuring its alignment with the evolving needs and requirements of our college.

In conclusion, the Placedoor project has laid the foundation for further advancements in the field of placement management, benefiting our college and setting a standard for technological solutions in the domain of placements.

6. FUTURE SCOPE

The future scope of a project is an important consideration as it outlines potential areas of development, expansion, and improvement beyond the current project completion. In this section, we will discuss the potential avenues for further enhancement and growth of the project, considering emerging technologies, evolving user requirements, and changing market trends. We will explore possible future directions, features, and functionalities that could be added to the project to further elevate its value, impact, and relevance. Some major future scopes are:

Advanced analytics and reporting: Incorporating advanced analytics and reporting capabilities into your placement management system could provide valuable insights and data-driven decision-making for both students and administrators. This could include tracking and analysing placement trends, student performance, and employer feedback, among other metrics, to continuously improve the placement process.

Job matching: AI algorithms can analyse job requirements and student profiles to automatically generate job matches based on skills, qualifications, and preferences. This can help students discover relevant job opportunities that align with their skills and interests, and increase the chances of finding the right job fit.

Chatbots for student queries: AI-powered chatbots can handle common student queries related to the placement process, such as job application status, interview preparation tips, and FAQs. Chatbots can provide instant responses, reduce the load on administrators, and offer round-the-clock support to students.

Predictive analytics: AI-driven predictive analytics can analyse historical placement data, job market trends, and student performance to provide insights and recommendations for improving the placement process. For example, it can identify high-demand skills or industries, forecast job market trends, and suggest strategies for better student placements.

References

Some of the references are:

- Nishadha (2022) *UML diagram types: Learn about all 14 types of UML diagrams*, *Creately Blog*. Available at: https://creately.com/blog/diagrams/uml-diagram-types-examples
- Reggio, G., Leotta, M., Ricca, F., & Clerissi, D. (2013, October). What are the used UML diagrams? A Preliminary Survey. In EESSMod@ MoDELS (pp. 3-12).
- *Software testing overview*. Tutorials Point. (n.d.). from https://www.tutorialspoint.com/software_engineering/software_testing_overview.htm
- Tilkov, S., & Vinoski, S. (2010). Node. js: Using JavaScript to build high-performance network programs. *IEEE Internet Computing*, *14*(6), 80-83.
- *GitHub.com help documentation*. GitHub Docs. (n.d.). 2023, from https://docs.github.com/en
- *The developer Data Platform*. MongoDB. (n.d.), from https://www.mongodb.com/
- React. (n.d.), from https://react.dev/
- *The visual collaboration platform for every team: Miro*. https://miro.com/. (n.d.). Retrieved April 12, 2023, from https://miro.com/
- Everett, G. D., & McLeod Jr, R. (2007). Software testing. Testing Across the Entire.

Appendix

GitHub link:

https://github.com/boyz-3000/place-door

Miro Board link:

https://miro.com/app/board/uXjVPhOND1w=/?share_link_id=190916562513

PPT link:

https://www.canva.com/design/DAFgSrqwYGY/ZtVRiicccWu-aRd1ORqbDA/edit?utm_content=DAFgSrqwYGY&utm_campaign=designshare&utm_medium=link2&utm_source=sharebutton

Project Poster: