Hirehub:

Problem definition:  
The existing universities mostly rely on mails and text messages for communication of placement management tasks. The motivation is to design a streamlined web application for better processing, communication and co-ordination between placement cell and students. An user friendly and handy interface can provide a solution to the existing problem and also fulfill needs of users and sets a long-tern success plan for career growth

Project Overview:

The purpose of the web application is to give universities and students a platform to manage businesses and students on campus. Due to the lengthy procedures involved in college placements, Hirehub's concept aims to facilitate communication between placement cells and students.

It provides insightful information regarding ongoing processing and companies to support data-driven decisions aimed at boosting student engagement and the placement process. It also simplifies the placement process in its entirety, students, and members of the placement cell physical labor and paperwork.  
  
2.1 Stakeholders  
  
One of the most important components of the product is its stakeholders. It is essential to take into account all stakeholders, expectations, and issues for a product to flourish.   
Although there are numerous indirect stakeholders, the following is a summary of the direct stakeholders involved in the application and how they relate to the Hirehub.

|  |  |
| --- | --- |
| **Stakeholder** | **Role** |
| University Employees | It will help to organize things in a better way and get data stored in a particular software instead of tons of excel files. |
| University Students | They will be benefitted in a way to be aware of the placement processes and companies. |
| Developers | They are the ones who are responsible for managing the development, enhancing and troubleshooting. |

2.2 Features

|  |  |
| --- | --- |
| Feature | Description |
| Authentication | Students and Placement cell members can register themselves on Hirehub and the existing user can login to the WebApp. |
| Opportunities | Students can view and apply for the opportunity based of certain rules like deadline and matching skills. Placement cell can post, edit, delete and view the opportunity. |
| Search Filters | Students and Placement cell members can filter out the opportunities based on location package, skills and process dates. |
| Process Information | Students can view the ongoing rounds and shortlisted students of a particular opportunity. Placement cell can post and view the processes in a opportunity. |
| Profile setup | Students and Placement Cell can built and edit their profile by filling out necessary information |

1.3 Hardware requirements

The application has minimal hardware requirements because it is web-based. To make sure the program functions well, it is advised to adhere to following guidelines.

|  |  |
| --- | --- |
| **Category** | **Specification** |
| RAM | 8 GB or above |
| Processor | Intel i5 |
| Storage | 128 GB or above |
| I/O Devices | Keyboard, Mouse |

Software Specifications:

|  |  |
| --- | --- |
| Component | Software/Tools used |
| Designing | Figma |
| Frontend Framework | React.js |
| UI Library | Material UI |
| Backend Framework | .Net Core |
| Database | Microsoft SQL Server |
| Api testing | Swagger |
| Frontend IDE | Visual Studio Code |
| Backend IDE | Visual Studio |

|  |  |
| --- | --- |
| Component | Technologies Used |
| Frontend Language | Javascript(ES6+) |
| State Management | React hooks |
| Backend Language | C# |
| Database Language | SQL |

Literature Survey:

Campus placement management systems are necessary tools for enabling communication between colleges and universities and students. These software provide a consolidated platform for application submissions, communication, and event management, improving the university employment placement process. The need of this summary of the literature is to look into current approaches, technological and salient features, and prospects related to campus placement management systems.

* Existing systems and technologies:

1.1 Traditional systems  
  
The current placement method is based on haphazard communication, which results in emails and messages being sent slowly and inefficiently. Students have to apply for the interested companies through google forms and the data was managed by placement cell members through excel sheets. To overcome these traditional practices, HireHub is made to provide a streamlined web app which helps the universities to carry out these processes more effectively.

1.2 Online Platforms

Online platforms have been a feasible option for more effective organizing campus placements using digitalization. Numerous features, catered to the requirements of colleges, students, and recruiters alike, are available on these platforms, such as application tracking systems, resume databases, job posting portals, and communication tools. Many such similar works in this domain are as follows:

* SmartRecruiters hiring platform is designed to help every company recruit top talent. It has all the tools you will need to attract, select, and hire the right people at the right time.
* Yello**:** focuses on campus recruiting and offers solutions for talent acquisition, including interview scheduling and candidate engagement.
* JobTeaser**:** is a European platform connecting students with employers for internships and job opportunities. It often integrates with university career services.

1.3 Technological Stack

Frontend(Reactjs)  
  
It is well-scripted JS user interface library that is used in development of frontend.

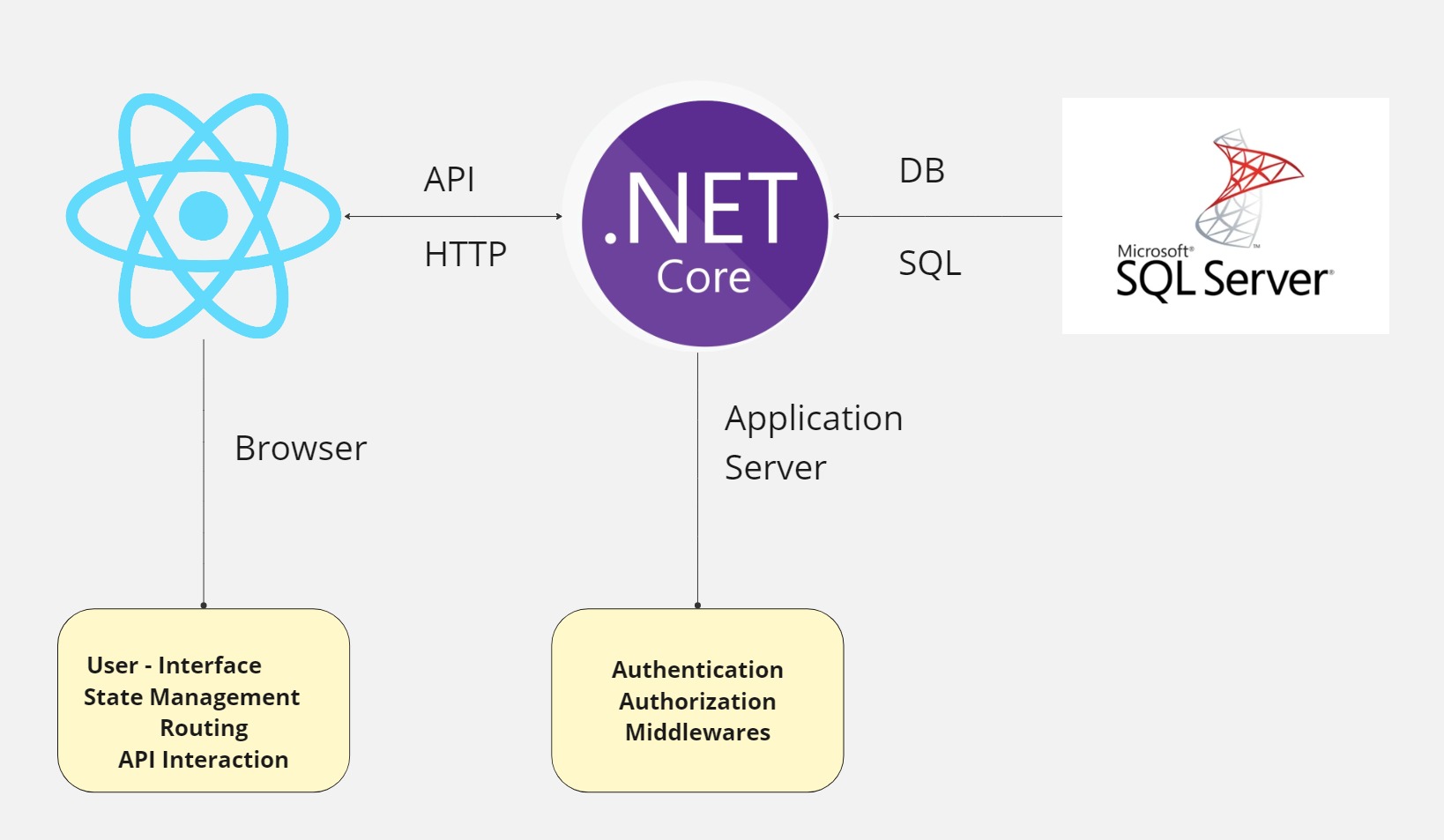
It is made up of different parts, sections, and user interface elements that communicate with users. In control of handling user interactions, displaying the user interface, and sending requests to the backend API.

Backend API (.Net Core)

It is used to develop backend APIs for creating web application. It offers endpoints for communicating with the database, processing business logic, and taking take of client requests. It consists of middleware, controllers, and services to handle incoming requests and produce relevant answers. It uses C# as its code language that effectively works on object-oriented fundamentals.

MS SQL Server

It is sed to store posts, application, statuses, and other pertinent information. Indexes, stored procedures, tables, and views are the tools that are used for effective organizing and manipulating data. ORM (Object-Relational Mapping) frameworks are used by the .net core API to connect with the database to access and manipulate data.



Methodology:

Purpose:

Hirehub comes with a purpose to provide a streamlined platform to universities and students and make the placement management more optimized and effective. The goal behind making Hirehub – A placement management system is to bridge the gap of communication and collaboration between job givers and job seekers.

Essential requirements:

* 1. Authentication and profile

Both the users (placement cell and students) are able to register themselves and log in to the web page securely. Moreover, they can build their profile that includes their personal, demographic and academic information that further helps students to seek opportunities of their choice.

* 1. Opportunity Posting

Placement cell members can post company information in the form of an opportunity that student views. Students can filter out and can show interest in any opportunity based on their likings. They can also apply based on their profile and availabilities.

* 1. Process tracking

0

Non functional requirements:

Availability :

The application should be available to all the users 24\*7. In any case of any failure, the application should be available in 1 or 2 hours.

Performance

The application should be able to handle multiple users simultaneously and the application should run smoothly without being crashed.

The system should have a high-performance rate when executing the user's input and should be able to provide a response within a short time span and slightly more for highly complicated tasks and less complicated tasks.

Data Integrity

The data should be least redundant and more consistent across the distributed databases.

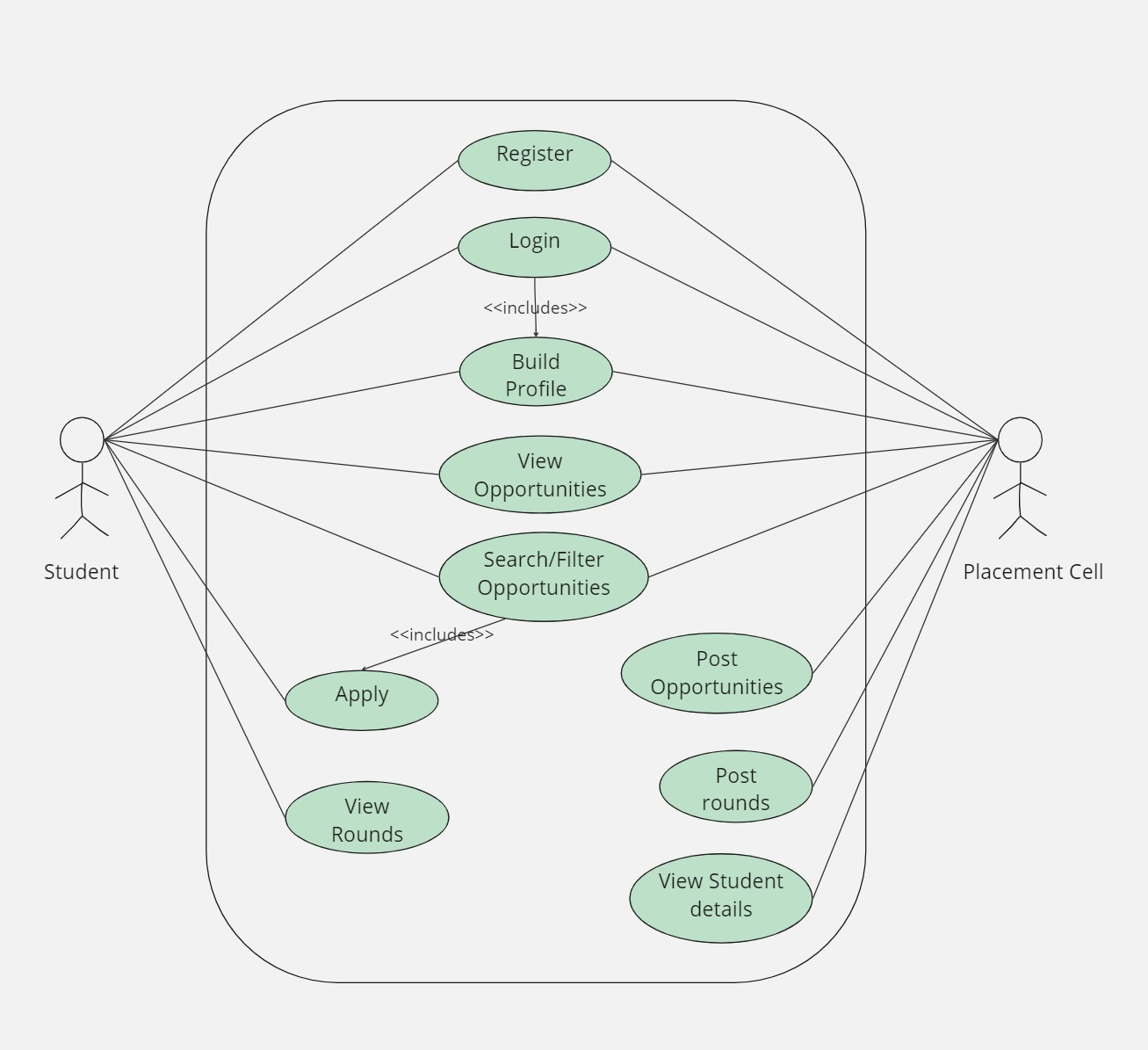
Speed

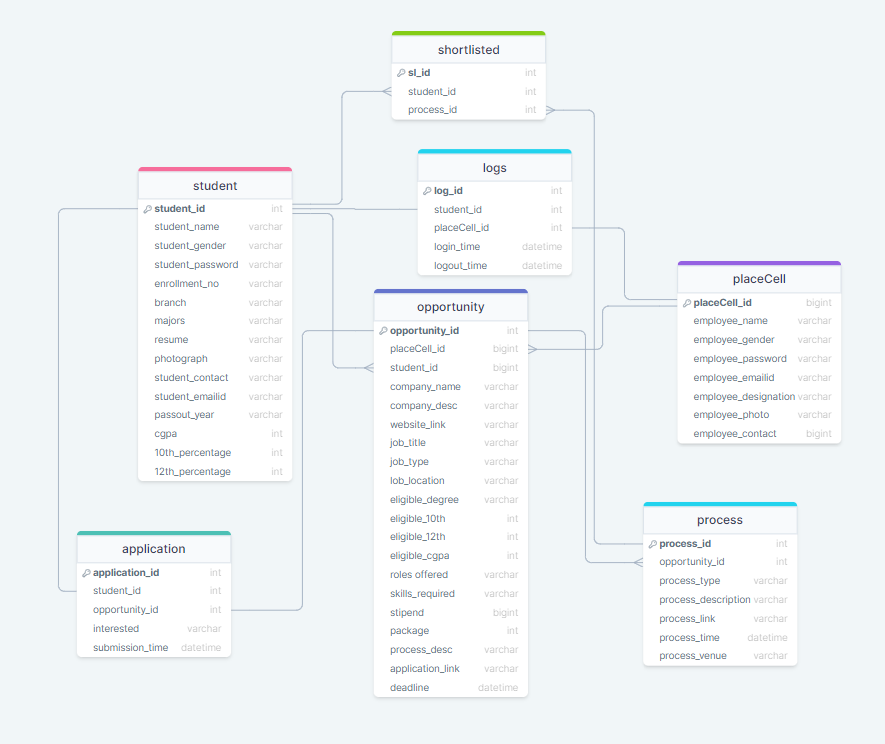
Security

- Data encryption for private and sensitive data, including login credentials.   
  
- Role-based access management to guarantee that certain functions and data are only accessible to authorized users.

Consistency

A systematic coding pattern should be followed throughout





DEVELOPER PHASE:

* Software development model

Depending on the project's requirements, one can choose from a number of software development lifecycle methods. Software development approaches include some of the following.

• Waterfall

• Agile

• Scrum

Scrum is a well-known equivalent for agile methodology. Agile updates progress through regular customer engagement. In Scrum, there are a lot of interactions among the team members. It is discovered that the scrum development technique is most appropriate for the creation of applications with intermediate complexity after comprehending the workings of each methodology.

* Frontend Implementation

The front-end part of the webapp has been made by using React.js.

1. Environment setup: Setting up required environments like npm, yarn and create-react-app as a first step to use Teact.js application.
2. UI/UX Designing: Made wireframes using Figma and used it as low fidelity prototype to replicate the design in react.js
3. Component development: Used functional based component for achieving different features using React.js component architecture.
4. State Management: Used react hooks (useState and useEffect) to manage the inputs and dataflow between the components.
5. Connecting with Backend: Used reactjs fetch() library to call all APIs and manage frontend with data driven approach.
6. Using session storage: Used session storage to store IDs and use it in cross functional components

* Backend development

1. Environment setup: Configured development environments by setting up NuGet packages in visual studio.
2. API design : Using.NET Cor, create RESTful API endpoints for a range of features, including application tracking, user management, job posting, and event management.

images

1. Business Logic: Executed CRUD operation on database using C# language and processed all the requests.
2. Repository pattern: Between an application's data access and business logic layers, we build an abstraction layer. We are encouraging a looser-coupled method of accessing our data from the database by leveraging it. In addition, the code is simpler to reuse and maintain.

* Database development

Database Design: Made schema understanding all the users and requirements considering entities, attributes, relationships, and normalization.

Stored procedures: Implemented logics understanding all the business requirements that is called by the APIs in backend.

**TESTING PHASE**

LEARNING OUTCOMES:

1. Technical proficiency:

* Tried implementing complex designs in Reactjs including modals, states and data fetching.

Learning the concepts of RESTful APIs, middleware, routing, validation and delving it into business use cases.

Enhanced skill in SQL that includes building schema, writing stored procedures, and query optimization.

Principles of software engineering

Understanding software development cycle and agile methods and applying it to the phases of development of the project.

Learning about code reusability, scalability, modularization and maintenance of software.

Problem Solving and troubleshooting

Developed problem solving skills by converting use cases into effective solutions using appropriate techniques and logic.

Enhanced troubleshooting skills while finding the root cause of any errors and bugs.

Continuous Learning and Adaptability:

Developed a passion for continuous learning by exploring the problems statements in real life world, recognizing it, and making the best use of technology to reach towards a solution.

* Real World Application

Applied the tecknical skills learnt and principals of software engineering to develop and provide solution to a real-world problem.

Developed a interactive user interface for students, placement cell, and administrators to seamlessly interact via the platform (HireHub).

The implementation of backend functionalities makes the platform to run through data driven approach and make it more responsive and scalable.

Bridges the communication gap between placement cell and students by putting everything on a single platform.

Minimizes the use of multiple platforms like emails, messaging, and making the process more efficient and organized.

Improves student engagement by user friendly interface, leading to higher rate of applications and job placement outcomes.

The statistics generated by application can help the placement cell make data driven decisions about posting jobs, process scheduling, and better offerings.

A long term success plan can be achieved by providing a comprehensive platform for career growth and help them to secure jobs of their interest which align their career growth.