



INDIAN INSTITUTE OF  
INFORMATION  
TECHNOLOGY



## **PROJECT REPORT ON “CAMPUS RECRUITMENT SYSTEM”**

**SUBMITTED BY:**

**TEAM 11**

**SUBMITTED TO :**

**DR UMA SHESHADRI MAM**

**DR MANJUNATH K VANAHALI SIR**

## **TEAM 11 :**

- 1. SOURAV BHAGAT (19BCS103)**
- 2. SHRESTH BHARADIA (19BCS100)**
- 3. AFFEF HASAN (19BCS106)**
- 4. YASHARTH DUBEY (19BCS114)**
- 5. KRISHNA SUTRAVE (19BCS105)**

## **ABSTRACT**

Campus Recruitment System aims at providing the compatibility to simplify the process of placement for students. This system that consists of a student login, company login and an admin login. This is beneficial for college students, various companies visiting the campus for recruitment and even the college placement officer. The software system allows the students to create their profiles and upload all their details including their marks on to the system. The admin can check each student details and can remove faculty accounts. The system also consists of a company login where various companies visiting the college can view a list of students in that college and also their respective resumes. The software system allows students to view a list of companies who have posted for vacancy. The admin has overall rights over the system and can moderate and delete any details not pertaining to college placement rules

Generally, now a day's every college is conducting a placement drives to provide maximum employment for the students so conducting placement drives is not only necessary we need to make the reach of that drives to students. So this Campus Recruitment System application provides the solution. In this application the admin will the add the Coordinators, Co-coordinators are the persons who bring the placements to the college so when the coordinator gets logged into the application he can add the next coming drive details by giving Company Name, Departments to attend and the informatory videos and images. After adding the coordinator admin can upload the materials for students. Students can know which type of question being asked and information about that placement. In this application, the drives which are going outside the college are not known to coordinators. So the students have an opportunity to add other placements. After adding other placements admin will see the other placements if the company is good he will not delete the details if the details are fake or company is not nice admin will delete the drive details.

# **CHAPTER 1 : INTRODUCTION**

## **1.1 INTRODUCTION**

A college Campus Recruitment System that consists of a student login, company login and an admin login. The project is beneficial for college students, various companies visiting the campus for recruitment and even the college placement officer. The software system allows the students to create their profiles and upload all their details including their marks onto the system. The admin can check each student details and can remove faulty accounts. The system also consists of a company login where various companies visiting the college can view a list of students in that college and also their respective resumes. The software system allows students to view a list of companies who have posted for vacancy. The admin has overall rights over the system and can moderate and delete any details not pertaining to college placement rules. The system handles student as well as company data and efficiently displays all this data to respective sides.

## **1.2 AIM**

The major aim of campus placement is to identify the talented and qualified professionals before they complete their education. It provide employment opportunities to students who are pursuing or in the final stage of completing the course. This process reduces the time for an industry to pick the candidates according to their need. It is a cumbersome activity and hence majority of the companies find it difficult to trace the right talent. Many students do not understand the importance of placement training that is being imparted, whether it is an aptitude training or soft skills. They show the least interest in this due to various factors viz., projects, assignments or more of activities loaded by the colleges as part of their curriculum thinking that it is not useful. It is the responsibility of the companies training on placement to make the students equipped on all aspects of career development along with creating a very good impact in them which makes them feel every minute they spend in the placement training session is worth being there and will help them in getting placed in their dream companies.

## **1.3 EXISTING SYSTEM**

All processes in existing system are handled manually. All the work that is done in the existing system is done by the human intervention .As all the work is done manually, there were a lot of workload on placement officer and it also increases the maximum chances of errors. This is so slow and time consuming. Due to increase in number of user's the process become more difficult. In the system. This

big problem is the searching; sorting and updating of the student data and no any notification method available for giving information to student except the notice board.

## **1.4 PROPOSED SYSTEM**

The proposed Online Placement system is intended to avoid all the drawbacks of existing system. It will add some more features than the existing system. The proposed Online Placement system is a cost effective way of doing the manual processes done in the existing system. This helps the organization to win the war in the existing competitive world.

## **1.5 FEASIBILITY STUDY**

A feasibility study is a high-level capsule version of the entire System analysis and Design Process. The study begins by classifying the problem definition. Feasibility is to determine if it's worth doing. Once an acceptance problem definition has been generated, the analyst develops a logical model of the system. A search for alternatives is analyzed carefully. There are 3 parts in feasibility study.

1) Operational Feasibility

2) Technical Feasibility

3) Economical Feasibility

### **1.5.1 OPERATIONAL FEASIBILITY**

Operational feasibility is the measure of how well a proposed system solves the problems, and takes advantage of the opportunities identified during scope definition and how it satisfies the requirements identified in the requirements analysis phase of system development. The operational feasibility assessment focuses on the degree to which the proposed development projects fits in with the existing business environment and objectives with regard to development schedule, delivery date, corporate culture and existing business processes. To ensure success, desired operational outcomes must be imparted during design and development. These include such design-dependent parameters as reliability, maintainability, supportability, usability, producibility, disposability, sustainability, affordability and others. These parameters are required to be considered at the early stages of design if desired operational behaviours are to be realised. A system design and

development requires appropriate and timely application of engineering and management efforts to meet the previously mentioned parameters. A system may serve its intended purpose most effectively when its technical and operating characteristics are engineered into the design. Therefore, operational feasibility is a critical aspect of systems engineering that needs to be an integral part of the early design phases.

### **1.5.2 TECHNICAL FEASIBILITY**

This involves questions such as whether the technology needed for the system exists, how difficult it will be to build, and whether the firm has enough experience using that technology. The assessment is based on outline design of system requirements in terms of input, processes, output, fields, programs and procedures. This can be qualified in terms of volume of data, trends, frequency of updating in order to give an introduction to the technical system. The application is the fact that it has been developed on windows XP platform and a high configuration of 1GB RAM on Intel Pentium Dual core processor. This is technically feasible. The technical feasibility assessment is focused on gaining an understanding of the present technical resources of the organization and their applicability to the expected needs of the proposed system. It is an evaluation of the hardware and software and how it meets the need of the proposed system.

### **1.5.3 ECONOMICAL FEASIBILITY**

Establishing the cost-effectiveness of the proposed system i.e. if the benefits do not outweigh the costs then it is not worth going ahead. In the fast paced world today there is a great need of online social networking facilities. Thus the benefits of this project in the current scenario make it economically feasible. The purpose of the economic feasibility assessment is to determine the positive economic benefits to the organization that the proposed system will provide. It includes quantification and identification of all the benefits expected. This assessment typically involves a cost/benefits analysis.

## **1.6 ORGANISATION OF THE REPORT**

### **1.6.1 INTRODUCTION**

This section includes the overall view of the project i.e. the basic problem definition and the general overview of the problem which describes the problem in layman terms. It also specifies the software used and the proposed solution strategy.

## 1.6.2 SOFTWARE REQUIREMENTS SPECIFICATION

This section includes the Software and hardware requirements for the smooth running of the application.

## 1.6.3 DESIGN & PLANNING

This section consists of the Software Development Life Cycle model. It also contains technical diagrams like the Data Flow Diagram and the Entity Relationship diagram.

## 1.6.4 IMPLEMENTATION DETAILS

This section describes the different technologies used for the entire development process of the Front-end as well as the Back-end development of the application.

## 1.6.5 RESULTS AND DISCUSSION

This section has screenshots of all the implementation i.e. user interface and their description.

## 1.6.6 SUMMARY AND CONCLUSION

This section has screenshots of all the implementation i.e. user interface and their description.

# CHAPTER 2 : SOFTWARE REQUIREMENTS SPECIFICATION

## 2.1 Hardware Requirements

Number	Description
1	PC with 250 GB or more Hard disk.
2	PC with 8 GB RAM.
3	PC with Intel i5 and Above.

## 2.2 Software Requirements

Number	Description	Type
1	Operating System	Windows XP / Windows
2	Language	reactjs,nodejs
3	Database	mongodb
4	IDE	Visual Code
5	Browser	Google Chrome

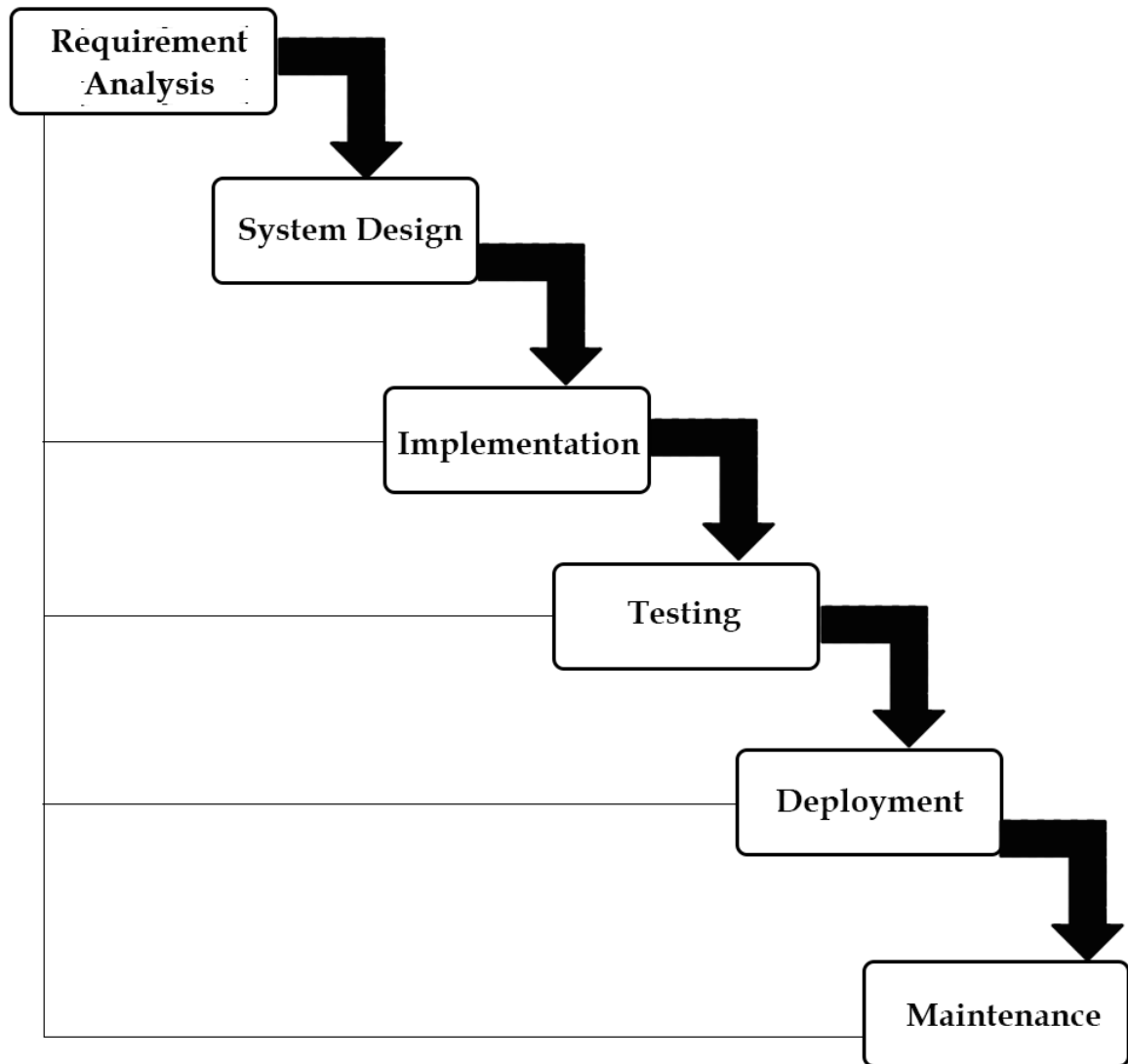
## 3.1 Software Development Life Cycle Model

### 3.1.1 WATERFALL MODEL

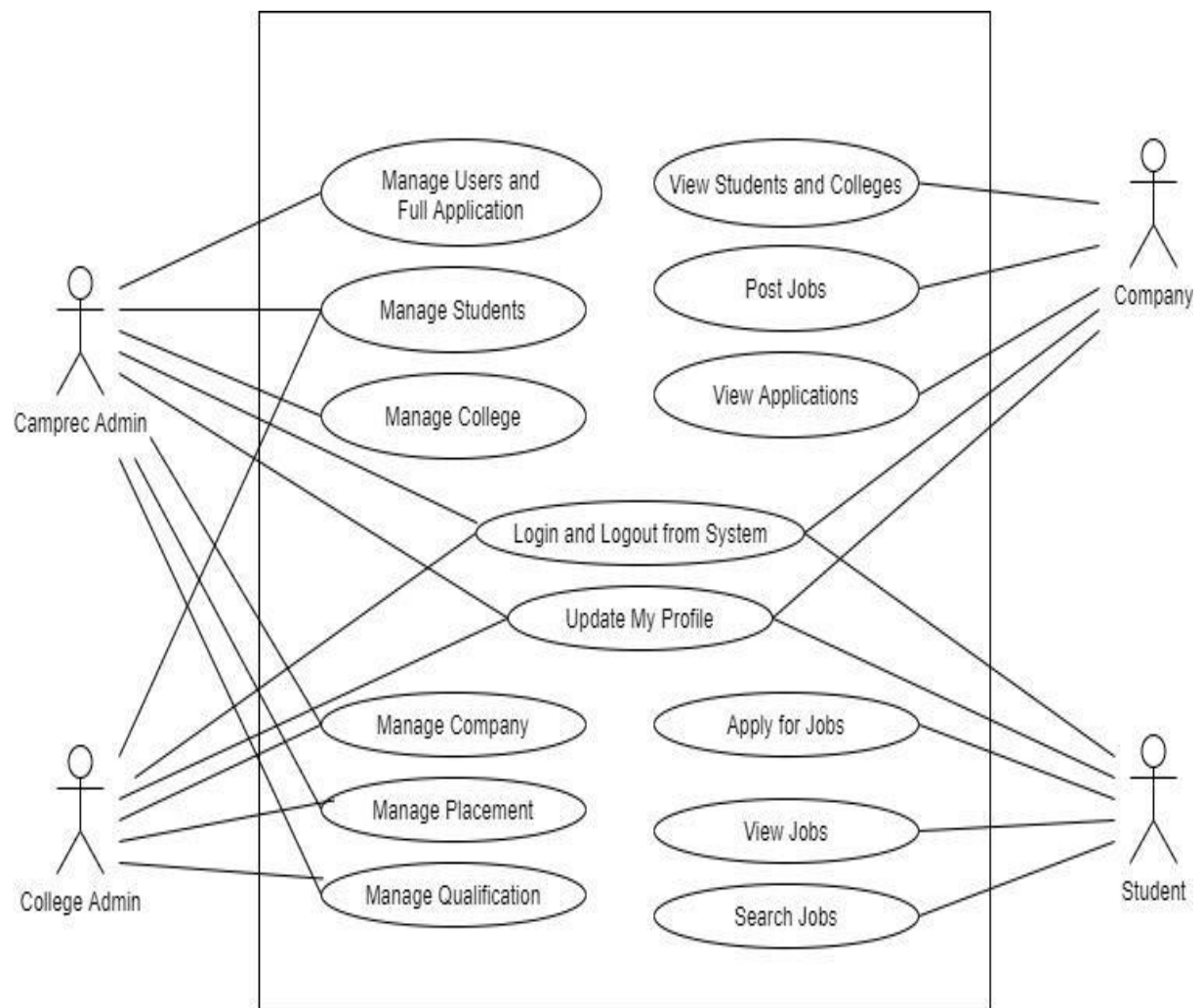
The waterfall model was selected as the SDLC model due to the following reasons:

- Requirements were very well documented, clear and fixed.
- Technology was adequately understood.
- Simple and easy to understand and use.
- There were no ambiguous requirements.
- Easy to manage due to the rigidity of the model. Each phase has specific deliverables and a review process.
- Clearly defined stages.
- Well understood milestones.Easy to arrange tasks.



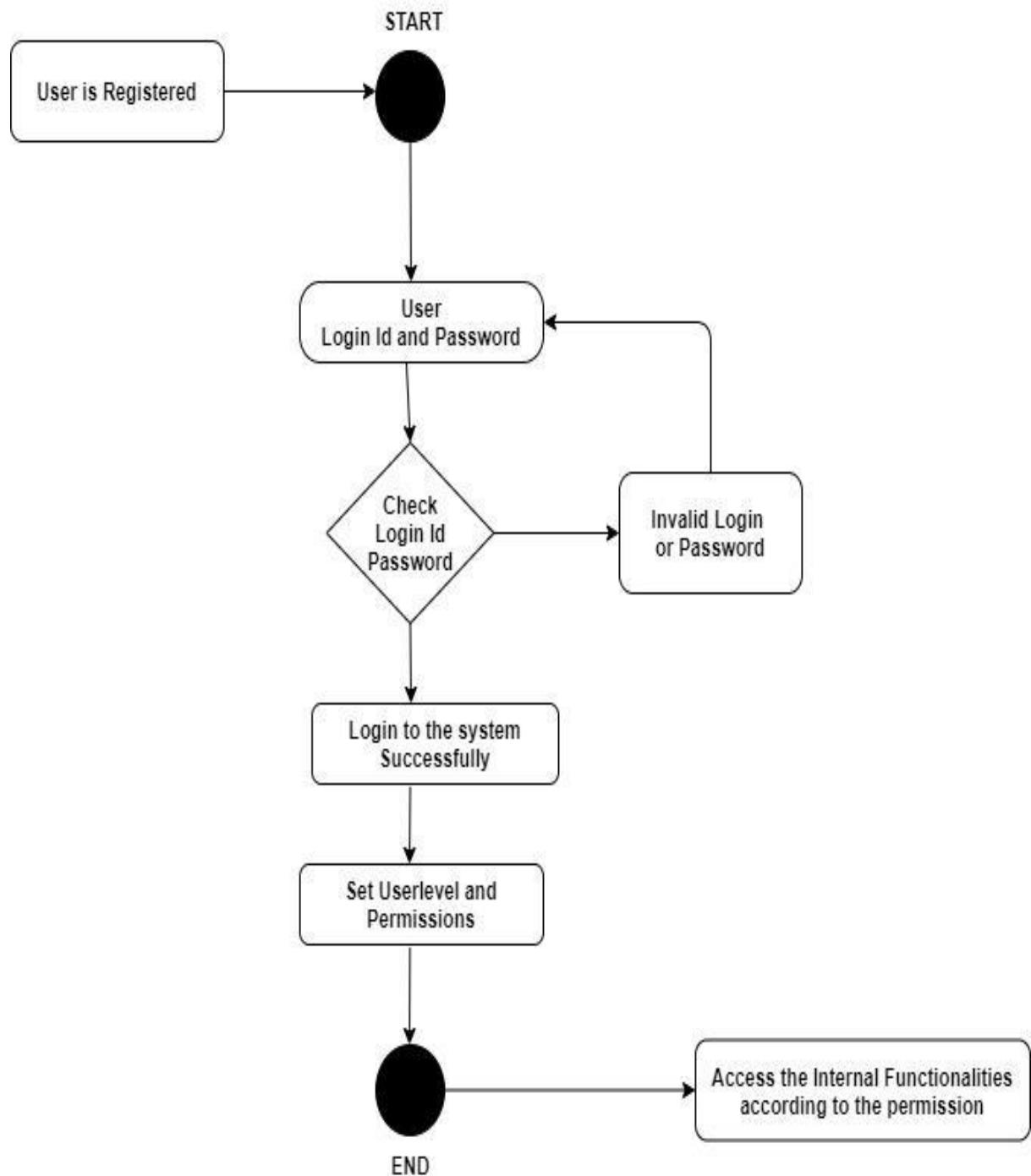


### 3.2 Use Case Diagram of Camprec



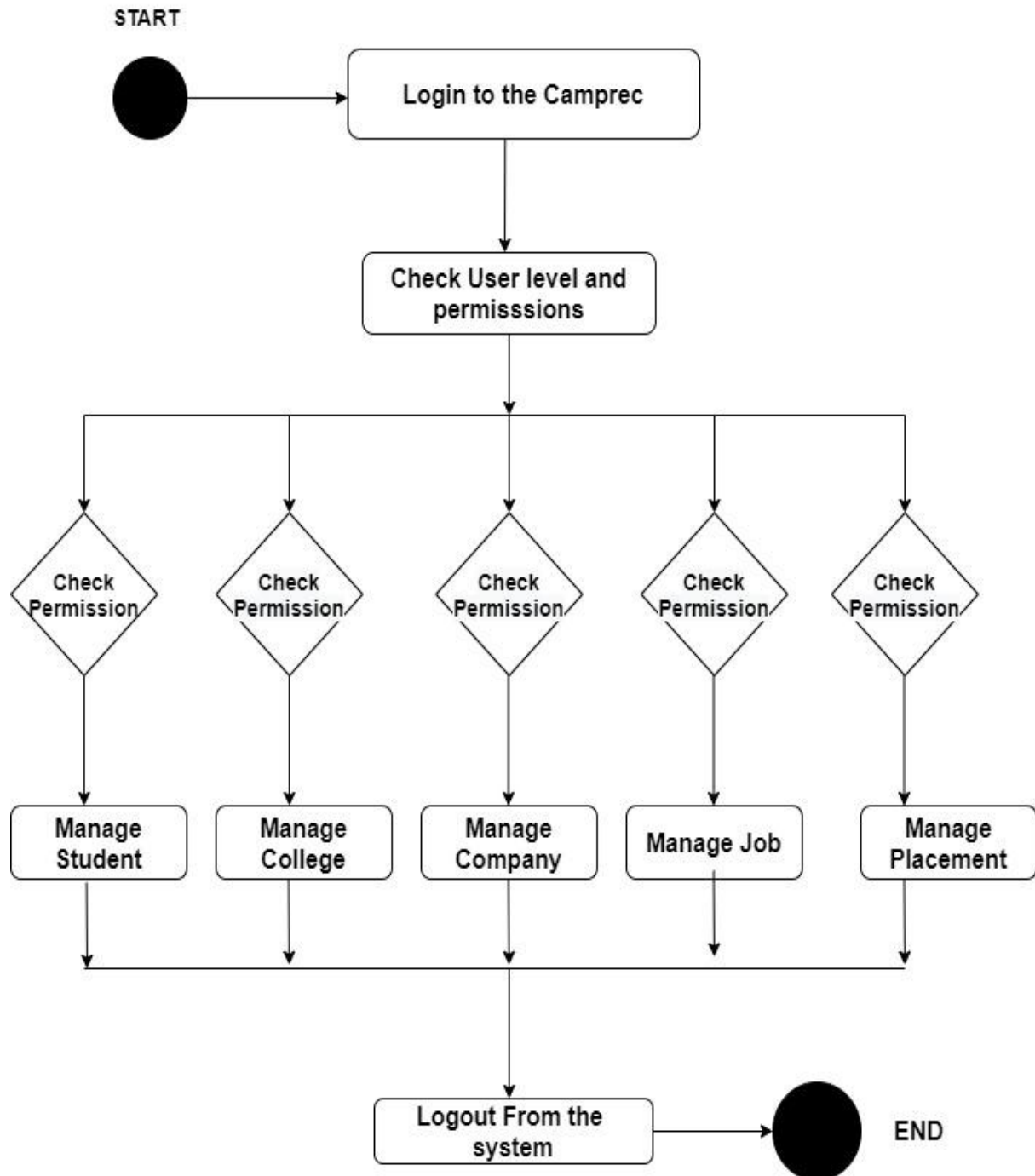
**Use Case Diagram For Camprec**

### 3.2 Login Activity Diagram of Camprec



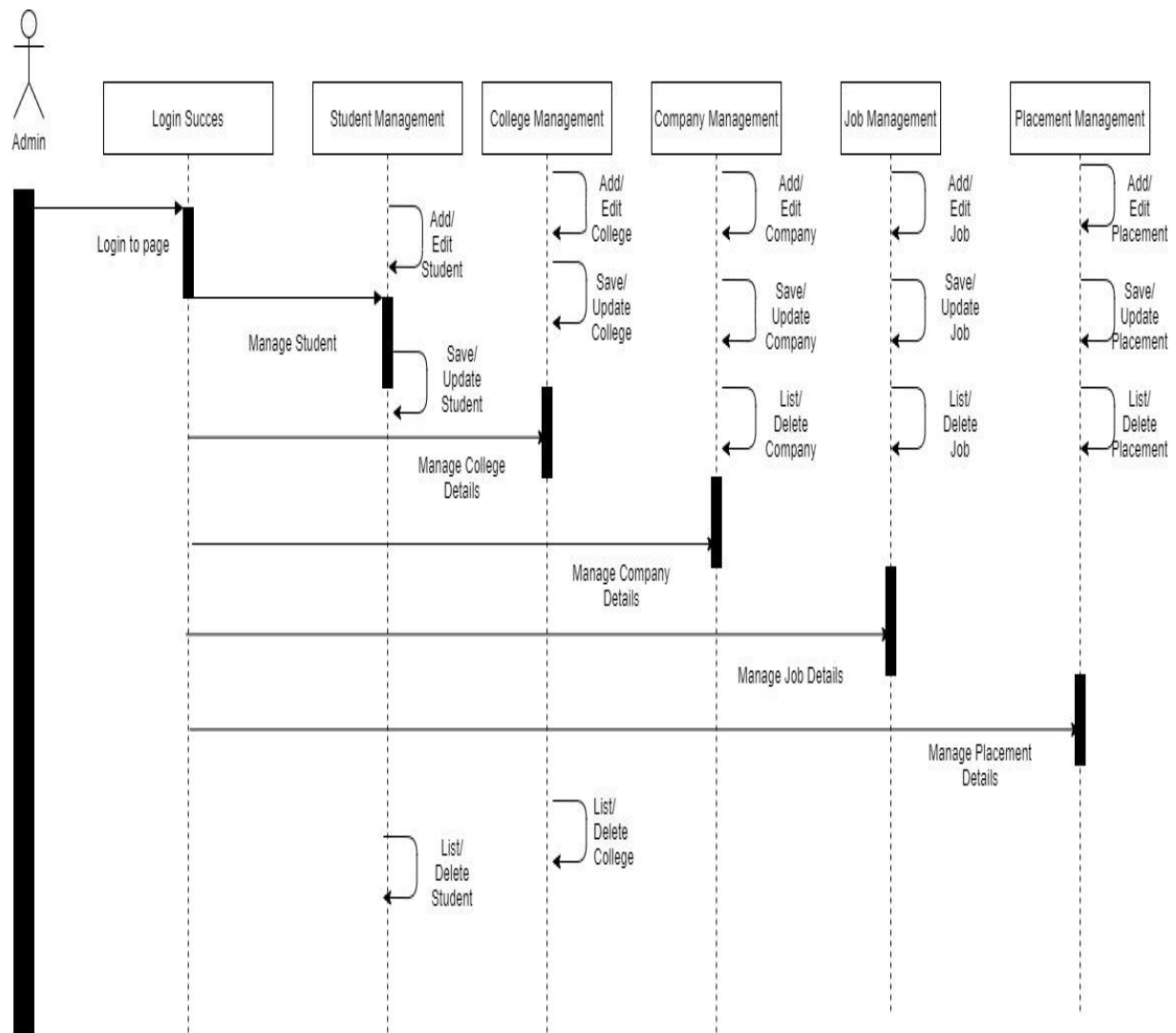
Login Activity Diagram of Camprec

### 3.3 Activity Diagram of Camprec



Activity diagram for Camprec

### 3.4 Sequence Diagram of Camprec



Sequence Diagram of Camprec

## CHAPTER 4 : IMPLEMENTATION DETAILS

In this Section we will do Analysis of Technologies to use for implementing the project.

### 4.1 : FRONT END

#### 4.1.1 : React.JS

## What is React?

React is a JavaScript library created by Facebook.

React is a tool for building UI components.

---

## How does React Work?

React creates a VIRTUAL DOM in memory.

Instead of manipulating the browser's DOM directly, React creates a virtual DOM in memory, where it does all the necessary manipulating, before making the changes in the browser DOM.

React only changes what needs to be changed!

React finds out what changes have been made, and changes **only** what needs to be changed.

You will learn the various aspects of how React does this in the rest of this tutorial.

---

## React.JS History

Current version of React.JS is V16.8.6 (March 2019).

Initial Release to the Public (V0.3.0) was in July 2013.

React.JS was first used in 2011 for Facebook's Newsfeed feature.

Facebook Software Engineer, Jordan Walke, created it.

The create-react-app version 2.0 package was released in October 2018.

Create-react-app version 2.0 supports Babel 7, webpack 4, and Jest23.

#### **4.1.2 : Css**

Cascading Style Sheets (CSS) is a style sheet language used for describing the presentation of a document written in a markup language like HTML. CSS is a cornerstone technology of the World Wide Web, alongside HTML and JavaScript. CSS is designed to enable the separation of presentation and content, including layout, colors, and fonts. This separation can improve content accessibility, provide more flexibility and control in the specification of presentation characteristics, enable multiple web pages to share formatting by specifying the relevant CSS in a separate .css file, and reduce complexity and repetition in the structural content.

CSS information can be provided from various sources. These sources can be the web browser, the user and the author. The information from the author can be further classified into inline, media type, importance, selector specificity, rule order, inheritance and property definition. CSS style information can be in a separate document or it can be embedded into an HTML document. Multiple style sheets can be imported. Different styles can be applied depending on the output device being used; for example, the screen version can be quite different from the printed version, so that authors can tailor the presentation appropriately for each medium. The style sheet with the highest priority controls the content display. Declarations not set in the highest priority source are passed on to a source of lower priority, such as the user agent style. The process is called cascading.

One of the goals of CSS is to allow users greater control over presentation. Someone who finds red italic headings difficult to read may apply a different style sheet. Depending on the browser and the web site, a user may choose from various style sheets provided by the designers, or may remove all added styles and view the site using the browser's default styling, or may override just the red italic heading style without altering other attributes.

### 4.1.3 : Axios

## Axios

*Axios* is a promise based HTTP client for the browser and Node.js. Axios makes it easy to send asynchronous HTTP requests to REST endpoints and perform CRUD operations. It can be used in plain JavaScript or with a library such as Vue or React.

### Axios is a library that helps us make http requests to external resources

In our React applications we often need to retrieve data from external APIs so it can be displayed in our web pages.

One way to build this feature is to use the Javascript *Fetch* API. Fetch is quite capable of retrieving external data, but it has some limitations.

A more popular way of performing this operation is to use the *Axios* library. Axios is designed to handle **http** requests and responses. It's used more often than Fetch because it has a larger set of features and it supports older browsers.

Axios deals with responses using *Promises*, so it's streamlined and easy to use in our code. Axios uses methods like **get()** and **post()** that perform **http GET** and **POST** requests for retrieving or creating resources.

## 4.2 : BACK END

### 4.2.1.ExpressJS

ExpressJS is a web application framework that provides you with a simple API to build websites, web apps and back ends. With ExpressJS, you need not worry about low level protocols, processes, etc.



## What is Express?

Express provides a minimal interface to build our applications. It provides us the tools that are required to build our app. It is flexible as there are numerous modules available on **npm**, which can be directly plugged into Express.

Express was developed by **TJ Holowaychuk** and is maintained by the Node.js foundation and numerous open source contributors.

## Why Express?

Unlike its competitors like Rails and Django, which have an opinionated way of building applications, Express has no "best way" to do something. It is very flexible and pluggable.

### Pug

Pug (earlier known as Jade) is a terse language for writing HTML templates. It –

- Produces HTML
- Supports dynamic code
- Supports reusability (DRY)

It is one of the most popular template language used with Express.

## MongoDB and Mongoose

MongoDB is an open-source, document database designed for ease of development and scaling. This database is also used to store data.

Mongoose is a client API for **node.js** which makes it easy to access our database from our Express application.

### 4.2.2 : NodeJS:

## What is Node.js?

- Node.js is an open source server environment
  - Node.js is free
  - Node.js runs on various platforms (Windows, Linux, Unix, Mac OS X, etc.)
  - Node.js uses JavaScript on the server
-

# Why Node.js?

## **Node.js uses asynchronous programming!**

A common task for a web server can be to open a file on the server and return the content to the client.

Here is how PHP or ASP handles a file request:

1. Sends the task to the computer's file system.
2. Waits while the file system opens and reads the file.
3. Returns the content to the client.
4. Ready to handle the next request.

Here is how Node.js handles a file request:

1. Sends the task to the computer's file system.
2. Ready to handle the next request.
3. When the file system has opened and read the file, the server returns the content to the client.

Node.js eliminates the waiting, and simply continues with the next request.

Node.js runs single-threaded, non-blocking, asynchronously programming, which is very memory efficient.

---

## What Can Node.js Do?

- Node.js can generate dynamic page content
- Node.js can create, open, read, write, delete, and close files on the server
- Node.js can collect form data
- Node.js can add, delete, modify data in your database

---

## What is a Node.js File?

- Node.js files contain tasks that will be executed on certain events
- A typical event is someone trying to access a port on the server
- Node.js files must be initiated on the server before having any effect
- Node.js files have extension ".js"

### **4.2.3 : Email Js**

#### **What is EmailJS?**

EmailJS is a service that allows us to send emails directly from your client-side JavaScript code. As you can see, if we use EmailJs we don't need a server to do that.

#### **How does EmailJS work?**

Just connect EmailJS to one of the supported email services, create an email template, and use Javascript to trigger an email.

### **4.3 : Database**

#### **4.3.1: MongoDB:**

##### **What Is MongoDB?**

MongoDB is an open-source document database built on a horizontal scale-out architecture. Founded in 2007, MongoDB has a worldwide following in the developer community.

Instead of storing data in tables of rows or columns like SQL databases, each row in a MongoDB database is a document described in JSON, a formatting language.

Document databases are extremely flexible, allowing variations in the structure of documents and allowing storage of documents that are partially complete. One document can have others embedded in it.

Fields in a document play the role of columns in a SQL database, and like columns, they can be indexed to increase search performance.

Best of all for many developers, the programmer can change the structure of the database easily as needs change. Some say this turns data into code.

From its founding, MongoDB was built on a scale-out architecture, a structure that allows many small machines to work together to create systems that are fast and handle huge amounts of data.

MongoDB has always focused on providing developers an excellent user experience, which, in addition to all its other properties, has made MongoDB a favorite of developers worldwide for a huge variety of applications.

## **What is NoSQL?**

When people use the term “NoSQL database”, they typically use it to refer to any non-relational database. Some say the term “NoSQL” stands for “non SQL” while others say it stands for “not only SQL.” Either way, most agree that NoSQL databases are databases that store data in a format other than relational tables.

A common misconception is that NoSQL databases or non-relational databases don’t store relationship data well. NoSQL databases can store relationship data — they just store it differently than relational databases do. In fact, when compared with SQL databases, many find modeling relationship data in NoSQL databases to be *easier* than in SQL databases, because related data doesn’t have to be split between tables.

NoSQL data models allow related data to be nested within a single data structure.

NoSQL databases emerged in the late 2000s as the cost of storage dramatically decreased. Gone were the days of needing to create a complex, difficult-to-manage data model simply for the purposes of reducing data duplication. Developers (rather than storage) were becoming the primary cost of software development, so NoSQL databases optimized for developer productivity.

## **4.4 : Deployment:**

### **4.4.1 : Heroku:**

## **What is Heroku?**

Heroku is a container-based cloud Platform as a Service (PaaS). Developers use Heroku to deploy, manage, and scale modern apps. Our platform is elegant, flexible, and easy to use, offering developers the simplest path to getting their apps to market.

Heroku is fully managed, giving developers the freedom to focus on their core product without the distraction of maintaining servers, hardware, or infrastructure. The Heroku experience provides services, tools, workflows, and polyglot support—all designed to enhance developer productivity.

## **CHAPTER 5 : TESTING AND IMPLEMENTATION**

The term implementation has different meanings ranging from the conversion of a basic application to a complete replacement of a computer system. The procedures however, are virtually the same. Implementation includes all those activities that take place to convert from old system to new. The new system may be totally new replacing an existing manual or automated system or it may be major modification to an existing system. The method of implementation and time scale to be adopted is found out initially. Proper implementation is essential to provide a reliable system to meet organization requirement.

### **5.1 : UNIT TESTING**

#### **5.1.1 Introduction**

In computer programming, unit testing is a software testing method by which individual units of source code, sets of one or more computer program modules together with associated control data, usage procedures, and operating procedures, are tested to determine whether they are fit for use. Intuitively, one can view a unit as the smallest testable part of an application. In procedural programming, a unit could be an entire module, but it is more commonly an individual function or procedure. In object-oriented programming, a unit is often an entire interface, such as a class, but could be an individual method. Unit tests are short code fragments created by programmers or occasionally by white box testers during the development process. It forms the basis for component testing. Ideally, each test case is independent from the others. Substitutes such as method stubs, mock objects, fakes, and test harnesses can be used to assist testing a module in isolation. Unit tests are typically written and run by software developers to ensure that code meets its design and behaves as intended.

### 5.1.2 Benefits

The goal of unit testing is to isolate each part of the program and show that the individual parts are correct. A unit test provides a strict, written contract that the piece of code must satisfy. As a result, it affords several benefits.

**1) Find problems early :** Unit testing finds problems early in the development cycle. In test-driven development (TDD), which is frequently used in both extreme programming and scrum, unit tests are created before the code itself is written. When the tests pass, that code is considered complete. The same unit tests are run against that function frequently as the larger code base is developed either as the code is changed or via an automated process with the build. If the unit tests fail, it is considered to be a bug either in the changed code or the tests themselves. The unit tests then allow the location of the fault or failure to be easily traced. Since the unit tests alert the development team of the problem before handing the code off to testers or clients, it is still early in the development process.

**2 ) Facilitates Change :** Unit testing allows the programmer to refactor code or upgrade system libraries at a later date, and make sure the module still works correctly (e.g., in regression testing). The procedure is to write test cases for all functions and methods so that whenever a change causes a fault, it can be quickly identified. Unit tests detect changes which may break a design contract.

**3 ) Simplifies Integration :** Unit testing may reduce uncertainty in the units themselves and can be used in a bottom-up testing style approach. By testing the parts of a program first and then testing the sum of its parts, integration testing becomes much easier.

**4 ) Documentation :** Unit testing provides a sort of living documentation of the system. Developers looking to learn what functionality is provided by a unit, and how to use it, can look at the unit tests to gain a basic understanding of the unit's interface (API). Unit test cases embody characteristics that are critical to the success of the unit. These characteristics can indicate appropriate/inappropriate use of a unit as well as negative behaviors that are to be trapped by the unit. A unit test case, in and of itself, documents these critical characteristics, although many software development environments do not rely solely upon code to document the product in development.

## **5.2 : INTEGRATION TESTING**

Integration testing (sometimes called integration and testing, abbreviated I&T) is the phase in software testing in which individual software modules are combined and tested as a group. It occurs after unit testing and before validation testing. Integration testing takes as its input modules that have been unit tested, groups them in larger aggregates, applies tests defined in an integration test plan to those aggregates, and delivers as its output the integrated system ready for system testing.

### **5.2.1 Purpose**

The purpose of integration testing is to verify functional, performance, and reliability requirements placed on major design items. These "design items", i.e., assemblages (or groups of units), are exercised through their interfaces using black-box testing, success and error cases being simulated via appropriate parameter and data inputs. Simulated usage of shared data areas and inter-process communication is tested and individual subsystems are exercised through their input interface. Test cases are constructed to test whether all the components within assemblages interact correctly, for example across procedure calls or process activations, and this is done after testing individual modules, i.e., unit testing. The overall idea is a "building block" approach, in which verified assemblages are added to a verified base which is then used to support the integration testing of further assemblages. Software integration testing is performed according to the software development life cycle (SDLC) after module and functional tests. The cross-dependencies for software integration testing are: schedule for integration testing, strategy and selection of the tools used for integration, define the cyclomathical complexity of the software and software architecture, reusability of modules and life-cycle and versioning management. Some different types of integration testing are big-bang, top-down, and bottom-up, mixed (sandwich) and risky-hardest. Other Integration Patterns[2] are: collaboration integration, backbone integration, layer integration, client-server integration, distributed services integration and high-frequency integration.

#### **5.2.1.1 Big Bang**

In the big-bang approach, most of the developed modules are coupled together to form a complete software system or major part of the system and then used for integration testing. This method is very effective for saving time in the integration testing process. However, if the test cases and their results are not recorded properly, the entire integration process will be more complicated and may prevent

the testing team from achieving the goal of integration testing. A type of big-bang integration testing is called "usage model testing" which can be used in both software and hardware integration testing. The basis behind this type of integration testing is to run user-like workloads in integrated user-like environments. In doing the testing in this manner, the environment is proofed, while the individual components are proofed indirectly through their use. Usage Model testing takes an optimistic approach to testing, because it expects to have few problems with the individual components. The strategy relies heavily on the component developers to do the isolated unit testing for their product. The goal of the strategy is to avoid redoing the testing done by the developers, and instead flesh-out problems caused by the interaction of the components in the environment. For integration testing, Usage Model testing can be more efficient and provides better test coverage than traditional focused functional integration testing. To be more efficient and accurate, care must be used in defining the user-like workloads for creating realistic scenarios in exercising the environment. This gives confidence that the integrated environment will work as expected for the target customers.

#### **5.2.1.2 Top-down And Bottom-up**

Bottom-up testing is an approach to integrated testing where the lowest level components are tested first, then used to facilitate the testing of higher level components. The process is repeated until the component at the top of the hierarchy is tested. All the bottom or low-level modules, procedures or functions are integrated and then tested. After the integration testing of lower level integrated modules, the next level of modules will be formed and can be used for integration testing. This approach is helpful only when all or most of the modules of the same development level are ready. This method also helps to determine the levels of software developed and makes it easier to report testing progress in the form of a percentage. Top-down testing is an approach to integrated testing where the top integrated modules are tested and the branch of the module is tested step by step until the end of the related module. Sandwich testing is an approach to combine top down testing with bottom up testing.

### **5.3 : SOFTWARE VERIFICATION AND VALIDATION**

#### **5.3.1 Introduction**

In software project management, software testing, and software engineering, verification and validation (V&V) is the process of checking that a software system meets specifications and that it fulfills its intended purpose. It may also be referred to as software quality control. It is normally the responsibility of software testers as part of the software development lifecycle. Validation checks



that the product design satisfies or fits the intended use (high-level checking), i.e., the software meets the user requirements. This is done through dynamic testing and other forms of review. Verification and validation are not the same thing, although they are often confused. Boehm succinctly expressed the difference between

- Validation : Are we building the right product?
- Verification : Are we building the product right?

According to the Capability Maturity Model (CMMI-SW v1.1)

**Software Verification:** The process of evaluating software to determine whether the products of a given development phase satisfy the conditions imposed at the start of that phase.

**Software Validation:** The process of evaluating software during or at the end of the development process to determine whether it satisfies specified requirements.

In other words, software verification is ensuring that the product has been built according to the requirements and design specifications, while software validation ensures that the product meets the user's needs, and that the specifications were correct in the first place. Software verification ensures that "you built it right". Software validation ensures that "you built the right thing". Software validation confirms that the product, as provided, will fulfill its intended use.

From Testing Perspective

- Fault – wrong or missing function in the code.
- Failure – the manifestation of a fault during execution.
- Malfunction – according to its specification the system does not meet its specified functionality

Both verification and validation are related to the concepts of quality and of software quality assurance. By themselves, verification and validation do not guarantee software quality; planning, traceability, configuration management and other aspects of software engineering are required. Within the modeling and simulation (M&S) community, the definitions of verification, validation and accreditation are similar:

- M&S Verification is the process of determining that a • computer model, simulation, or federation of models and simulations implementations and

their associated data accurately represent the developer's conceptual description and specifications.

- M&S Validation is the process of determining the degree to which a model, simulation, or federation of models and simulations, and their associated data are accurate representations of the real world from the perspective of the intended use(s).

### **5.3.2 Classification of Methods**

- In mission-critical software systems, where flawless performance is absolutely necessary, formal methods may be used to ensure the correct operation of a system. However, often for non-mission-critical software systems, formal methods prove to be very costly and an alternative method of software V&V must be sought out. In such cases, syntactic methods are often used.

### **5.3.3 Test Cases**

- A test case is a tool used in the process. Test cases may be prepared for software verification and software validation to determine if the product was built according to the requirements of the user. Other methods, such as reviews, may be used early in the life cycle to provide for software validation.

## **5.4 : Black-Box Testing**

- Black-box testing is a method of software testing that examines the functionality of an application without peering into its internal structures or workings. This method of test can be applied virtually to every level of software testing: unit, integration, system and acceptance. It typically comprises most if not all higher level testing, but can also dominate unit testing as well.

### **5.4.1 Test Procedures**

- Specific knowledge of the application's code/internal structure and programming knowledge in general is not required. The tester is aware of what the software is supposed to do but is not aware of how it does it. For instance, the tester is aware that a particular input returns a certain,

invariable output but is not aware of how the software produces the output in the first place.

### 5.4.2 Test Cases

- Test cases are built around specifications and requirements, i.e., what the application is supposed to do. Test cases are generally derived from external descriptions of the software, including specifications, requirements and design parameters. Although the tests used are primarily functional in nature, non-functional tests may also be used. The test designer selects both valid and invalid inputs and determines the correct output, often with the help of an oracle or a previous result that is known to be good, without any knowledge of the test object's internal structure.

## 5.5 : White-Box Testing

- White-box testing (also known as clear box testing, glass box testing, transparent box testing, and structural testing) is a method of testing software that tests internal structures or workings of an application, as opposed to its functionality (i.e. black-box testing). In white-box testing an internal perspective of the system, as well as programming skills, are used to design test cases. The tester chooses inputs to exercise paths through the code and determine the appropriate outputs. This is analogous to testing nodes in a circuit, e.g. in-circuit testing (ICT). White-box testing can be applied at the unit, integration and system levels of the software testing process. Although traditional testers tended to think of white-box testing as being done at the unit level, it is used for integration and system testing more frequently today. It can test paths within a unit, paths between units during integration, and between subsystems during a system-level test. Though this method of test design can uncover many errors or problems, it has the potential to miss unimplemented parts of the specification or missing requirements.

### 5.5.1 Levels

- **1 ) Unit testing :** White-box testing is done during unit testing to ensure that the code is working as intended, before any integration happens with previously tested code. White-box testing during unit testing catches any defects early on and aids in any defects that happen later on after the code is integrated with the rest of the application and therefore prevents any type of errors later on.

- **2 ) Integration testing :** White-box testing at this level are written to test the interactions of each interface with each other. The Unit level testing made sure that each code was tested and working accordingly in an isolated environment and integration examines the correctness of the behaviour in an open environment through the use of white-box testing for any interactions of interfaces that are known to the programmer.

**3 ) Regression testing :** White-box testing during regression testing is the use of recycled white-box test cases at the unit and integration testing levels.

### **5.5.2 Procedures**

White-box testing's basic procedures involves the tester having a deep level of understanding of the source code being tested. The programmer must have a deep understanding of the application to know what kinds of test cases to create so that every visible path is exercised for testing. Once the source code is understood then the source code can be analyzed for test cases to be created. These are the three basic steps that white-box testing takes in order to create test cases:

- Input involves different types of requirements, functional specifications, detailed designing of documents, proper source code, security specifications. This is the preparation stage of white-box testing to layout all of the basic information.
- Processing involves performing risk analysis to guide whole testing process, proper test plan, execute test cases and communicate results. This is the phase of building test cases to make sure they thoroughly test the application the given results are recorded accordingly.
- Output involves preparing final report that encompasses all of the above preparations and results.

### **5.5.3 Advantages**

White-box testing is one of the two biggest testing methodologies used today. It has several major advantages:

- Side effects of having the knowledge of the source code is beneficial to thorough testing.
- Optimization of code by revealing hidden errors and being able to remove these possible defects.

- Gives the programmer introspection because developers carefully describe any new implementation.
- Provides traceability of tests from the source, allowing future changes to the software to be easily captured in changes to the tests.
- White box testing give clear, engineering-based, rules for when to stop testing.

### **5.5.5 Disadvantages**

Although white-box testing has great advantages, it is not perfect and contains some disadvantages:

- White-box testing brings complexity to testing because the tester must have knowledge of the program, including being a programmer. White-box testing requires a programmer with a high level of knowledge due to the complexity of the level of testing that needs to be done.
- On some occasions, it is not realistic to be able to test every single existing condition of the application and some conditions will be untested.
- The tests focus on the software as it exists, and missing functionality may not be discovered.

## **5.6 : SYSTEM TESTING**

System testing of software or hardware is testing conducted on a complete, integrated system to evaluate the system's compliance with its specified requirements. System testing falls within the scope of black-box testing, and as such, should require no knowledge of the inner design of the code or logic. As a rule, system testing takes, as its input, all of the "integrated" software components that have passed integration testing and also the software system itself integrated with any applicable hardware system(s). The purpose of integration testing is to detect any inconsistencies between the software units that are integrated together (called assemblages) or between any of the assemblages and the hardware. System testing is a more limited type of testing; it seeks to detect defects both within the "inter-assemblages" and also within the system as a whole.

System testing is performed on the entire system in the context of a Functional Requirement Specification(s) (FRS) and/or a System Requirement Specification (SRS). System testing tests not only the design, but also the behavior and even the believed expectations of the customer. It is also intended to test up to and beyond the bounds defined in the software/hardware requirements specification(s).

## **CHAPTER 6 : ADVANTAGES**

- It tracks all the information of college students, various companies visiting the campus for recruitment and even the college placement officer.
- Shows the information and description of the companies.
- Integration of all records of Company.
- To increase efficiency of managing Student qualification.

## **CHAPTER 7 : CONCLUSION**

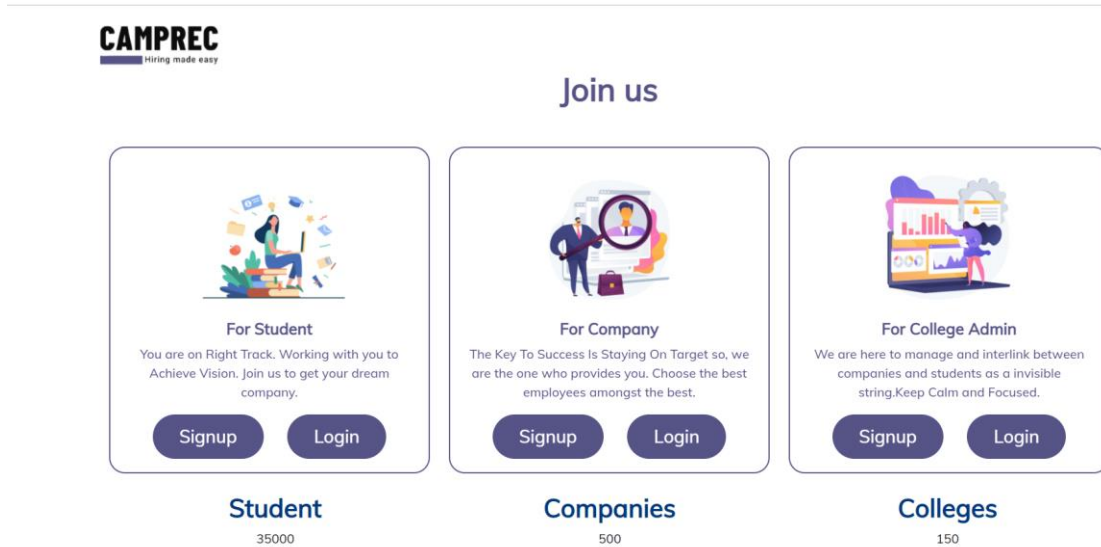
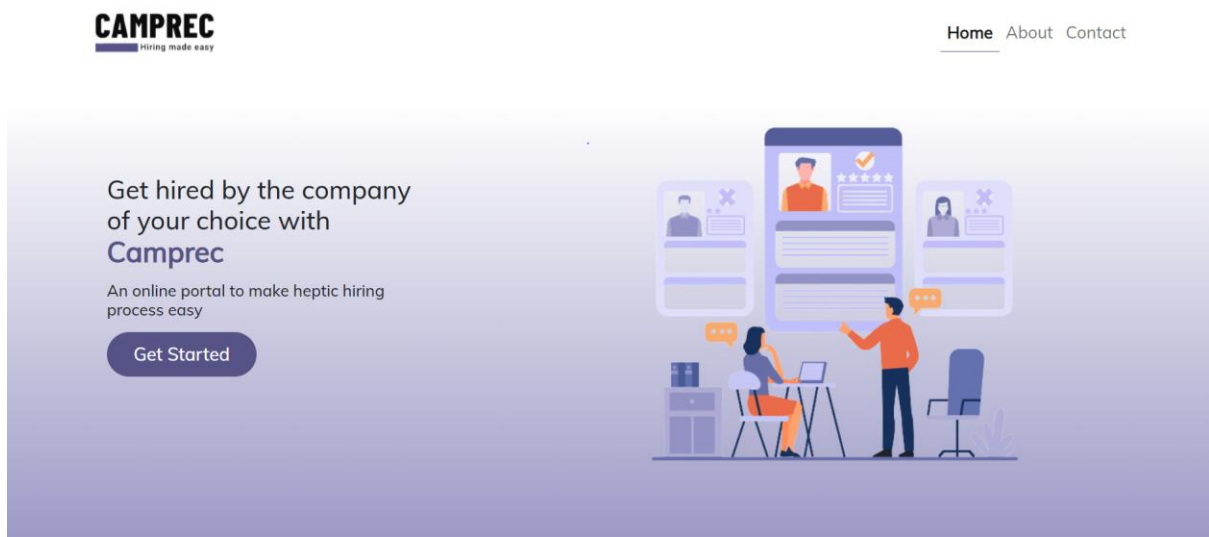
Increasing need of comfort and inculcating all the data at one place has always been a challenging process for everybody. With the introduction of this web based training and placement portal we promise to make the lives of students and administration a little easier by proposing an alternative for the current system being used. Easy accessibility and functioning of this portal will allow easy management of the allocation process during placement period. With the increasing demand of digitalization in every aspect of day to day activities we can anticipate the great demand for such portals in the near future and the comfort it will bring with it to the lives of all. Also the rapidly increasing concerns of global warming due to increase deforestation for large amount of paper that it requires we here have a minor role to save Mother Nature. So we hope all of you can sit back and relax and enjoy the luxury of Digitalization. More so in this busy and exhausting life we are saving one of the most crucial factor that keeps us running that is human energy

## **CHAPTER 8 : BIBLIOGRAPHY**

1. <https://stackoverflow.com/>
2. <https://www.geeksforgeeks.org/>
3. <https://getbootstrap.com/docs/4.1/getting-started/introduction/>
4. <https://reactjs.org/docs/getting-started.html>
5. <https://expressjs.com/>
6. <https://nodejs.org/en/docs/>

# Screenshots of complete Camprec website:

## WELCOME PAGE:



# STUDENT HOME:



sita

Email :  
sita@gmail.com

Phone no:  
2050305030

#### Description :

I am sita. I did my graduation from IIIT Hyderabad. I have job experience with google, bsnl & microsoft.

#### Education :

Course: Btech

Institute: Indian Institute of Information Technology, Dharwad

Marks: 9.8



#### Work experience :

Job Title: Intern

Company: IIT Bombay

Duration: 3 months

Description: I made a website

Link: <https://meet.google.com/yvg-kahf-xzz?pli=1&authuser=4>



#### Certification :

Course: Kunfu

Institutes: Sholin kungfu Institute of Xiangji

Valid\_till: 2022-12-01

Link: <https://meet.google.com/yvg-kahf-xzz?pli=1&authuser=4>





## COMPANY HOME:



[Home](#) [Jobs](#) [Colleges](#) [Logout](#)



### Facebook

Location : Menlo Park,California,United States.

Email: facebook@gmail.com

Year of Establishment: 2004

**Description :** Facebook (stylized as facebook) is an American online social media and social networking service based in Menlo Park, California, and a flagship service of the namesake company Facebook, Inc. It was founded by Mark Zuckerberg, along with fellow Harvard College students and roommates Eduardo Saverin, Andrew McCollum, Dustin Moskovitz, and Chris Hughes.

Website : [www.facebook.ac.in](http://www.facebook.ac.in)



Edit

# COLLEGE ADMIN HOME:



[Home](#) [Student](#) [Companies](#) [Logout](#)



IIIT dwd

Location : Dharwad,Karnataka-580029

ज्ञानेन विकासः

Email: [iiitdwd@gmail.com](mailto:iiitdwd@gmail.com)

Year of Establishment: 2015

Description : best college

Webite : [iiitdwd.ac.in](http://iiitdwd.ac.in)



Webite : [iiitdwd.ac.in](http://iiitdwd.ac.in)



Edit

## **Source Code:**

**Please visit the following Github link for source code:**

**<https://github.com/camprec>**

**Please visit the hosted site:**

**<https://camprec.herokuapp.com/>**