CAREER INFORMATION AND RECRUITMENT PORTAL

PROJECT REPORT

submitted by

**JINCY P JANARDHANAN** (IEAREIT017)

**ALEENA SUNNY** (IEAREIT006)

**ALKA BHAGAVALDAS K** (IEAREIT007)

**AMEENA SHIRIN** (IEAREIT009)

to

the University of Calicut

in partial fulfilment of the requirements for the award of the Degree

of

**Bachelor of Technology**

in

Information Technology



Department of Information Technology

Institute of Engineering and Technology, University of Calicut, Thenjipalam

Kerala

July 18, 2020

DECLARATION

We undersigned hereby declare that the project report **Career Information and Recruitment Portal**, submitted for partial fulfilment of the requirements for the award of degree of Bachelor of Technology of the University of Calicut, Kerala is a bonafide work done by us under supervision of **Ms. Sruthimol M P**. This submission represents our ideas in our own words and where ideas or words of others have been included, we have adequately and accurately cited and referenced the original sources. We also declare that we have adhered to ethics of academic honesty and integrity and have not misrepresented or fabricated any data or idea or fact or source in our submission. We understand that any violation of the above will be a cause for disciplinary action by the institute and/or the University and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been obtained. This report has not been previously formed the basis for the award of any degree, diploma or similar title of any other University.

**Place:** Thenjipalam

**Date:** July 18, 2020

Jincy P Janardhanan

Aleena Sunny

Alka Bhagavaldas K

Ameena Shirin

DEPARTMENT OF INFORMATION TECHNOLOGY  
INSTITUTE OF ENGINEERING AND TECHNOLOGY UNIVERSITY OF CALICUT, THENJIPALAM



CERTIFICATE

This is to certify that the report entitled **"CAREER INFORMATION AND RECRUITMENT PORTAL",** submitted by **Jincy P Janardhanan, Aleena Sunny, Alka Bhagavaldas K, Ameena Shirin** to the **UNIVERSITY OF CALICUT** in partial fulfilment of the requirements for the award of the degree of Bachelor of Technology in Information Technology is a bonafide record of the project presented by them under our guidance and supervision. This report in any form has not been submitted to any other University or Institute for any purpose.

|  |  |
| --- | --- |
| **Ms. Anu Manohar**  Assistant Professor  Department of IT  (Coordinator) | **Ms. Sruthimol M P**  Lecturer  Department of IT  (Coordinator & Guide) |

ACKNOWLEDGEMENT

We would like to express our warmest gratitude to all the people who had been a great help, support and motivation throughout this project. Prima facie, we would like to thank the Almighty for giving us the wisdom and grace for the successful completion of our project. Forever we owe our deepest gratitude to our guide and coordinator, **Ms. Sruthimol M P**, Lecturer, Department of Information Technology, for her expert guidance, co-operation and immense encouragement in pursuing this project. With a profound sense of gratitude, we would like to express our heartfelt thanks to our coordinator, **Ms. Anu Manohar**,Assistant Professor, Department of Information Technology for her earnest co-operation, support and constant inspiration. We extend our sincere gratitude to all the teachers of the Department of IT and to all our friends for their help and support.

Jincy P Janardhanan

Aleena Sunny

Alka Bhagavaldas K

Ameena Shirin

ABSTRACT

Contents

[List of Figures vii](#_Toc46356966)

[List of Tables viii](#_Toc46356967)

[Abbreviations ix](#_Toc46356968)

[Chapter 1 Introduction 1](#_Toc46356969)

[1.1 Problem Statement 1](#_Toc46356970)

[1.2 Motivation and Objective 2](#_Toc46356971)

[Chapter 2 Literature Review 4](#_Toc46356972)

[2.1 Existing Methodologies 4](#_Toc46356973)

[Chapter 3 Proposed System and Feasibility Study 7](#_Toc46356974)

[3.1 Proposed Solution 7](#_Toc46356975)

[3.2 Technical Feasibility 7](#_Toc46356976)

[3.3 Operational Feasibility 7](#_Toc46356977)

[3.4 Economic Feasibility 7](#_Toc46356978)

[3.5 Schedule Feasibility 8](#_Toc46356979)

[Chapter 4 Requirements Gathering and Analysis 10](#_Toc46356980)

[4.1 End User Specification 10](#_Toc46356981)

[4.2 Software Specification 10](#_Toc46356982)

[4.3 Hardware Specification 11](#_Toc46356983)

[4.4 SRS Document 11](#_Toc46356984)

[4.4.1 Functional Requirements 12](#_Toc46356985)

[4.4.2 Non-Functional Requirements 25](#_Toc46356986)

[Chapter 5 System Design 28](#_Toc46356987)

[5.1 System Architecture 28](#_Toc46356988)

[5.2 Interface Design 28](#_Toc46356989)

[5.3 Data Flow Diagrams 28](#_Toc46356990)

[5.4 Data Dictionary 28](#_Toc46356991)

[Chapter 6 Implementation 31](#_Toc46356992)

[6.1 Module Description 31](#_Toc46356993)

[References 34](#_Toc46356994)

List of Figures

List of Tables

Abbreviations

# Introduction

Career Information and Recruitment Portal is a web application developed to connect colleges, students, alumni and recruiters on a single platform. It helps colleges for efficient management of their placement cell. Moreover, recruiters can use this application as a job board for their HR hiring activities related to campus recruitment. With this application, students and alumni can find and apply for job opportunities relevant to them and keep track of recruiter updates. Students and alumni can also stay informed about various career choices available for them using the information portal. Besides, the application allows students and alumni to request and receive recommendations from colleges, recruiters and fellow alumni. It helps them to add more value to their resumes.

## Problem Statement

HR recruitment had always been a hard time for recruiters to choose the right talent that best meets their expectations and perfectly matches their job vacancies. It is a lengthy and tiring process in many companies and usually spans over multiple weeks. The process requires a great deal of human effort, planning, strategy and time. Many companies still have not moved on for online recruitment. Also, companies which have opted for e-Recruitment have concerns regarding credibility and trustworthiness of the details submitted by an applicant. Security and ease of use of the application is another concern for a recruiter.

College students and fresher graduates are very resourceful and a worthwhile consideration for most of the job positions in a company. The student and graduate applicant pool are readily available and probably an economical choice for a company. It benefits both the employer and the student. Students might require an industry expertise certification for completion of their course. Additionally, it is a plus to their work profile. With student hiring, recruiters get an excellent opportunity to leverage their profits on a capable and agile worker pool, and also find employees of great potential for their company. Moreover, many colleges are looking out for recruiters who can offer campus placement opportunities for their students. Even with all the rising demand and added benefits to it, there is not yet an efficient platform to connect these stakeholders.

After completing their graduation or under-graduation, most students are in no man's land of thoughts while deciding to pursue their higher studies or get employed soon. Many talented graduates are jobless and do not know various possible career options for them and also does not have access to active job profiles that might fit their qualifications and skills. It is not an easy task for an individual to keep track of job updates and hiring activities carried out by a company. Moreover, having worthful and prised recommendations on their resume can increase the value of any candidate. However, receiving recommendations may not be very easy. There is a high chance that students might miss many potential opportunities to get hired by a company.

## Motivation and Objective

A web application to connect colleges, students, alumni and recruiters on a single platform is obviously, the right choice to solve problems related to campus recruitment. There would be no more hassle for colleges to find companies who can offer placement opportunities for their students and alumni. Managing job listings, tracking received job applications and communicating to applicants from a single platform can considerably reduce the effort of an HR manager at a company.

A web application to find and apply for jobs is an easy-to-use substitute for a manual and paper-based hiring process. It is more efficient to store job information and job applications on an online database which supports fast queries. Using a web application to interact with an online database saves time for both recruiters as well as students (and alumni). Besides, a paperless economy makes us one step closer to environmental sustainability.

All stakeholders would prefer a secure and reliable web application and would not want any trust issues or data compromises. Thanks to modern technologies that we have all the tools to secure a web application in the best possible way. Spring Security offers a standard for securing Spring-based web applications. The web application can allow colleges to verify student and alumni data before registering them on the platform. Moreover, the web application allows a server admin to verify all college and recruiter registrations on the platform. Thus, we can provide all stakeholders with reliable and trustworthy information.

Requesting and receiving recommendations can be made easy with a web application. Students and alumni can take advantage of a web application with an added feature for the recommendation system. With all the student (or alumni) details available on the platform, it is easy to generate an automatic CV for all the students (and alumni) on the platform. This resume can be easily attached to their job applications. Similar to the job search functionality, recruiters can use a resume search functionality to find relevant candidates for their job vacancies.

An information portal can be attached to this web application which allows students and alumni to discover various career and higher education opportunities available for them. It can help increase awareness and knowledge about career choices for a graduate. Hopefully, it can help us to build a society of less jobless people.

# Literature Review

## Existing Methodologies

# Proposed System and Feasibility Study

## Proposed Solution

## Technical Feasibility

## Operational Feasibility

## Economic Feasibility

## Schedule Feasibility

# Requirements Gathering and Analysis

A requirement is a necessary attribute in a system, a statement that identifies a capability, characteristic, or quality factor of a system in order for it to have value and utility to a customer or user [1, pp. 1 - 2]. Requirements gathering or requirements elicitation is the process of gaining an understanding of the customers’ and users’ needs for the planned system and their expectations of it [1, p. 4]. Requirements analysis is a structured (organized) method to understand the attributes that will satisfy a customer need [1, p. 222].

## End User Specification

End-users are people who uses a computer application, as opposed to those who developed or support it [2]. End user specifications are pre-requisite functionalities required by an end-user to use the developed system. End-user specifications for the proposed system are as follows:

* PC with minimum requirements
* Stable network connection

## Software Specification

Software specifications include the pre-requisite software required to develop the proposed system. Software specifications for the proposed system are as follows:

* Operating System:
* Windows 7+
* Mac OS X Yosemite 10.10+
* Linux: 64-bit Ubuntu 14.04+, Debian 8+, openSUSE 13.3+ or Fedora Linux 24+
* Front end: Thymeleaf
* Back end: Spring boot, Spring security and MongoDB
* Web server: Embedded Tomcat
* Supported browsers: Chrome, Firefox, Internet Explorer 9+
* Java JDK 1.8+

## Hardware Specification

Hardware specifications include the pre-requisite hardware required to develop the proposed system. Hardware specifications for the proposed system are as follows:

* 4 GHz minimum, multi-core processor
* RAM: Minimum 2 GB
* Hard disk space: Minimum 10 GB

## SRS Document

The software requirements specification (SRS) specifies the requirements for a computer software configuration item and the methods to be used to ensure each requirement has been met [3]. Benefits of documenting the SRS include [4]:

* It provides a realistic basis for estimating product costs**,** risks and schedules.
* It provides an informed basis for deploying a product to new users or new operational environments.
* It provides a basis for product enhancement.
* It forces a rigorous assessment of requirements before design can begin and minimizes later redesign.
* It establishes the basis for agreement between the acquirers or suppliers on what the product is to do (in market driven projects, the user input may be provided by marketing).
* Organizations can use the specifications to develop validation and verification plans.

Functional requirements and non-functional requirements for the proposed system according to the SRS document is as follows.

### Functional Requirements

A function is a useful capability provided by one or more components of a system. Functional requirements describe what the system or software must do [1, p. 72].

Functional requirements for the proposed system are as follows.

1. **Career Information**

**Description:** This section includes webpages for information related to local and international higher studies and job opportunities information for all branches. Users can click on the corresponding menu option to navigate to the respective page.

**Input:** User clicks on a menu option.

**Output:** Corresponding webpage is loaded in the web browser.

1. **Student, Alumni and Recruiters - Portal**
   1. **Colleges Portal**
      1. **Registration**

**Description:** Allow registration of new colleges. By clicking on register as a college link, the user receives the registration form to fill out college details. On submitting the filled application form, a new college registration request is sent to the server.

**Input:** User clicks new college registration, completes and submits the registration form.

**Output:** Server receives registration request on server homepage.

* + 1. **Opt Out**

**Description:** If the college admin wishes to remove the college from this portal, he/she can submit an opt out request by clicking the link for opt out.

**Input:** User submits opt out request.

**Output:** Server receives the request.

* + 1. **Login**

**Description:** Users can login using their registered email id and password. Server verifies the login details and grants access to the user session.

**Input:** User submits login credentials.

**Output:** Server verifies and grants or declines user session according to successful validation or failure of authentication.

* + 1. **Request for Forgot Password**

**Description:** User can request for resetting password, if he/she forgets the password, by clicking on the forgot password link. On clicking on forgot password link, the user will be prompted to submit the registered email id.

**Input:** User submits forgot password request for a registered email id.

**Output:** Server receives the request.

* + 1. **Update Details**

**Description:** Colleges can update their communication details by clicking on the update details link. It redirects to a form for editing the details. After making the necessary changes, the user can click on save button to save the changes.

**Input:** User makes changes in the college details using the edit form and clicks the save button.

**Output:** The new details are saved to the database.

* + 1. **Student and Alumni Management**
       1. **Add New Student Enrolments of the College to the Portal**

**Description:** Colleges can add multiple new students by uploading an excel file containing all the details of new student enrolments (name, contact number, email, communication address, educational qualifications) by clicking add new students and uploading the file.

**Input:** User uploads an excel spreadsheet of student details.

**Output:** All new student entries are added to the student database and students get login credentials via email.

* + - 1. **Add Alumni Students**

**Description:** Colleges can receive alumni registration requests on homepage. The college admin verifies the alumni details and approves or cancels the registration by clicking verify or cancel registration link. Applicants get notified by email regarding approval or cancellation of registration request.

**Input:** Alumni requests for registration.

**Output:** College admin verifies or cancels the registration by clicking verify or cancel registration button.

* + - 1. **Remove Alumni Students**

**Description:** Colleges can receive alumni opt out requests on homepage. On clicking on the request, he/she is directed to a list of all related records of the alumni out of which information to be retained can be selected. The admin can click on the delete alumni button to retain all selected records and remove other records.

**Input:** Alumni requests for opt out.

**Output:** College admin approves the request by selecting the records to keep and clicking delete alumni button.

* + - 1. **Update Student Details**

**Description:** When some student’s details are to be updated or semester marks to be uploaded the admin can either upload a new excel spreadsheet file of the details by clicking the edit student details in the students’ tab, or selecting individual student with email search and clicking on edit details. All records of the students with the login details specified in the spreadsheet will be updated.

**Input:** Student details are submitted via spreadsheet or edit details form.

**Output:** Updated student details are stored in the database.

* + - 1. **Recommend Student or Alumni**

**Description:** Colleges receive recommendation request from student or alumni on dashboard. The college admin can submit recommendation letter(s) from faculties by clicking on the request and selecting upload files.

**Input:** Student or alumni requests for recommendation.

**Output:** College admin uploads recommendation letter(s) from faculty.

* 1. **Student Portal**
     1. **Login**

**Description:** Users can login using their registered email id and password. Server verifies the login details and grants access to the user session.

**Input:** User submits login credentials.

**Output:** Server verifies and grants or declines user session according to successful validation or failure of authentication.

* + 1. **Request for Forgot Password**

**Description:** User can request for resetting password, if he/she forgets the password, by clicking on the forgot password link. On clicking on forgot password link, the user will be prompted to submit the registered email id.

**Input:** User submits forgot password request for a registered email id.

**Output:** Server receives the request.

* + 1. **Personalization**

**Description:** Students can add skills, interests, experience, project links, awards and honours, organizations, profile pic and a description to their profile by adding or editing details from the personalization tab. An automatic CV is generated with these details if the student does not upload a CV.

**Input:** Student edits personalization details from the personalization tab.

**Output:** Updated details are stored in the student’s database. Automatic CV is generated with these details if no uploaded CVs of the student are available.

* + 1. **Follow**

**Description:** Students can follow companies and topics of interests, and set notification preferences for each company or topic from the follow tab and selecting a company or topic and preference from the options indicated against it.

**Input:** Student selects a company or topic to follow and indicates notification preference for the same.

**Output:** The follow details are stored or updated in the student’s database.

* + 1. **Job Feed**

**Description:** Students receive job feed in their homepage according to the follow details in his/her student record. Students can click on each job listing and view its details.

**Input:** Student follow details from database.

**Output:** Job listings are shown in the student’s homepage according to the follow details.

* + 1. **Notifications**

**Description:** Students receive notifications in the notifications tab about new job listings from a company or a following topic according to his/her notification preferences for that company or topic.

**Input:** Student follow details from database.

**Output:** Job listing notifications are shown in the student’s notifications tab according to notification preferences in follow details.

* + 1. **Apply for Jobs**

**Description:** On clicking apply button on a job details page, students will be prompted to enter description for applying. This can be submitted by clicking the submit button. The recruiter gets the applicant’s CV and description for applying in his/her applications tab.

**Input:** Students submit a job application.

**Output:** Recruiter receives the applicant’s CV and description for applying in his/her applications tab.

* + 1. **Request Recommendations**

**Description:** Students can request for recommendations from his or her college or from an alumnus by searching for the name of the alumnus and submitting a request for recommendation by the request recommendation button. The college / alumnus receives the request on dashboard.

**Input:** Student requests recommendation from college or alumnus.

**Output:** College or alumni receives the request for recommendation dashboard.

* + 1. **Reply to Recruiters**

**Description:** Students can receive messages from recruiters and reply them in the chat feature.

**Input:** Student receives message from recruiters in the chat feature.

**Output:** Student replies to the message in the chat feature.

* 1. **Alumni Portal**
     1. **Registration**

**Description:** Allow registration of new alumni. By clicking on register as an alumnus link, the user receives the registration form to fill out personal details and college details. On submitting the filled application form, a new alumni registration request is sent to the respective college admin.

**Input:** User clicks new alumnus registration, completes and submits the registration form.

**Output:** College admin receives registration request on college homepage.

* + 1. **Opt Out**

**Description:** If an alumnus wishes to remove his company from this portal, he/she can submit an opt out request by clicking the link for opt out. College admin receives the request from alumni on dashboard.

**Input:** Alumni submits opt out request.

**Output:** College admin receives the request on his dashboard.

* + 1. **Login**

**Description:** Users can login using their registered email id and password. Server verifies the login details and grants access to the user session.

**Input:** User submits login credentials.

**Output:** Server verifies and grants or declines user session according to successful validation or failure of authentication.

* + 1. **Request for Forgot Password**

**Description:** User can request for resetting password, if he/she forgets the password, by clicking on the forgot password link. On clicking on forgot password link, the user will be prompted to submit the registered email id.

**Input:** User submits forgot password request for a registered email id.

**Output:** Server receives the request.

* + 1. **Personalization**

**Description:** Alumni can add or update personal details, higher education details, skills, interests, experience, project links, awards and honours, organizations, profile pic and a description to their profile by adding or editing details from the personalization tab. An automatic CV is generated with these details if the alumnus does not upload a CV.

**Input:** Alumnus edits personalization details from the personalization tab.

**Output:** Updated details are stored in the alumni database. Automatic CV is generated with these details if no uploaded CVs of the alumnus are available.

* + 1. **Follow**

**Description:** Alumni can follow companies and topics of interests, and set notification preferences for each company or topic from the follow tab and selecting a company or topic and preference from the options indicated against it.

**Input:** Alumnus selects a company or topic to follow and indicates notification preference for the same.

**Output:** The follow details are stored or updated in the alumni database.

* + 1. **Job Feed**

**Description:** Alumni receive job feed in their homepage according to the follow details in his/her record. He/she can click on each job listing and view its details.

**Input:** Student follow details from database.

**Output:** Job listings are shown in the alumnus’s homepage according to the follow details.

* + 1. **Notifications**

**Description:** Alumni receive notifications in the notifications tab about new job listings from a company or a following topic according to his/her notification preferences for that company or topic.

**Input:** Alumni follow details from database.

**Output:** Job listing notifications are shown in the alumni’s notifications tab according to notification preferences in follow details.

* + 1. **Apply for Jobs**

**Description:** On clicking apply button on a job details page, alumni will be prompted to enter description for applying. This can be submitted by clicking the submit button. The recruiter gets the applicant’s CV and description for applying in his/her applications tab.

**Input:** Alumni submit a job application.

**Output:** Recruiter receives the applicant’s CV and description for applying in his/her applications tab.

* + 1. **Request Recommendations**

**Description:** Alumni can request for recommendations from his or her college or from a fellow alumnus by searching for the name of the alumnus and submitting a request for recommendation by the request recommendation button. The college / alumnus receives the request on dashboard.

**Input:** Alumni requests recommendation from college or alumnus.

**Output:** College or alumni receives the request for recommendation dashboard.

* + 1. **Recommend Students or Fellow Alumni**

**Description:** Alumni receive recommendation request from students or fellow alumni on dashboard. He/she can submit a recommendation letter by clicking on the request and selecting upload files.

**Input:** Student or alumni requests for recommendation.

**Output:** The alumni uploads a recommendation letter.

* + 1. **Reply to Recruiters**

**Description:** Alumni can receive messages from recruiters and reply them in the chat feature.

**Input:** Alumni receives message from recruiters in the chat feature.

**Output:** Alumni replies to the message in the chat feature.

* 1. **Recruiter Portal**
     1. **Registration**

**Description:** Allow registration of new recruiters. By clicking on register as a recruiter link, the user receives the registration form to fill out company details. On submitting the filled application form, a new company registration request is sent to the server.

**Input:** User clicks new recruiter registration, completes and submits the registration form.

**Output:** Server receives registration request on server homepage.

* + 1. **Login**

**Description:** Users can login using their registered email id and password. Server verifies the login details and grants access to the user session.

**Input:** User submits login credentials.

**Output:** Server verifies and grants or declines user session according to successful validation or failure of authentication.

* + 1. **Request for Forgot Password**

**Description:** User can request for resetting password, if he/she forgets the password, by clicking on the forgot password link. On clicking on forgot password link, the user will be prompted to submit the registered email id.

**Input:** User submits forgot password request for a registered email id.

**Output:** Server receives the request.

* + 1. **Opt Out**

**Description:** If a recruiter wishes to remove his company from this portal, he/she can submit an opt out request by clicking the link for opt out.

**Input:** User submits opt out request.

**Output:** Server receives the request.

* + 1. **Update Details**

**Description:** Colleges can update their communication details by clicking on the update details link. It redirects to a form for editing the details. After making the necessary changes, the user can click on save button to save the changes.

**Input:** User makes changes in the college details using the edit form and clicks the save button.

**Output:** The new details are saved to the database.

* + 1. **Create Job Listings**

**Description:** Recruiters can create job listings by clicking listings tab and create new button. He/she will be prompted to enter a job title and description, salary offered and add all related tags (topics) for the job. On clicking submit button, a new job listing will be created.

**Input:** Recruiter submits job details for new job listing.

**Output:** A new job listing is created.

* + 1. **Edit / Delete Job Listing**

**Description:** Recruiters can delete job listings by clicking listings tab and selecting an already created job listing. He/she will be directed to an edit details form. By clicking update button or delete button, the job listing will be updated or deleted accordingly.

**Input:** Recruiter clicks update or delete button on a job listing.

**Output:** The job listing is updated or deleted according to the button clicked.

* + 1. **Review CV**

**Description:** Recruiters can view CV of all students and alumni on the portal following tags associated with job listings of the company from the Potential tab. The student profiles (name, profile picture, and description) will be listed in a sorted order according to a score calculated based on work experience, educational qualifications, skills and recommendations. On clicking a student’s or alumnus’s name, the recruiter will be directed the student’s or alumni’s CV. From this page, the recruiter can click on contact button to message the student or alumnus regarding recruitment. The webpage directs to the chat tab with the student or alumni and recruiter can type his message and click on send button.

**Input:** Recruiter selects a CV and sends a message to the student or alumni.

**Output:** The student or alumni receives the messages in the chat tab.

* + 1. **Review Applications**

**Description:** Recruiters can view all received application for a job listing from the applications page of the listing. The description for application along with student profile (name, profile picture) will be listed in a sorted order according to a score calculated based on work experience, educational qualifications, skills and recommendations. On clicking a student’s or alumnus’s name, the recruiter will be directed the student’s or alumni’s job application. He/she can click on view CV button to view the CV of the candidate. The recruiter can click on contact button to message the student or alumnus regarding recruitment. The webpage directs to the chat tab with the student or alumni and recruiter can type his message and click on send button.

**Input:** Recruiter selects a job application and sends a message to the student or alumni.

**Output:** The student or alumni receives the messages in the chat tab.

* + 1. **Recommend Student or Alumni Employees**

**Description:** Recruiters can recommend current or past employees of the company registered as student or alumni in the portal by searching for the name of the employee and selecting recommend button. The recruiter can upload a recommendation letter by clicking on upload files on the redirected page.

**Input:** Recruiter selects a current or previous employee of the company and uploads a recommendation letter.

**Output:** The student or alumni receives the recommendation in his profile.

* 1. **Server Admin**
     1. **Verify / Cancel College and Recruiter Registration**

**Description:** The server admin personally verifies details of registration of college users and recruiter users before allowing them login access. All new college and recruiter registrations pending verification will be displayed on the server homepage. The server admin can click on each registration to view the details submitted. If the admin finds the registration details are correct (credible), he/she can click verify registration button to verify the college or recruiter registration; otherwise, the admin can click cancel button to cancel the registration. On approval or cancellation of registration, email messages are automatically sent to the respective users. This is used for ensuring credibility of registered colleges and recruiters.

**Input:** Colleges and recruiters submit registration.

**Output:** Server admin clicks on new registrations of colleges and recruiters, verifies the details and clicks verify or cancel registration button. Autogenerated emails are sent to the respective users regarding confirmation or cancellation of registration.

* + 1. **Handle Opt Out Requests of Colleges and Recruiters**

**Description:** When colleges or recruiters request for opting out from the application, the server admin gets the request on the server homepage. On clicking on the request, he/she is directed to a list of all related records of the user out of which information to be retained can be selected. The admin can click on the delete user button to retain all selected records and remove other records.

**Input:** Colleges and recruiters request for opt out.

**Output:** Opt out request is displayed on server homepage. Admin clicks on the request to view all related records. He/she selects records to be retained and deletes all other records by clicking on delete user button.

* + 1. **Close Active Login Sessions with No Activity for a Long Duration**

**Description:** When user session is inactive for over 15 minutes, the session is automatically closed by the server.

**Input:** User session remains inactive for 15 minutes.

**Output:** Server closes the request.

* + 1. **Handle Forgot Password Requests**

**Description:** When a user submits forgot password request, the server automatically sends a password reset link to the registered email of the user.

**Input:** User submits forgot password request.

**Output:** Server automatically sends a password reset link to the registered email of the user.

### Non-Functional Requirements

Users have implicit expectations about how well the software will work. These characteristics include how easy the software is to use, how quickly it executes, how reliable it is, and how well it behaves when unexpected conditions arise. The non-functional requirements define these aspects about the system [5].

The non-functional requirements for the proposed system are as follows.

1. **Portability:**

Since this is a web application it can be used from all supported browsers irrespective of the underlying operating system.

1. **Security:**

The web application is developed using Spring boot framework for Java, Spring security core, and MongoDB RESTful authentication is used in the back end. The APIs are guarded by spring security REST and JWT. This is a great combination of a secure modern web application. The database used is highly secured and protected from unauthorized access.

1. **Usability:**

The web application will be available all the time since it is hosted on a real time server. It has a simple and user-friendly interface. Both skilled and unskilled users can use the application efficiently to meet their requirements. The user interface is designed to make the user to interaction simple and easy as possible.

1. **Reliability:**

Performance and reliability are two key components of the system. The application offers high performance and reliability of data and information as well as transactions of the database.

1. **Maintainability:**

The database used provides high performance. It can be used to store files of any size easily without any complications. In the case of a failure, it is easy to be administered. The system can be easily maintained by the admin users.

1. **General Constraints on Design and Development:**

The software and technologies used for development are Java, Spring boot framework for Java, Angular 7 and MongoDB database.

# System Design

## System Architecture

## Interface Design

## Data Flow Diagrams

## Data Dictionary

# Implementation

## Module Description

# References

|  |  |
| --- | --- |
| [1] | R. R. Young, The Requirements Engineering Handbook, Norwood, MA, USA: Artech House, 2003. |
| [2] | D. Howe, "FOLDOC entry for "end-user"," 29 03 1997. [Online]. Available: https://foldoc.org/end-user. [Accessed 22 07 2020]. |
| [3] | "DI-IPSC-81433A, DATA ITEM DESCRIPTION: SOFTWARE REQUIREMENTS SPECIFICATION (SRS)," EverySpec, 15 December 1999. [Online]. Available: http://everyspec.com/DATA-ITEM-DESC-DIDs/DI-IPSC/DI-IPSC-81433A\_3709/. [Accessed 07 22 2020]. |
| [4] | D. Garbar, "SRS document Helps to Protect IT Projects From Failure," BELITSOFT, 8 August 2016. [Online]. Available: https://belitsoft.com/php-development-services/software-requirements-specification-helps-protect-it-projects-failure. [Accessed 22 07 2020]. |
| [5] | A. Stellman and J. Greene, "Glossary," in *Applied Software Project Management*, O'Reilly Media, 2014. |