**SOFTWARE REQUIREMENTS SPECIFICATION DOCUMENT**

**TOPIC: CAMPUS RECRUITMENT SYSTEM.**

**DONE BY – RIYA NILESH CHHADVA**

**GR NUMBER – 21810811**

**ROLL NUMBER – 324016**

**DIVISION – D**

* **TABLE OF CONTENTS :**

|  |  |  |
| --- | --- | --- |
| **Introduction** | **Overall**  **Description** | **External Interface Requirements** |
| Purpose | Product Perspective | User Interfaces |
| Document Conventions | Product Functions | Hardware Interfaces |
| Intended Audience | User Classes and Characteristics | Software Interfaces |
| Product Scope | Operating Environment | Communication Interfaces |
|  | Design and Implementation Constraints |  |
|  | User Documentation |  |
|  |  |  |

* **INTRODUCTION :**
* **PURPOSE :**
* The primary purpose to develop this system is to optimize the recruitment process for college.
* Besides, the qualified applicants could be sort by this system, based on their qualifications and company requirements.
* Based on the applicant’s skills and areas of interest, the company suitable or the company in which he/she is going to place can be predicted.
* Another purpose of the software is to facilitate the student and the company to register and communicate with placement office.
* **DOCUMENT CONVENTIONS :**
* The font used in this document is “Times New Roman having size 16”.
* The text is made bold for the highlighting purpose so that it can be easily differentiated.
* In this document, every functionality is equally important and every requirement has its own priority.
* **INTENDED AUDIENCE :**
* The document is meant for the student of the college, the company and the admin will operate the system.
* This document will serve as a reference document for the project management and development team who will analyze, design and implement the system.
* **PRODUCT SCOPE :**
* Campus Recruitment System enables the user to have the typical recruitment facilities and features at their disposal.
* It resolves the typical issue of manual staffing processes and activities into a controlled and closely monitored work flow in the architecture of the application.
* The objective of this application is to serve as a common meeting ground for jobseekers and company, locally.
* This kind of system is specifically designed for organization to help in solving staffing problems and managing human resource department activities at higher degree of optimization.
* **OVERALL DESCRIPTION :**
* **PRODUCT PERSPECTIVE :**
* In the “Campus Recruitment System”, we will be having a user interface for the admin, student and company.
* The user can log in to their account.
* Depending on the user, the utilities will be decided.
* The student can login to their respective account and can apply for the respective post available.
* If the user doesn’t have an account the he/she can create an account and can apply for the jobs.
* The company can login to their account and can update the details of the jobs available.
* If the company doesn’t have an account then they can create an account and can update the job details.
* Admin will already be having an ID and password.
* The admin will login and manage the student and companies activities.
* After the creation or the login of the user, a successful message will be displayed to the user.

OUTPUT MESSAGE

INPUT MESSAGE

FORMATTED DATA

USER RAW DATA

USER LOGIN INFORMATION

SUCCESS MESSAGE

CAMPUS RECRUITMENT SYSTEM

USER INTERFACE

[ ADMIN,

STUDENT,

COMPANY ]

* **PRODUCT FUNCTIONS :**
* The below Data Flow Diagram [DFD] clearly shows various process associated with the project and how the data flows between the entities and database.
* The student can create the account.
* If the account is already present then he/she can search for the available jobs, can view the jobs and also apply for the jobs.
* Further he/she can update the details, can change the password etc.
* The company can create the account.
* If the company already have an account then the company can post the job and can also view how many candidates have applied for the jobs of that company.
* The admin can manage the student and the company.
* Admin validates the account of the student.
* Admin can also remove company and the student if found any discrepancy.
* Admin manages all activities of company and student.

**Data Flow Diagram [DFD]:**

USER

COMPANY INFO.

STUDENT INFO.

DETAILS

USER ID/ PASSWORD

LOGIN

Company Account Info.

Student Account Info.

3.0 SIGN UP

2.0

TYPE SELECTION

1.0 VALIDATION

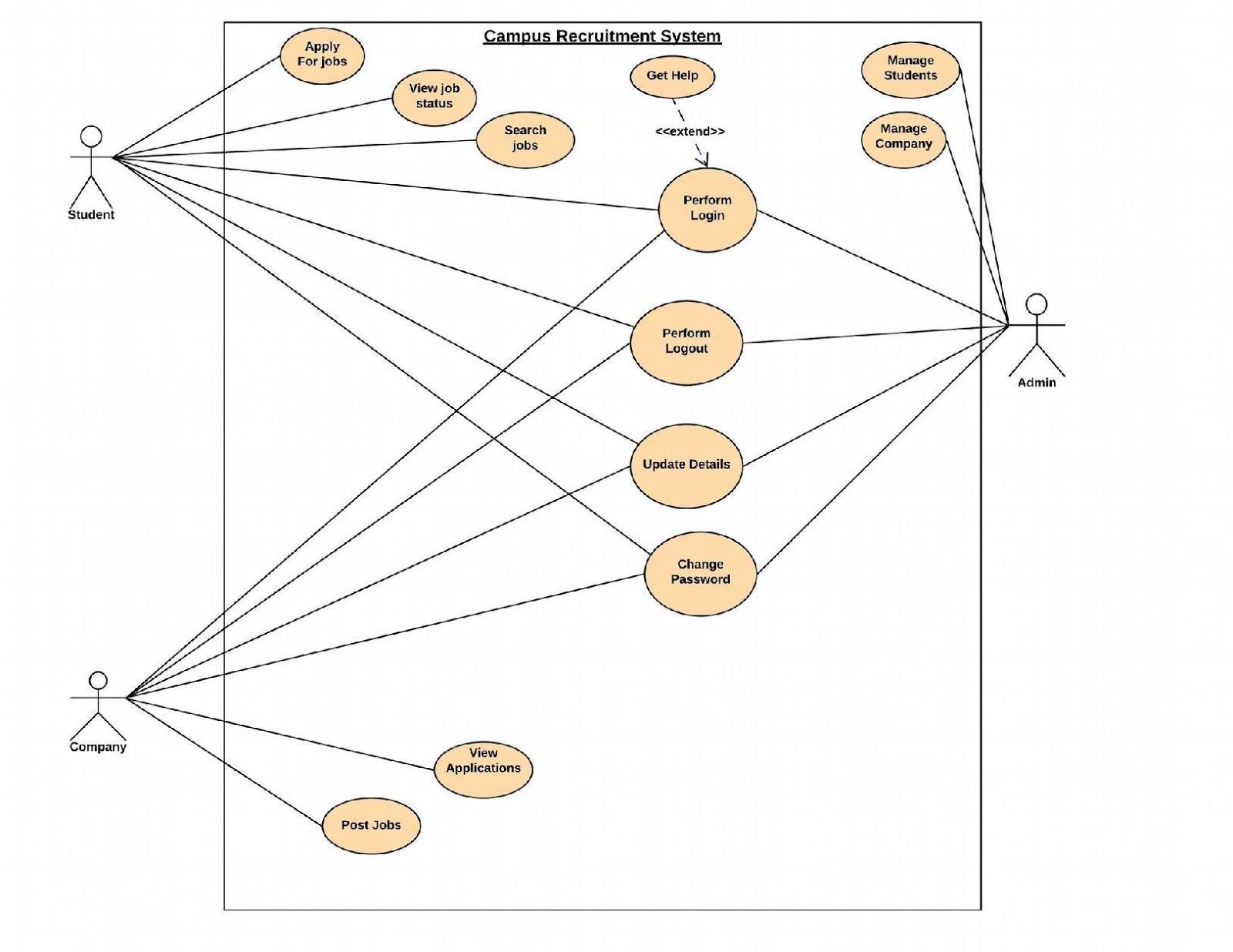
Success/ Failure Message

User ID/ Password

5.0 UTILITIES

4.0 VERIFY AUTHENTICATION

* **USER CLASSES AND CHARACTERISTICS :**
* The actors and use cases are clearly shown in below **use case** diagram:



* In software and systems engineering , a **use case** is a list of actions or event steps typically defining the interactions between a role (known in the Unified Modeling Language as an actor) and a system to achieve a goal.
* The actor can be a human or other external system.
* In systems engineering use cases are used at higher level than within software engineering often representing missions or stakeholder goals.
* The following are the actors who perform the use cases as stated above:

|  |  |  |
| --- | --- | --- |
| **S.No** | **Actor Name** | **Description / Actor’s Role** |
| 01 | ADMIN | Manage students and companies. |
| 02 | STUDENT | Apply for jobs, view job status & search jobs. |
| 03 | COMPANY | Post jobs & view applications of students. |

- The following section describes the Use Cases with Pre and Post conditions :

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **S.No** | **Use case Name** | **Description** | **Pre-condition** | **Post condition** |
| 01 | Perform Login | User(admin,student or company) can perform login. | User having account | Login successful |
| 02 | Perform Logout | User(admin, student or company) can perform logout. | User logged in | Successfully logged out |
| 03 | Update Details | User(admin, student or company) can update their details. | User logged in | View details |
| 04 | Change Password | User(admin, student or company) can change passwords. | User having password. | Password is reset. |
| 05 | Manage Company | Admin can manage companies | Existing admin & atleast one company | Validated by admin & company can  continue with it’s  account |
| 06 | Manage Students | Admin can manage students | Existing admin & at least one student | Validated by admin & student can  continue with it’s account |
| 07 | Apply for jobs | Student can apply for job | Job should be there posted by company &  student must be eligible for job | Applied for job successfully & wait  for response from company |
| 08 | View Job Status | Student can view job status | Student must have successfully applied for job. | Viewed job status & can accept job if selected else can apply for other job. |
| 09 | Search jobs | Student can search for job | Student must have account & logged in. | Can find a job or not. |
| 10 | Post Jobs | Company can post jobs | Company must have account & logged in. | Student can now apply for jobs. |
| 11 | View  Applications | Company can view applications of students. | Atleast one student must have applied. | Company can react to applications. |
| 12 | Get Help | User can get help for login | User should have  tried for it or just get help if does not know how to login | User will now login using this help. |

* **OPERATING ENVIRONMENT :**
* The campus recruitment system shall be deployed on a smartphone like S8 Edge Galaxy, G5-Plus and server running on android Operating System Version.
* SQLite Database shall be used to maintain the databases.
* The system shall be accessed & downloaded from google play Store.
* It can run without wifi even.
* Ths system will be built with the help of following **Softwares :**
* **IDE (Android Studio) :**
* Android Studio is the official Integrated Development Environment (IDE) for Android app development, based on IntelliJ Idea.
* On top of IntelliJ’s powerful code editor and developer tools, Android Studio offers even more features that enhance your productivity when building Android apps such as:

1. A flexible Gradle based build system.
2. A fast and feature rich emulator.
3. A unified environment where you can develop for all Android devices.
4. Instant Run to push changes to your running app without building a new APK.
5. Code templates and Github integration to help you build common app features and import sample code.
6. Extensive testing tools and frameworks.
7. Lint tools to catch performance, usability, version compatibility, and other problems.
8. C++ and NDK support.
9. Built-in support for Google Cloud Platform, making it easy to integrate Google Cloud Messaging and App Engine.
10. In Android Studio, Java Language would be used.

* **DATABASE (SQLite) :**
* SQLite in android will be used for saving data to a database is ideal for repeating or structured data, such as contact information.
* We used SQLite in development & coding part for storing database.
* We used SQLite in integration and testing part if during testing software fails or more database to be added.
* **JAVA (Programming Environment) :**
* Java is a programming Language originally developed by Sun Microsystems and released in 1995.
* **DESIGN AND IMPLEMENTATION CONSTRAINTS :**
* The system is designed to be the cross platform supportable.
* The system is supported on a wide range of hardware and any android platform which is having any version of android built into the system.
* This application is being developed using android studio, hence it is extremely portable.
* To prevent multiple students of the same speciality to log-in onto same company portal. Request, a workflow system needs to be designed which routes the company vacancy . Requests to students.
* System is expected to store maximum 64GB of data.
* **USER DOCUMENTATION :**
* The following documents shall be prepared:

1. Installation Guide.
2. User Manual for end users.
3. Even video tutorials of installation of campus recruitment will be provided as DVD.

* **EXTERNAL INTERFACE REQUIREMENTS :**
* **USER INTERFACES :**
* The user interface section defines the way various stakeholders interact with the system.
* All the screens will be developed to work on android mobile.
* Error messages will appear as a popup on the screen.
* The maximum size of error message will be 40 characters.
* Buttons will there to make the navigation simpler.
* A first time user of the mobile should see the login screen when he/she will open the android application.
* If the user has not registered to the screen, then he/she should be able to re-direct to the sign up page from login screen.
* Every user should have the profile where he/she can apply for the job.
* After the creation of account the user can login to the application and will be able to apply for the jobs.
* **HARDWARE INTERFACES :**
* Since the mobile application does not have designated hardware, it does not have any hardware interface.
* The hardware connection between the database server and application is managed by the underlying operating system on the mobile phone.
* **SOFTWARE INTERFACES :**
* The system is self-contained and no data is supposed to share with the third party.
* The communication of the mobile application between the database consist of both reading and modifying the data, while the communication between the database and the mobile application consists of only reading operation.
* **COMMUNICATION INTERFACES :**
* The communication between the different parts of the system is important since they depend on each other.
* However, in what way the communication is achieved is not important for the system and is therefore handled by the underlying operating system for the mobile application.