****

**COMSATS University, Islamabad Pakistan**

**Career Development Buddy**

**(CDB)**

***By***

**Shehryar Afgan CIIT/SP19-BCS-033/ISB**

**Shah Fahad CIIT/FA18-BSE-080/ISB**

***Supervisor*Mr. Waseem Abbas**

***Bachelor of Science in Computer Science***

***(2019-2023)***

**The candidate confirms that the work submitted is their own and appropriate  
 credit has been given where reference has been made to the work of others**.

****

**COMSATS University, Islamabad Pakistan**

**Career Development Buddy**

**(CDB)**

**A project presented to**

**COMSATS University, Islamabad**

**In partial fulfillment**

**Of the requirement for the degree of**

***Bachelors of Science in Computer Science***

***(2019-2023)***

**By**

**Shehryar Afgan CIIT/SP19-BCS-033/ISB**

**Shah Fahad CIIT/FA18-BSE-080/ISB**

**DECLARATION**

We hereby declare that this software, neither whole nor as a part has been copied out from any source. It is further declared that we have developed this software and accompanied report entirely on the basis of our personal efforts. If any part of this project is proved to be copied out from any source or found to be reproduction of some other, we will stand by the consequences. No Portion of the work presented has been submitted of any application for any other degree or qualification of this or any other university or institute of learning.

--------------------------- ---------------------------

Shehryar Afgan Shah Fahad

**CERTIFICATE OF APPROVAL**

It is to certify that the final year project of BS (CS); **Career Development Buddy** was developed by **Shehryar Afgan (CIIT/SP19-BCS-033)** and **Shah Fahad (CIIT/FA18-BSE-080)** under the supervision of Mr. Waseem Abbas and that in his opinion; it is fully adequate, in scope and quality for the degree of Bachelors of Science in Computer Sciences.

---------------------------------------

**Supervisor**

---------------------------------------

**External Examiner**

---------------------------------------

**Head of Department**

**(Department of Computer Science)**

**Executive Summary**

In a most of the countries around the world, career counseling and unemployment is a major issue. With increasing population, finding a job can be very tedious task. ‘Career Development Buddy’ helps its users to shape up their career and later on to help in finding a suitable job for the user in future. Helping the user to find the job would be made much easier by this proposed system.

Proposed system has three users; student, admin and employer. After conducting an aptitude test for the student, the system would help him clear his about his strength and weaknesses via a detailed summary report. Top universities according to the user preferences entered will be recommended to user. The shortlisted universities for which he is eligible to apply would be displayed. From a single platform; user will get information of which universities he can apply for by one stop admission criteria check. Details of the university along with mock test will be available for the user. Learning materials will be provided to the student to make his concepts clear and aiding him in his preparation for admission tests. Moreover, the presence of an AI chat-bot will assist to answer user queries relating to career counseling domain. With the help of this system, user would have a clear mindset in selecting the specific field of their choice from the best institute. In addition, this system will also help them in finding a suitable place for them in job market and help employers in finding the employee suitable for their job. Other features such as online sessions i.e audio/video call with employer are also available in order to facilitate employer in conducting remote interview/ session in case of remote jobs.

**Acknowledgement**

All praise is to Almighty Allah who bestowed upon us a minute portion of His boundless knowledge by virtue of which we were able to accomplish this challenging task.

We are greatly indebted to our project supervisor Mr. Waseem Abbas. Without their personal supervision, advice and valuable guidance, completion of this project would have been doubtful. We are grateful to them for their encouragement and continual help during this work.

And we are also thankful to our parents and family who have been a constant source of encouragement for us and brought us with the values of honesty & hard work.

Shehryar Afgan Shah Fahad

--------------------------- ---------------------------

**Abbreviations**

|  |  |
| --- | --- |
| **SRS** | Software Require Specification |
| **SDS** | Software Design Specification |
| **ST** | Software Testing |
| **CDB** | Career Development Buddy |
| **UC** | Use Case |
| **FR** | Functional Requirement |
| **NFR** | Non-functional Requirement |

**Table of Contents**

[1 Introduction 1](#_Toc121328736)

[1.1 Vision Statement 1](#_Toc121328737)

[1.2 Related System Analysis/Literature Review 1](#_Toc121328738)

[1.3 Project Deliverables 2](#_Toc121328739)

[1.4 System Limitations/Constraints 2](#_Toc121328740)

[1.5 Tools and Technologies 2](#_Toc121328741)

[1.6 Relevance to Course Modules 3](#_Toc121328742)

[2 Problem Definition 4](#_Toc121328743)

[2.1 Problem Statement 4](#_Toc121328746)

[2.2 Problem Solution 4](#_Toc121328747)

[2.3 Objectives of the Proposed System 4](#_Toc121328748)

[2.4 Scope 5](#_Toc121328749)

[2.5 Modules 5](#_Toc121328750)

[2.5.1 Module 1: User Profiling 5](#_Toc121328751)

[2.5.2 Module 2: Virtual Sessions 5](#_Toc121328752)

[2.5.3 Module 3: Aptitude test 6](#_Toc121328753)

[2.5.4 Module 4: Institute recommendation 6](#_Toc121328754)

[2.5.5 Module 5: Entry Test Preparation 6](#_Toc121328755)

[2.5.6 Module 6: AI chat-bot 7](#_Toc121328756)

[2.5.7 Module 7: One-Stop admission meeting criteria check 7](#_Toc121328757)

[2.5.8 Module 8: Jobs portal 7](#_Toc121328758)

[3 Requirement Analysis 8](#_Toc121328759)

[3.1 User classes and characteristics 8](#_Toc121328761)

[3.2 Requirement Identifying Technique 9](#_Toc121328762)

[3.2.1 Use-Case Diagram: 9](#_Toc121328763)

[3.2.2 Tabular Use-case 12](#_Toc121328764)

[3.2.2.1 Use-Case 1.1 - Sign up 12](#_Toc121328765)

[3.2.2.2 Use-Case 1.2 - Sign in 13](#_Toc121328766)

[3.2.2.3 Use-Case 1.3 – Forgot Password 14](#_Toc121328767)

[3.2.2.4 Use-Case 1.4 – Log out 15](#_Toc121328768)

[3.2.2.5 Use-Case 1.5 – Update Account Info 15](#_Toc121328769)

[3.2.2.6 Use-Case 2.1 – Place Audio Call 16](#_Toc121328770)

[3.2.2.7 Use-Case 2.2 – Receive Voice call 17](#_Toc121328771)

[3.2.2.8 Use-Case 2.3 – Place Video Call 18](#_Toc121328772)

[3.2.2.9 Use-Case 2.4 – Receive Audio Call 18](#_Toc121328773)

[3.2.2.10 Use-Case 2.5 – Call Logs 19](#_Toc121328774)

[3.2.2.11 Use-Case 3.1 – Generate Test 19](#_Toc121328775)

[3.2.2.12 Use-Case 3.2 – View Test Results 20](#_Toc121328776)

[3.2.2.13 Use-Case 3.3 – Generate Result Card 21](#_Toc121328777)

[3.2.2.14 Use-Case 4.1 – View University 21](#_Toc121328778)

[3.2.2.15 Use-Case 4.2 – Manage Preferences 22](#_Toc121328779)

[3.2.2.16 Use-Case 4.3 – Get University recommendation 23](#_Toc121328780)

[3.2.2.17 Use-Case 4.4 – Compare University 24](#_Toc121328781)

[3.2.2.18 Use-Case 4.5 – Contact University 25](#_Toc121328782)

[3.2.2.19 Use-Case 4.6 – FAQ’s 26](#_Toc121328783)

[3.2.2.20 Use-Case 5.1- Mock Test 27](#_Toc121328784)

[3.2.2.21 Use-Case 5.2 – Learning Material 28](#_Toc121328785)

[3.2.2.22 Use-Case 5.3-Discussion Form 28](#_Toc121328786)

[3.2.2.23 Use-Case 5.4-Past Papers 29](#_Toc121328787)

[3.2.2.24 Use-Case 6.1 – Ask Query 30](#_Toc121328788)

[3.2.2.25 Use-Case 6.2 – Get university Rankings 30](#_Toc121328789)

[3.2.2.26 Use-Case 6.3 – Top Careers 31](#_Toc121328790)

[3.2.2.27 Use-Case 6.4 – User feedback 31](#_Toc121328791)

[3.2.2.28 Use-Case 7.1 – Add Academic Record 32](#_Toc121328792)

[3.2.2.29 Use-Case 7.2 – Update Academic Record 33](#_Toc121328793)

[3.2.2.30 Use-Case 7.3 – Calculate aggregate 34](#_Toc121328794)

[3.2.2.31 Use-Case 7.4 – Display Applicable Institutes 35](#_Toc121328795)

[3.2.2.32 Use-Case 7.5 – Get Admission guidelines 36](#_Toc121328796)

[3.2.2.33 Use-Case 7.6 - Dashboard 37](#_Toc121328797)

[3.2.2.34 Use-Case 8.1 – Display Job Listings 38](#_Toc121328798)

[3.2.2.35 Use-Case 8.2–Filter Job Listings 39](#_Toc121328799)

[3.2.2.36 Use-Case 8.4–Upload Attachments 40](#_Toc121328800)

[3.2.2.37 Use-Case 8.5– Job Apply 41](#_Toc121328801)

[3.2.2.38 Use-Case 8.6–Manage Job Listing 42](#_Toc121328802)

[3.2.2.39 Use-Case 8.7–View Applicant Details 43](#_Toc121328803)

[3.3 Functional Requirements 44](#_Toc121328804)

[3.3.1 Enter First name 44](#_Toc121328805)

[3.3.2 Enter Last name 44](#_Toc121328806)

[3.3.3 Enter email 45](#_Toc121328807)

[3.3.4 Upload profile picture 45](#_Toc121328808)

[3.3.5 Add contact number 46](#_Toc121328809)

[3.3.6 Enter password 46](#_Toc121328810)

[3.3.7 Confirm password 47](#_Toc121328811)

[3.3.8 Sign-up 47](#_Toc121328812)

[3.3.9 Sign-in 47](#_Toc121328813)

[3.3.10 Forgot password 48](#_Toc121328814)

[3.3.11 Enter new password 48](#_Toc121328815)

[3.3.12 Confirm new password 49](#_Toc121328816)

[3.3.13 Confirm Log out 49](#_Toc121328817)

[3.3.14 Update Account Info 49](#_Toc121328818)

[3.3.15 Enter updated contact number 50](#_Toc121328819)

[3.3.16 Update profile picture 50](#_Toc121328820)

[3.3.17 Enter old password 51](#_Toc121328821)

[3.3.18 Select Role 51](#_Toc121328822)

[3.3.19 View University 52](#_Toc121328823)

[3.3.20 Select university 52](#_Toc121328824)

[3.3.21 Add preferred field 52](#_Toc121328825)

[3.3.22 Add preferred Location 53](#_Toc121328826)

[3.3.23 Update preferred field 53](#_Toc121328827)

[3.3.24 Update preferred Location 54](#_Toc121328828)

[3.3.25 Show university recommendation 54](#_Toc121328829)

[3.3.26 Enter first university name 54](#_Toc121328830)

[3.3.27 Enter second university name 55](#_Toc121328831)

[3.3.28 Compare university 55](#_Toc121328832)

[3.3.29 Compare universities courses 55](#_Toc121328833)

[3.3.30 Compare universities statistics 56](#_Toc121328834)

[3.3.31 Send email 56](#_Toc121328835)

[3.3.32 Ask Question 57](#_Toc121328836)

[3.3.33 View FAQ form 57](#_Toc121328837)

[3.3.34 Add matric marks 57](#_Toc121328838)

[3.3.35 Add Fsc marks 58](#_Toc121328839)

[3.3.36 Add School 58](#_Toc121328840)

[3.3.37 Add College 59](#_Toc121328841)

[3.3.38 Save Academic Details 59](#_Toc121328842)

[3.3.39 Edit Matric marks 59](#_Toc121328843)

[3.3.40 Edit FSC Marks 60](#_Toc121328844)

[3.3.41 Edit school 60](#_Toc121328845)

[3.3.42 Edit college 61](#_Toc121328846)

[3.3.43 Compute Aggregate 61](#_Toc121328847)

[3.3.44 Show Applicable Institutes 61](#_Toc121328848)

[3.3.45 Select University 62](#_Toc121328849)

[3.3.46 View Admission Schedule 62](#_Toc121328850)

[3.3.47 How to Apply? 62](#_Toc121328851)

[3.3.48 Show Job Listings 63](#_Toc121328852)

[3.3.49 Filter Job Listings by date 63](#_Toc121328853)

[3.3.50 Filter Job Listings by location 64](#_Toc121328854)

[3.3.51 Filter Job Listings by Employer Name 64](#_Toc121328855)

[3.3.52 Upload CV 64](#_Toc121328856)

[3.3.53 Upload Matric Transcript 65](#_Toc121328857)

[3.3.54 Upload Fsc Transcript 65](#_Toc121328858)

[3.3.55 Select Job Listing 66](#_Toc121328859)

[3.3.56 Apply Job 66](#_Toc121328860)

[3.3.57 Add Job title 66](#_Toc121328861)

[3.3.58 Add Job Location 67](#_Toc121328862)

[3.3.59 Add Job experience required 67](#_Toc121328863)

[3.3.60 Enter Job Description 68](#_Toc121328864)

[3.3.61 Job availability status 68](#_Toc121328865)

[3.3.62 Cancel Job Listing 68](#_Toc121328866)

[3.3.63 View Applicant CV 69](#_Toc121328867)

[3.3.64 Mic request 69](#_Toc121328868)

[3.3.65 Allow mic access 70](#_Toc121328869)

[3.3.66 Allow camera access 70](#_Toc121328870)

[3.3.67 Place audio call 70](#_Toc121328871)

[3.3.68 Place video call 71](#_Toc121328872)

[3.3.69 Decline audio call 71](#_Toc121328873)

[3.3.70 Decline video call 71](#_Toc121328874)

[3.3.71 Call Feedback 72](#_Toc121328875)

[3.3.72 Display Call log 72](#_Toc121328876)

[3.3.73 Generate call Id 73](#_Toc121328877)

[3.3.74 Copy call Id 73](#_Toc121328878)

[3.3.75 Start Test 73](#_Toc121328879)

[3.3.76 End Test 74](#_Toc121328880)

[3.3.77 Select MCQs Option 74](#_Toc121328881)

[3.3.78 Deselect MCQs Option 75](#_Toc121328882)

[3.3.79 Submit Test 75](#_Toc121328883)

[3.3.80 View Results 75](#_Toc121328884)

[3.3.81 View Test standing 76](#_Toc121328885)

[3.3.82 View Detailed results 76](#_Toc121328886)

[3.3.83 Enter Query 77](#_Toc121328887)

[3.3.84 Show university rankings 77](#_Toc121328888)

[3.3.85 Show top careers 77](#_Toc121328889)

[3.3.86 Submit feedback 78](#_Toc121328890)

[3.3.87 Select institution 78](#_Toc121328891)

[3.3.88 View Past Papers 79](#_Toc121328892)

[3.3.89 Select Year 79](#_Toc121328893)

[3.3.90 Start Mock test 79](#_Toc121328894)

[3.3.91 View Mock Test result 80](#_Toc121328895)

[3.3.92 Submit test 80](#_Toc121328896)

[3.3.93 End test 80](#_Toc121328897)

[3.3.94 Show Results 81](#_Toc121328898)

[3.3.95 View discussion form 81](#_Toc121328899)

[3.3.96 Enter question 82](#_Toc121328900)

[3.3.97 Submit answer 82](#_Toc121328901)

[3.3.98 Add Video (learning material) 82](#_Toc121328902)

[3.3.99 Add notes (learning material) 83](#_Toc121328903)

[3.3.100 View video 83](#_Toc121328904)

[3.3.101 View notes 84](#_Toc121328905)

[3.4 Non-Functional Requirements 84](#_Toc121328906)

[3.4.1 Reliability 84](#_Toc121328907)

[3.4.2 Usability 84](#_Toc121328908)

[3.4.3 Performance 84](#_Toc121328909)

[3.4.4 Security 84](#_Toc121328910)

[3.5 External Interface Requirements 85](#_Toc121328911)

[3.5.1 Sample user Interfaces 85](#_Toc121328912)

[3.5.2 Software interfaces 89](#_Toc121328913)

[3.5.3 Communications interfaces 89](#_Toc121328914)

[4 Design and Architecture 90](#_Toc121328915)

[4.1 Architectural Design 90](#_Toc121328920)

[4.2 Design Models 90](#_Toc121328921)

[4.2.1 Class Diagram 91](#_Toc121328922)

[4.2.2 Activity Diagrams 93](#_Toc121328923)

[4.2.2.1 Sign-in 93](#_Toc121328924)

[4.2.2.2 Sign-up 94](#_Toc121328925)

[4.2.2.3 View/ Search Job Listing 95](#_Toc121328926)

[4.2.2.4 Logout 96](#_Toc121328927)

[4.2.2.5 Chatbot 97](#_Toc121328928)

[4.2.2.6 Join call 98](#_Toc121328929)

[4.2.3 Sequence Diagrams 99](#_Toc121328930)

[4.3 Data Design 103](#_Toc121328931)

[4.3.1 Data Dictionary 112](#_Toc121328932)

[4.4 Human Interface Design 113](#_Toc121328933)

[4.4.1 Screen Images 113](#_Toc121328934)

[5 Implementation 117](#_Toc121328935)

[5.1 Algorithm 117](#_Toc121328937)

[5.2 External APIs/SDKs 119](#_Toc121328938)

[5.3 User Interface 119](#_Toc121328939)

[5.3.1 Login Screen 120](#_Toc121328940)

[5.3.2 Virtual Session 121](#_Toc121328941)

[5.3.3 Aptitude test 122](#_Toc121328942)

[5.3.4 Chatbot 123](#_Toc121328943)

[5.3.5 One-Stop admission check 125](#_Toc121328944)

[5.4 Deployment 125](#_Toc121328945)

[6 Testing and Evaluation 125](#_Toc121328946)

[6.1 Unit Testing 125](#_Toc121328949)

[6.2 Functional Testing 140](#_Toc121328951)

[6.3 Integration Testing 149](#_Toc121328952)

[7 Conclusion and Future Work 151](#_Toc121328953)

[7.1 Conclusion 151](#_Toc121328955)

[7.2 Future Work 151](#_Toc121328956)

[8 References 152](#_Toc121328957)

# 

# Introduction

This section deals with the rationale and motivation behind developing the proposed system i.e., shaping up high school students career paths by providing with all the help and counseling as discussed in detail in upcoming section. Factors that make our proposed system superior over other existing systems and its limitations and constraints are also discussed below.

## Vision Statement

For students who are doubtful of their future career our system will counsel them selecting their right careers by taking an aptitude test from all the candidates helping them recognize their strength and weaknesses to clarify them in choosing their right path. Once the students decide their path our platform will guide them to their preferred university; which includes helping via admission guidelines, mock tests and all the necessary preparation a student needs in getting into a university. After the student graduates, we will also help him get his preferred job. As the proposed system redefines the career counseling by helping a student from the start of his high school career to help in finding a job for him in future all under one platform; we conclude that Career Development Buddy (CDB) is unique and will help the young generation in guiding and developing their career a lot.

*.*

## Related System Analysis/Literature Review

**Table 1:** Related System Analysis with proposed project solution

|  |  |  |
| --- | --- | --- |
| **Application Name** | **Weakness** | **Proposed Project Solution** |
| CareerGuide | It only answers three questions:   * What career should I choose as a student? * Where do I study to achieve my career goals? * What is the scope of my career & Education? | * Proposed project further helps the user to help him apply for a specific institute from a single platform.      * Receive all the guidelines needed to aid him with his admission in the concerned field of a specific institute. |
| Seekex | * It doesn’t help the user with guidelines to apply for a particular institute. * For test preparation there is no collaborative learning. | * Helps user to apply for any institute by providing him with all necessary details and guidelines. * Enables collaborative learning of students by discussion forms. |

## Project Deliverables

* Scope Document.
* Software Requirement Specification Document.
* Software Design Specification Document.
* Software Implementation.
* Final Report (along with testing).

## System Limitations/Constraints

*.*

*LI-1: As no viable Pakistani universities dataset was found so recommendation model was modelled accordingly to the dataset made from scratch on limited entries to get recommendations.*

*L1-2: Chat-bot may not respond to any random query posed by user due to unavailability to be trained on large data and requires a stable internet connection in order to avoid any abnormal behavior.*

## Tools and Technologies

**Table 2:** Tools and Technologies for Proposed Project

|  |  |  |  |
| --- | --- | --- | --- |
| **Tools**  **And**  **Technologies** | **Tools** | **Version** | **Rationale** |
| CSS | CSS3 | Web Styling |
| HTML | HTML5 | Web Markup |
| Express | 4.18.1 | Backend |
| JavaScript | ES6 | Web Development |
| MongoDB | 5.0.13 | DBMS |
| Python | 3.10.9 | ML |
| ReactJS | 16.0.0 | Web Development |
| Redux | 4.2.0 | State Management |
| Node | 18.12.0 | Backend Development |
| Visual Studio Code | 1.73 | IDE |
| MS Project | 2020 | Gant chart |
| Pencil | 2020 | Mock-ups |
| **Technology** | **Version** | **Rationale** |
| Open AI | CX | Chatbot |
|  | TensorFlow | 2.11.0 | ML/AI |

## Relevance to Course Modules

* **Database Management:**

System uses non-relational database i.e., MongoDB to store and access its data.

* **Object oriented programming:**

System follows object-oriented concepts of separate classes and reusable components.

* **Machine learning:**

System uses machine learning algorithm to develop university recommendation system.

* **Responsive application:**

System is responsive for different platforms and screen sizes.

# Problem Definition

Problems that our proposed system intend to solve and its objectives are described below in detail.



# Problem Statement

Purpose of Career Development Buddy is to help the user clearing his ambiguity regarding his selection of future path by providing him with aptitude test to determine his interests, getting university recommendations, having virtual sessions with career counselors, getting prepared for admission tests of a specific university by getting all the necessary material for preparation and taking mock tests according to the specific university test patterns and criteria to get an early idea of his standing, accessing a one-stop admission criteria by entering all the required information regarding his academics and then calculating his aggregate in order to know which universities and their particular fields he can apply for. Apart from it, there will be a job portal to help students find a job that suits them as per their needs from a list of job postings from different companies.

# Problem Solution

Our Problem Solution is divided into three phases.

(Phase I) Upon registration of a student in to our platform, student will be asked to give an aptitude test from helping him to clarify his future choices.

(Phase II) After making up his mind regarding field and university, Students will be prepared to match their university requirements for getting admission with help of mock tests and other guiding material and also get university recommendations according to his academic record.

(Phase III) Later upon graduation, we would show them different companies with their listing that match their CV in order to help them to get a job.

*.*

# Objectives of the Proposed System

Some of the objectives are:

*BO-1: Ease and Time convenience*

*BO-2: Providing students a platform where all university info is one click away.*

*BO-3: One Stop Admission-test meeting criteria check Solution*

*BO-4: University Suggestions according to user’s interests.*

*BO-5: Helping to find a suitable Job and connect applicant and employer virtually.*

*.*

# Scope

The proposed system helps students with vague mind for their future career taking aptitude tests to determine their strengths and weakness allowing helping him to explore himself, getting universities recommendations based on his interests, getting prepared for admission tests of a specific university be getting all the necessary material for preparation and taking mock tests according for a specific university to get an early idea of his standing, accessing a one-stop admission check criteria by entering all the required information regarding his academics and calculating aggregate in order to know which universities he can apply for and get all the guidelines needed for helping them get admission. Some other features include virtual sessions. An AI chat-bot will be available to facilitate the user regarding any issue or query that he might possess. System also provides the user to find a suitable job for him via job portal. Companies logged in as user can add job listing on the portal and have virtual sessions (Audio/Video call with applicants) whereas undergraduate students logged in as users can view all the job listings and apply for them to find a suitable job.

# Modules

Modules along with its functionality are listed below:

# Module 1: User Profiling

This module will help user in managing his profile like creating his new account on system, logging in, logging out and updating his account details.

M1-UC1: Sign-up

M1-UC2: Sign in.

M1-UC3: Forget password.

M1-UC4: Logout.

M1-UC5: Update account info.

# Module 2: Virtual Sessions

This module helps user to place audio and video virtual sessions with consultants.

M2-UC1: Place Audio Call

M2-UC2: Receive Audio Call

M2-UC3: Place Video Call

M2-UC4: Receive Video Call

M2-UC5: Call Logs

# Module 3: Aptitude test

This module allows user to take aptitude test in order to recognize his strengths and weaknesses to choose his career path.

M3-UC1: Generate test.

M3-UC2: View test results.

M3-UC3: Generate Result Card.

M3-UC4: View test standings.

# Module 4: Institute recommendation

This module will help the user to choose a university by providing him all the details at a single place and getting university recommendation according to his selected preferences.

M4-UC1: View University

M4-UC2: Compare University.

M4-UC3: Manage preferences.

M4-UC4: Get university recommendation.

M4-UC5: FAQ’s

# Module 5: Entry Test Preparation

This Module helps the student in preparation for the entry test via providing him with multiple materials, a collaborative discussion form for students and universities past admission tests.

M5-UC1: Mock Test.

M5-UC2: Learning Material.

M5-UC3: Discussion forms.

M5-UC4: Past Papers

# Module 6: AI chat-bot

This Module assists the user by responding to his posed queries.

M6-UC1: Ask Query.

M6-UC2: Get University Rankings.

M6-UC3: Top Careers.

M6-UC4: User Feedback.

# Module 7: One-Stop admission meeting criteria check

This module will help the user to calculate his aggregate and check all the universities that he can apply to, further helping him to get all the admission guidelines all from a single place.

M7-UC1: Add Academic Record.

M7-UC2: Update Academic Record.

M7-UC3: Calculate Aggregate.

M7-UC4: Display applicable Institutes.

M7-UC5: Get Admission Guidelines.

M7-UC6: Dashboard

# Module 8: Jobs portal

This module will help the user to find a suitable job for him and for companies to easily approach and find employee as per their needs.

M8-UC1: Display job listings.

M8-UC2: Filter Job Listing

M8-UC3: Upload Attachments

M8-UC4: Job Apply

M8-UC5: Manage Job Listing.

M8-UC6: View Applicant Details.

# Requirement Analysis

All the requirements that we gathered i.e., functional, non-functional, external along with the method used in gathering these requirements are discussed below in detail.

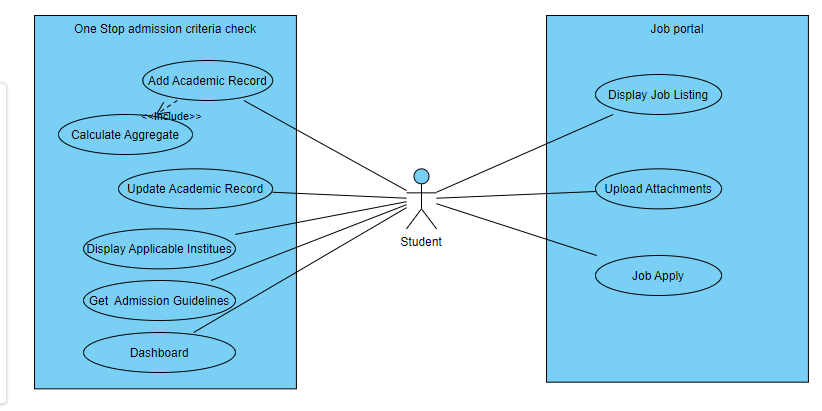


# User classes and characteristics

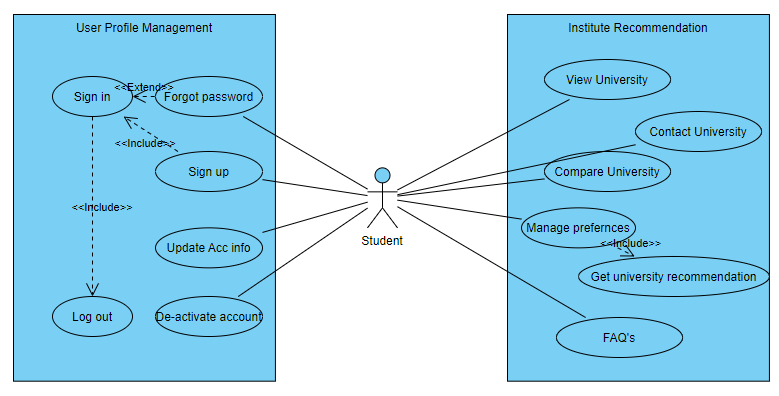
|  |  |
| --- | --- |
| **User class** | **Description** |
| **Student** | * They are the primary users of the app. * Students who passed their intermediate (F.SC) and seeking for guidance or needs preparation help to apply in universities or find a job will be using the app. |
| **Employer** | * Employer seeking a suitable person for their job shall register themselves into the system and find themselves an employee for their job listing. * Employer shall conduct virtual sessions with applicants. |
| **Admin** | * Admin will be responsible to manage users, entry test material management, and user feedback, manage job listings. |

# Requirement Identifying Technique

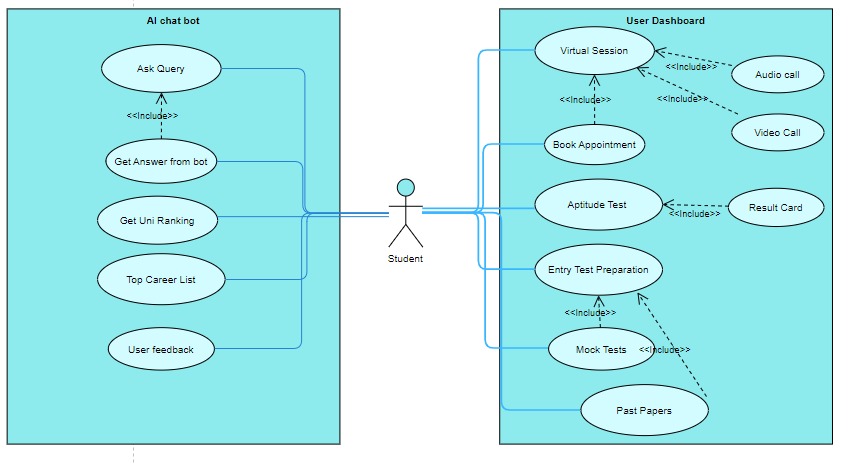
# Use-Case Diagram:



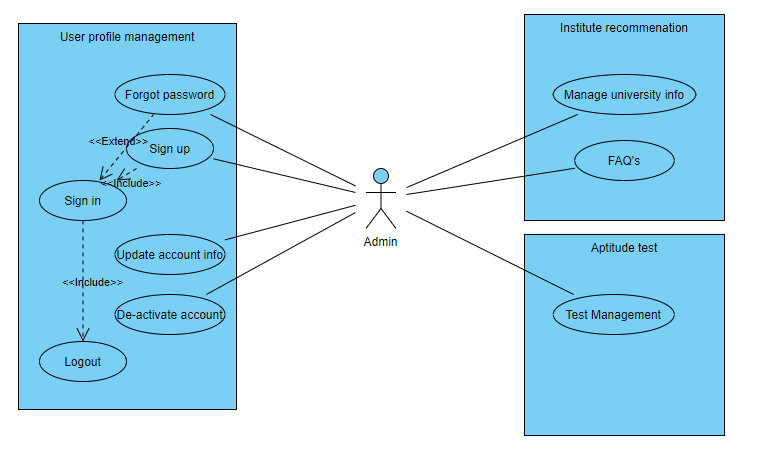
**Figure 1: Use-Case Diagram for Student (1/3)**



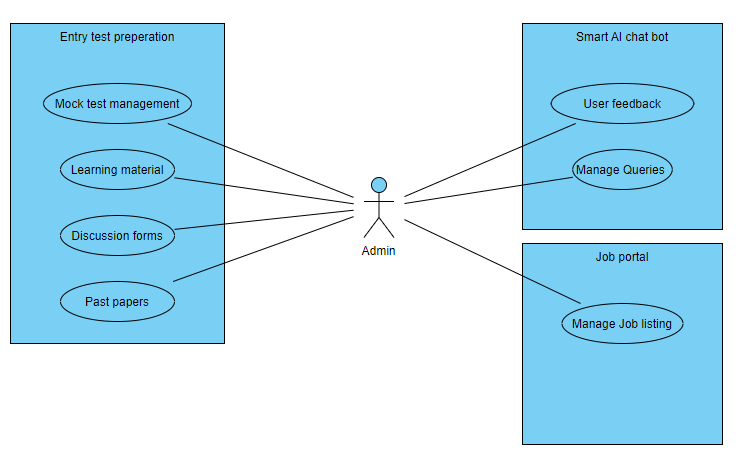
**Figure 2: Use-Case Diagram for Student (2/3)**



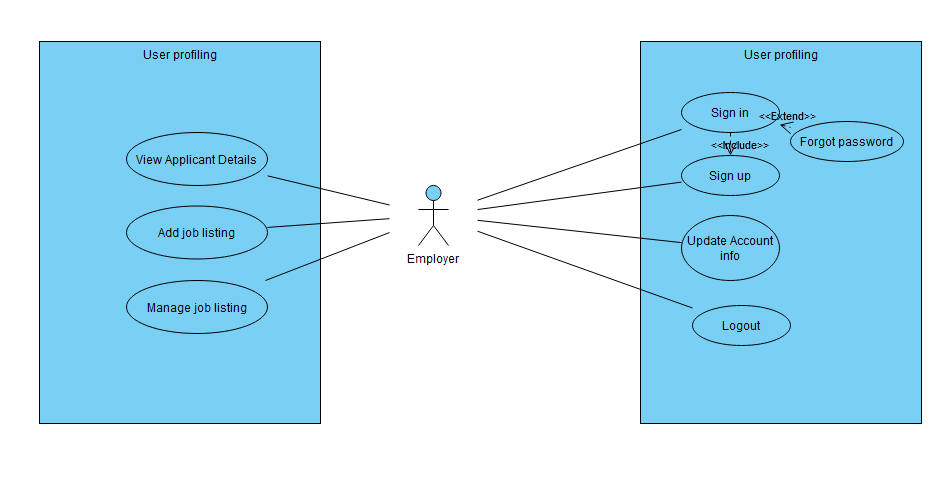
**Figure 3: Use-Case Diagram for Student (3/3)**



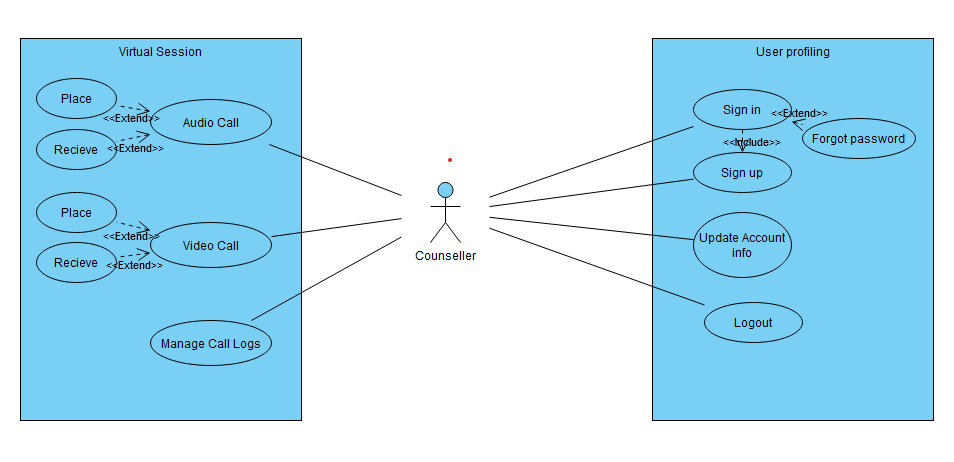
**Figure 4: Use-Case Diagram for Admin (1/2)**



**Figure 5: Use-Case Diagram for Admin (2/2)**



**Figure 6: Use-Case Diagram for Employer (1/2)**



**Figure 7: Use-Case Diagram for Employer (2/2)**

# Tabular Use-case

**Module 1: User profiling**

# Use-Case 1.1 - Sign up

**Table 4:** UC-1 Sign-up Account

|  |  |
| --- | --- |
| **Use Case ID:** | UC-1.1 |
| **Use Case Name:** | Sign-up |
| **Actors:** | Student, Counselor, Employer, Admin |
| **Description:** | User will enter his account information resulting in his new account to be created. |
| **Trigger:** | User clicks on the Sign-up button. |
| **Preconditions:** | N/A |
| **Post conditions:** | POST-1. Student’s new account is created. |
| **Normal Flow:** | 1. User clicks on the Sign-up button displayed on main screen.  2. Sign-up Screen appears.  3. User enters his first name in the required place.  4. User enters his last name in the required place.  5. User enters his email in the required place.  6. User clicks on verify email.  7. Verification link is sent to student email.  8. User opens his email and clicks on verification link.  9. User enters his password in the required place.  10. User confirms his password by typing it again.  11. User uploads his profile picture.  12. User enters his company name (if any).  12. User enters his contact number.  13. User clicks on Sign-up button. |
| **Alternative Flows:** | N/A |
| **Exceptions:** | 1a. User enters Invalid/non-existing email.  1b. User enters Invalid password format.  1c. Confirm password field doesn’t match with previously entered password. |
| **Business Rules** | BR-1. An email can only be linked to a single account. |
| **Assumptions:** | * User has a active internet connection. * User has a Google account created. |

# Use-Case 1.2 - Sign in

**Table 4.2:** UC-2 Account Log in

|  |  |
| --- | --- |
| **Use Case ID:** | UC-1.2 |
| **Use Case Name:** | Sign-in |
| **Actors:** | Student, Counselor, Employer, Admin |
| **Description:** | User will enter his email and password to log into the system. |
| **Trigger:** | User clicks on the Sign-in button. |
| **Preconditions:** | PRE-1. User account is already created (sign-up). |
| **Post conditions:** | POST-1. User is logged into the system. |
| **Normal Flow:** | 1. User clicks on the Sign-in button displayed on main screen.  2. Sign-in Screen appears.  3. User enters valid email in the required place.  4. User enters his valid password in the required place.  5. User clicks on Sign-in. |
| **Alternative Flows:** | 4a. If password entered is incorrect, text box turns red.  4b. User enters his password again.  4c. Control transferred to step 5 of normal flow. |
| **Exceptions:** | 3a. User enters invalid email.  4a. User enters invalid password. |
| **Business Rules** | N/A |
| **Assumptions:** | * User has his account created previously. |

# Use-Case 1.3 – Forgot Password

**Table 4.3:** UC-3 Forget password

|  |  |
| --- | --- |
| **Use Case ID:** | UC-1.3 |
| **Use Case Name:** | Forget password |
| **Actors:** | Student, Counselor, Employer, Admin |
| **Description:** | User entered email will receive a verification link which upon clicking will help him to reset his account password. |
| **Trigger:** | User clicks on the Forgot password button. |
| **Preconditions:** | PRE-1. User account is already created (sign-up).  PRE-2. User enters valid email. |
| **Post conditions:** | POST-1. User’s account password is reset. |
| **Normal Flow:** | 1. User clicks on the Forgot password button displayed on Sign in screen.  2. User enter his email and click on continue.  3. A verification link is sent to the email.  4. Upon successful verification, user is prompted to enter his new password.  5. User enters his new password.  6. User enters his new password again in the confirm password box.  7. User clicks on Enter. |
| **Alternative Flows:** | N/A |
| **Exceptions:** | 1. User enters invalid email.  2. User has not created his account previously. |
| **Business Rules** | BR-1. User can only change his password once per week. |
| **Assumptions:** | * User has his account created previously. |

# Use-Case 1.4 – Log out

**Table 4.4:** UC-4 Logout

|  |  |
| --- | --- |
| **Use Case ID:** | UC-1.4 |
| **Use Case Name:** | Log out |
| **Actors:** | Student, Counselor, Employer, Admin |
| **Description:** | User will be logged out of the system. |
| **Trigger:** | User clicks on the Log out button. |
| **Preconditions:** | PRE-1. User is already signed in. |
| **Post conditions:** | POST-1. User will be logged out of the system. |
| **Normal Flow:** | 1. User clicks his profile icon at the upper right corner of the screen.  2. User clicks on Log out form the dropdown menu.  3. A confirmation message appears confirming user action of logging out of the system.  4. User clicks on confirm. |
| **Alternative Flows:** | N/A |
| **Exceptions:** | 1. User is not logged in . |
| **Business Rules** | N/A |
| **Assumptions:** | * User is logged in. |

# Use-Case 1.5 – Update Account Info

**Table 4.5:** UC-5 Update account info

|  |  |
| --- | --- |
| **Use Case ID:** | UC-1.5 |
| **Use Case Name:** | Update Account Info |
| **Actors:** | Student, Counselor, Employer, Admin |
| **Description:** | User will enter his account update information to update his account with new information. |
| **Trigger:** | User clicks on Update Account Info from profile Menu. |
| **Preconditions:** | PRE-1. User is already signed in. |
| **Post conditions:** | POST-1. User account will be update with new information. |
| **Normal Flow:** | 1. User clicks his profile icon at the upper right corner of the screen.  2. User clicks on Update Account Info form the dropdown menu.  3. Account Info Screen Appears.  4. User chooses and updates what he wants to edit/update from the options displayed on the screen i.e. first name, last name, email, password, profile picture or contact number.    5. User clicks on update. |
| **Alternative Flows:** | 4a. If User updates his email, verification link will be sent to that email.  4b. User verifies the email to update it.  4c. If User updates his password, user will be prompted to enter his old password.  4d. User enters his ld password.  4e. User enters his new password.  4f. User re-enter his new password.  4g. User click on confirm. |
| **Exceptions:** | 1. User is not logged in. |
| **Business Rules** | N/A |
| **Assumptions:** | * User is logged in. |

**Module 2: Virtual Sessions**

# Use-Case 2.1 – Place Audio Call

**Table 4.6:** UC-1 Initiate Audio Call

|  |  |
| --- | --- |
| **Use Case ID:** | 2.1 |
| **Use Case Name:** | Place Audio Call |
| **Actors:** | Student, Counselor |
| **Description:** | This use case shall allow the student, counselor to make voice calls |
| **Trigger:** | Clicking voice call button |
| **Preconditions:** | PRE-1. User have a registered account  PRE-2. User have an appointment from the counselor  PRE-3. User clicks voice call option from main menu |
| **Post conditions:** | POST-1. User makes a call |
| **Normal Flow:** | 1. User selects call option from menu. 2. User clicks on voice call option. 3. System connects call to recipient. 4. Voice call screen appears. |
| **Alternative Flows:** | N/A |
| **Exceptions:** | N/A |
| **Business Rules** | N/A |
| **Assumptions:** | 1. Both the users have good internet connection |

# Use-Case 2.2 – Receive Audio call

**Table 4.7:** UC-2 Receive Audio Call

|  |  |
| --- | --- |
| **Use Case ID:** | 2.2 |
| **Use Case Name:** | Receive Voice Call |
| **Actors:** | Student, Counselor |
| **Description:** | This use case shall allow the student, counselor to receive voice calls |
| **Trigger:** | Clicking answer/decline button |
| **Preconditions:** | PRE-1. User have a registered account  PRE-1. User have an appointment from the counselor |
| **Post conditions:** | POST-1. User receives a call |
| **Normal Flow:** | After a voice call is made   1. Voice call screen appears. 2. Recipient slides right to attend a call. 3. Recipient clicks “Message” button to send message automatically declining the call. |
| **Alternative Flows:** | In step 2 of normal flow if recipient wants to decline the call   1. Recipient slides left to decline the call. |
| **Exceptions:** | N\A |
| **Business Rules** | N\A |
| **Assumptions:** | 1. Both the users have good internet connection |

# Use-Case 2.3 – Place Video Call

**Table 4.8:** UC-3 Initiate Video Call

|  |  |
| --- | --- |
| **Use Case ID:** | 2.3 |
| **Use Case Name:** | Place Video Call |
| **Actors:** | Student, Counselor |
| **Description:** | This use case shall allow the student, counselor to make Video calls |
| **Trigger:** | Clicking video call button |
| **Preconditions:** | PRE-1. User have a registered account  PRE-2. User have an appointment from the counselor  PRE-3. User clicks video call option from main menu |
| **Post conditions:** | POST-1. User makes a call |
| **Normal Flow:** | 1. User selects call option from menu. 2. User clicks on Video call option. 3. System connects call to recipient. 4. Video call screen appears. |
| **Alternative Flows:** | N\A |
| **Exceptions:** | N\A |
| **Business Rules** | N\A |
| **Assumptions:** | 1. Both the users have good internet connection |

# Use-Case 2.4 – Receive Audio Call

**Table 4.9:** UC-4 Receive Video Call

|  |  |
| --- | --- |
| **Use Case ID:** | 2.4 |
| **Use Case Name:** | Receive Video Call |
| **Actors:** | Student, Counselor |
| **Description:** | This use case shall allow the student, counselor to receive video calls |
| **Trigger:** | Clicking answer/decline button |
| **Preconditions:** | 2.4.1 User have a registered account  2.4.2 User have an appointment from the counselor |
| **Post conditions:** | 2.4.1 User receives a call |
| **Normal Flow:** | After a video call is made   1. Video call screen appears. 2. Recipient slides right to attend a call. 3. Recipient clicks “Message” button to send message automatically declining the call. |
| **Alternative Flows:** | In step 2 of normal flow if recipient wants to decline the call   1. Recipient slides left to decline the call. |
| **Exceptions:** | None |
| **Business Rules** | None |
| **Assumptions:** | 1. Both the users have good internet connection |

# Use-Case 2.5 – Call Logs

**Table 4.10:** UC-5 Receive Video Call

|  |  |
| --- | --- |
| **Use Case ID:** | 2.5 |
| **Use Case Name:** | Call logs. |
| **Actors:** | Student, Counselor, Admin |
| **Description:** | This use case shall allow the student, counselor to view their call logs. |
| **Trigger:** | Clicking call logs button. |
| **Preconditions:** | 2.5.1 User has a registered account. |
| **Post conditions:** | 2.5.1 Call logs screen will be displayed on screen. |
| **Normal Flow:** | 1. User clicks on dashboard. 2. User clicks on call logs button. 3. Call logs screen appears. 4. User view call logs of incoming/outgoing voice and video calls, |
| **Alternative Flows:** | None |
| **Exceptions:** | None |
| **Business Rules** | None |
| **Assumptions:** | 1. User has received/made some calls. |

**Module 3: Aptitude Test**

# Use-Case 3.1 – Generate Test

**Table 4.11:** UC-1 Generate Aptitude test

|  |  |
| --- | --- |
| **Use Case ID:** | UC-3.1 |
| **Use Case Name:** | Generate Test |
| **Actors:** | Student |
| **Description:** | User generates test to start |
| **Trigger:** | User clicks on Generate Test button from aptitude test screen. |
| **Preconditions:** | PRE-1. User is already signed in. |
| **Post conditions:** | POST-1. MCQs based test screen will appear. |
| **Normal Flow:** | 1. User navigates to aptitude test screen. 2. User clicks on generate test button. 3. User selects type of test. 4. User clicks on finish test. |
| **Alternative Flows:** | N/A  . |
| **Exceptions:** | N/A |
| **Business Rules** | N/A |
| **Assumptions:** | 1. User is logged in. |

# Use-Case 3.2 – View Test Results

**Table 4.12:** UC-2 View test results

|  |  |
| --- | --- |
| **Use Case ID:** | UC-3.2 |
| **Use Case Name:** | View Test Results |
| **Actors:** | Student |
| **Description:** | User view test results after finishing test. |
| **Trigger:** | User clicks on View results button from main test screen. |
| **Preconditions:** | PRE-1. User is already signed in.  PRE-2. User has attempted the test. |
| **Post conditions:** | POST-1. Results page screen will appear. |
| **Normal Flow:** | 1. User finishes test and navigates to test main screen. 2. User clicks on view test results button. 3. User clicks on Test score button. 4. User clicks on Test standing button. |
| **Alternative Flows:** | N/A  . |
| **Exceptions:** | N/A |
| **Business Rules** | N/A |
| **Assumptions:** | 1. User is logged in. 2. User has attempted test. |

# Use-Case 3.3 – Generate Result Card

**Table 4.13:** UC-3 Generate result card

|  |  |
| --- | --- |
| **Use Case ID:** | UC-3.3 |
| **Use Case Name:** | Generate result card. |
| **Actors:** | Student |
| **Description:** | User generates result card after viewing test results. |
| **Trigger:** | User clicks on generate result card button from test result screen. |
| **Preconditions:** | PRE-1. User is already signed in.  PRE-2. User has attempted the test. |
| **Post conditions:** | POST-1. Results card will download in pdf. |
| **Normal Flow:** | 1. User navigates to test result page. 2. User clicks on view result button. 3. User clicks on generate result card button available on result page. |
| **Alternative Flows:** | 1. User only view result and doesn’t generate result card. |
| **Exceptions:** | N/A |
| **Business Rules** | N/A |
| **Assumptions:** | 1. User is logged in. 2. User has attempted test. |

**Module 4: Institute Recommendation**

# Use-Case 4.1 – View University

**Table 4.14:** UC-1 View University

|  |  |
| --- | --- |
| **Use Case ID:** | UC-4.1 |
| **Use Case Name:** | View University |
| **Actors:** | Student |
| **Description:** | User gets to view compressive details about a university. |
| **Trigger:** | User clicks on View University button from Institute/Field recommendation screen. |
| **Preconditions:** | PRE-1. User is already signed in. |
| **Post conditions:** | POST-1. Comprehensive details of university will be displayed. |
| **Normal Flow:** | 1. User clicks his Institute/Field recommendation button after logging into the system.  2. User clicks on View University button form the dropdown menu .  3. User type university name in the search bar.  4. Universities matching the name appears.  5. User clicks on the university.  6. User is directed to the selected university page. |
| **Alternative Flows:** | N/A  . |
| **Exceptions:** | 1. User entered invalid university name. |
| **Business Rules** | N/A |
| **Assumptions:** | * User is logged in. |

# Use-Case 4.2 – Manage Preferences

**Table 4.15:** UC-2 Manage preferences

|  |  |
| --- | --- |
| **Use Case ID:** | UC-4.2 |
| **Use Case Name:** | Manage Preferences |
| **Actors:** | Student |
| **Description:** | User will enter his preferences regarding his preferred field, location which will later help him get university recommendation. . |
| **Trigger:** | User clicks on Manage preferences button from Institute/Field recommendation screen. |
| **Preconditions:** | PRE-1. User is already signed in. |
| **Post conditions:** | POST-1. User entered Preferences will be added or updated into the system. |
| **Normal Flow:** | 1. User clicks his Institute/Field recommendation button after logging into the system.  2. User clicks on Manage Preferences button.  3. User is prompted to select from add or update preferences.  4. User made changes are saved.  5. User clicks on Enter to proceed. |
| **Alternative Flows:** | 3.1. Add preferences  a. User selects add preference.  b. User is prompted to select his preferred field from the given list.  c. User is prompted to select his preferred location i.e city from the given list.  3.2 Update Preferences  a. User selects update preference.  b. User is prompted to select his preferred field from the given list.  c. User is prompted to select his preferred location i.e city from the given list.  d. Control transferred to step 5 of normal flow. |
| **Exceptions:** | 1. User preferred location is not found. 2. In case update preferences, user has not added his preferences before. 3. User is not logged in. |
| **Business Rules** | N/A |
| **Assumptions:** | * User is logged in. |

# Use-Case 4.3 – Get University recommendation

**Table 4.16:** UC-3 Get university recommendation

|  |  |
| --- | --- |
| **Use Case ID:** | UC-4.3 |
| **Use Case Name:** | Get university Recommendation |
| **Actors:** | Student |
| **Description:** | Users will receive university recommendation according to his preferences (location, budget, program preference). |
| **Trigger:** | User clicks on Get University Recommendation button from Institute/Field recommendation screen. |
| **Preconditions:** | PRE-1. User is already signed in.  PRE-2. User has already entered his preferences (location, budget, program preference). |
| **Post conditions:** | POST-1. User will be displayed with his university recommendation. |
| **Normal Flow:** | 1. User clicks Institute/Field recommendation button after logging into the system.  2. User clicks on Get University Recommendation button.  3. User will be displayed with a list of universities recommended for him. |
| **Alternative Flows:** | N/A  . |
| **Exceptions:** | * User is not logged in. |
| **Business Rules** | N/A |
| **Assumptions:** | * User is logged in. * User has entered his preferences. |

# Use-Case 4.4 – Compare University

**Table 4.17:** UC-4 Compare University

|  |  |
| --- | --- |
| **Use Case ID:** | UC-4.4 |
| **Use Case Name:** | Compare University |
| **Actors:** | Student |
| **Description:** | User will be able to compare two universities. |
| **Trigger:** | User clicks on Compare University button from Institute/Field recommendation screen. |
| **Preconditions:** | PRE-1. User is already signed in. |
| **Post conditions:** | POST-1. User will be displayed with comparison of two universities. |
| **Normal Flow:** | 1. User clicks Institute/Field recommendation button after logging into the system.  2. User clicks on Compare University button.  3. User is displayed with list of universities.  3. User selects first university name from the given list of universities.  4. User selects second university name from the given list of universities that he wants to compare with.  4. User is prompted to select the type of comparison i.e compare courses or compare statistics.  5. Result of comparison is displayed to the user as per his selection.  . |
| **Alternative Flows:** | 4.1. Compare Courses  a. User select compare courses option.  b. User is prompted to select the program of the particular universities of which he wants the courses to be compared.  c. Complete list of courses offered by the university of the concerned program is displayed.  .  4.2. Compare Statistics  a. User selects compare statistics option.  b. User is displayed with a complete comparison of statistics of both universities |
| **Exceptions:** | * User is not logged in. |
| **Business Rules** | N/A |
| **Assumptions:** | * User is logged in. |

# Use-Case 4.5 – Contact University

**Table 4.18:** UC-5 Contact University

|  |  |
| --- | --- |
| **Use Case ID:** | UC-4.5 |
| **Use Case Name:** | Contact University |
| **Actors:** | Student |
| **Description:** | User will be able to contact the university. |
| **Trigger:** | User clicks on Get Field Recommendation button from Institute/Field recommendation screen. |
| **Preconditions:** | PRE-1. User is already signed in. |
| **Post conditions:** | POST-1. User will contact the university via phone call or email. |
| **Normal Flow:** | 1. User clicks Institute/Field recommendation button after logging into the system.  2. User clicks on Contact University button.  3. User is displayed with list of universities.  3. User selects university name from the given list of universities.  4. User selects branch of university (if any) that he wants to make contact with.  5. User will be prompted to select contact method among the available options of Contact number or email.  6. User selects Contact number.  a. Contact number of that university is displayed to the user.  b. User clicks on dial number icon.  c. Call is made to the concerned contact number.  7. User selects email.  a. Email of the university is displayed to the user.  b. User types the email message.  c. User clicks on send to send the email. |
| **Alternative Flows:** | N/A  . |
| **Exceptions:** | * User is not logged in. * User has not added his contact number in his account. |
| **Business Rules** | N/A |
| **Assumptions:** | * User is logged in. * User has added his contact number to his account. |

# Use-Case 4.6 – FAQ’s

**Table 4.19:** UC-6 FAQ

|  |  |
| --- | --- |
| **Use Case ID:** | UC-4.6 |
| **Use Case Name:** | FAQ’s |
| **Actors:** | Student, Admin |
| **Description:** | User will ask a question for which he seek an answer or view frequently asked questions related to a particular institute. |
| **Trigger:** | User clicks on FAQs from Institute/Field recommendation screen. |
| **Preconditions:** | PRE-1. User is already signed in. |
| **Post conditions:** | POST-1. If user selects on Ask question, he would be later provided with an answer to his query regarding a particular institute.  POST-2. If user selects view FAQs, a form containing all the FAQ’s will be displayed to the user. |
| **Normal Flow:** | 1. User clicks Institute/Field recommendation button after logging into the system.  2. User clicks on FAQ’s button.  3. User is prompted to select the institute from the given list of institutes about which he seeks query.  3. User is prompted to select if he wants to ask a question or view FAQ’s form.  3. User selects Ask a Question.  a. A new screen is displayed with text box asking user to enter his question.  b. User clicks on Ask button.  c. Answer to the user question will be displayed to the user later when another user answers it.  4. User selects branch view FAQs.  a. User is directed to the FAQs screen displaying all the Frequently asked questions by other user. |
| **Alternative Flows:** | N/A  . |
| **Exceptions:** | * User is not logged in. |
| **Business Rules** | N/A |
| **Assumptions:** | * User is logged in. |

**Module 5: Entry Test Preparation**

# Use-Case 5.1- Mock Test

**Table 4.20:** UC-1 Mock test

|  |  |
| --- | --- |
| **Use Case ID:** | UC-5.1 |
| **Use Case Name:** | Mock Test |
| **Actors:** | Student |
| **Description:** | User gives mock test of specific institutions. |
| **Trigger:** | User clicks on Mock Test button in Entry test preparation section. |
| **Preconditions:** | PRE-1. User is already signed in. |
| **Post conditions:** | POST-1. Mock test MCQs for selected institution will appear on screen. |
| **Normal Flow:** | 1. User navigates to Entry test preparation section. 2. User clicks on Mock test button. 3. User selects institution for mock test mcqs. 4. User view/browse Mcq`s of selected institution for mock test. |
| **Alternative Flows:** | N/A |
| **Exceptions:** | N/A |
| **Business Rules** | N/A |
| **Assumptions:** | 1. User is logged in. |

# Use-Case 5.2 – Learning Material

**Table 4.21:** UC-2 Learning material

|  |  |
| --- | --- |
| **Use Case ID:** | UC-5.2 |
| **Use Case Name:** | Learning Material |
| **Actors:** | Student, Admin |
| **Description:** | User browse learning materials of specific available courses. |
| **Trigger:** | User clicks on Learning Material button in Entry test preparation section. |
| **Preconditions:** | PRE-1. User is already signed in. |
| **Post conditions:** | POST-1. User will be able to access all pdf books and video lectures of available courses. |
| **Normal Flow:** | 1. User navigates to Entry test preparation section. 2. User clicks on Learning Material Button. 3. User selects course. 4. User selects material type i.e. pdf /video lectures. |
| **Alternative Flows:** | N/A |
| **Exceptions:** | N/A |
| **Business Rules** | N/A |
| **Assumptions:** | 1. User is logged in. |

# Use-Case 5.3-Discussion Form

**Table 4.22:** UC-3 Discussion form

|  |  |
| --- | --- |
| **Use Case ID:** | UC-5.3 |
| **Use Case Name:** | Discussion Forum |
| **Actors:** | Student, Admin |
| **Description:** | User post query in discussion forum to get answer from community. |
| **Trigger:** | User clicks on Forum button on Entry test preparation screen. |
| **Preconditions:** | PRE-1. User is already signed in. |
| **Post conditions:** | POST-1. User Query/Answer will display to user on screen. |
| **Normal Flow:** | 1. User navigates to entry test preparation section. 2. User clicks on Forum button in entry test section. 3. User type his query. 4. User will click on submit button to post query. |
| **Alternative Flows:** | 1. User posts answer to an open query. |
| **Exceptions:** | N/A |
| **Business Rules** | N/A |
| **Assumptions:** | 1. User is logged in. |

# Use-Case 5.4-Past Papers

**Table 4.23:** UC-4 Past papers

|  |  |
| --- | --- |
| **Use Case ID:** | UC-5.4 |
| **Use Case Name:** | Past Papers |
| **Actors:** | Student, Admin |
| **Description:** | User browses past paper of available institutions for a selected year. |
| **Trigger:** | User clicks on Past papers button on Entry test preparation screen. |
| **Preconditions:** | PRE-1. User is already signed in. |
| **Post conditions:** | POST-1. Available past paper will be displayed to user on screen. |
| **Normal Flow:** | 1. User navigates to entry test preparation section. 2. User clicks on Past paper button in entry test section. 3. User selects institution. 4. User selects test year. |
| **Alternative Flows:** | N/A |
| **Exceptions:** | N/A |
| **Business Rules** | N/A |
| **Assumptions:** | 1. User is logged in. |

**Module 6: AI Chatbot**

# Use-Case 6.1 – Ask Query

**Table 4.24:** UC-1 Ask query

|  |  |
| --- | --- |
| **Use Case ID:** | UC-6.1 |
| **Use Case Name:** | Ask query |
| **Actors:** | Student |
| **Description:** | User start chat for general query to get answer from bot. |
| **Trigger:** | User clicks on message button on AI chat bot screen. |
| **Preconditions:** | PRE-1. User is already signed in. |
| **Post conditions:** | POST-1. Bot Messenger appears on screen.  POST-2. User gets answer |
| **Normal Flow:** | 1. User navigates to entry AI Chat bot. 2. User clicks on message button on chat screen. 3. User type query and enter send. |
| **Alternative Flows:** | 1. User Ends chat |
| **Exceptions:** | N/A |
| **Business Rules** | N/A |
| **Assumptions:** | 1. User is logged in. |

# Use-Case 6.2 – Get University Rankings

**Table 4.25:** UC-2 Get university rankings

|  |  |
| --- | --- |
| **Use Case ID:** | UC-7.2 |
| **Use Case Name:** | Get University Rankings |
| **Actors:** | Student |
| **Description:** | User get university updated ranking from HEC in one touch. |
| **Trigger:** | User clicks on Get Latest Ranking button on AI chat bot screen. |
| **Preconditions:** | PRE-1. User is already signed in. |
| **Post conditions:** | POST-1. University updated ranking list appears |
| **Normal Flow:** | 1. User navigates to entry AI Chat bot. 2. User clicks on Get Latest Ranking button on chat screen. 3. User scroll through ranking list |
| **Alternative Flows:** |  |
| **Exceptions:** | N/A |
| **Business Rules** | N/A |
| **Assumptions:** | 1. User is logged in. |

# Use-Case 6.3 – Top Careers

**Table 4.26:** UC-3 Top careers

|  |  |
| --- | --- |
| **Use Case ID:** | UC-6.3 |
| **Use Case Name:** | Top Careers |
| **Actors:** | Student |
| **Description:** | User gets top 10 career list according to ranking |
| **Trigger:** | User clicks on Top careers icon on ranking list screen. |
| **Preconditions:** | PRE-1. User is already signed in. |
| **Post conditions:** | POST-1. Top career list according to ranking displays. |
| **Normal Flow:** | 1. User navigates to ranking list. 2. User clicks on Top careers icon on ranking list screen. 3. User scroll through careers list |
| **Alternative Flows:** | N/A. |
| **Exceptions:** | N/A |
| **Business Rules** | N/A |
| **Assumptions:** | 1. User is logged in. |

# Use-Case 6.4 – User feedback

**Table 4.27:** UC-4 User feedback

|  |  |
| --- | --- |
| **Use Case ID:** | UC-6.4 |
| **Use Case Name:** | User Feedback |
| **Actors:** | Student, Admin |
| **Description:** | User will give feedback about chat in stars from 1 to 5. |
| **Trigger:** | User will select 1 to 5 stars. |
| **Preconditions:** | PRE-1. User is already signed in.  PRE-2. User has interacted with bot. |
| **Post conditions:** | POST-1. Feedback will appear. |
| **Normal Flow:** | 1. User selects stars from 1 to 5. 2. User adds any optional comment. 3. User submits feedback. |
| **Alternative Flows:** | 1. User doesn’t submit feedback. |
| **Exceptions:** | N/A |
| **Business Rules** | N/A |
| **Assumptions:** | 1. User is logged in. |

**Module 7: One Stop Admission meeting criteria check**

# Use-Case 7.1 – Add Academic Record

**Table 4.28:** UC-1 Add Academic record

|  |  |
| --- | --- |
| **Use Case ID:** | UC-7.1 |
| **Use Case Name:** | Add Academic Details |
| **Actors:** | Student |
| **Description:** | User enters his academic record to be saved by the system. |
| **Trigger:** | User clicks on Add Academic Record from One stop Admission Meeting Criteria Check screen. |
| **Preconditions:** | PRE-1. User is already signed in. |
| **Post conditions:** | POST-1. User’s academic record will be saved by system. |
| **Normal Flow:** | 1. User clicks Add Academic details button after logging into the system.  2. Academic Details Screen is displayed.  3. User selects his secondary school form the given list of schools.  4. User enters his matric marks.  5. User selects his college from the given list of colleges.  6. User enters his Fsc marks.  7. User clicks on Save to save the record. |
| **Alternative Flows:** | N/A  . |
| **Exceptions:** | * User is not logged in. * User has done O/A levels instead of matric/Fsc system. |
| **Business Rules** | N/A |
| **Assumptions:** | * User is logged in. * User has completed his education till FSc. |

# Use-Case 7.2 – Update Academic Record

**Table 4.29:** UC-2 Update Academic record

|  |  |
| --- | --- |
| **Use Case ID:** | UC-7.2 |
| **Use Case Name:** | Update Academic Details |
| **Actors:** | Student |
| **Description:** | User enters his new details to update his existing Academic record. |
| **Trigger:** | User clicks on Update Academic Record from One stop Admission Meeting Criteria Check screen. |
| **Preconditions:** | PRE-1. User is already signed in.  PRE-2. User has added his academic record previously into the system. |
| **Post conditions:** | POST-1. User’s academic record will be updated and saved. |
| **Normal Flow:** | 1. User clicks Update Academic details button after logging into the system.  2. Academic Details Screen is displayed.  3. User selects secondary school box and updates it form the given list of schools.  4. User select Enter matric marks box and updates it.  5. User updates his college from the given list of colleges.  6. User updates his Fsc marks.  7. User clicks on Save to save his changes. |
| **Alternative Flows:** | N/A  . |
| **Exceptions:** | * User is not logged in. * User has done O/A levels instead of matric/FSC system. * User has not added his academic details beforehand. * User has not selected calculate aggregate option before. |
| **Business Rules** | N/A |
| **Assumptions:** | * User is logged in. * User has completed his education till FSC. * User has added his academic details. * User has calculated his aggregate. |

# Use-Case 7.3 – Calculate aggregate

**Table 4.30:** UC-3 Calculate aggregate

|  |  |
| --- | --- |
| **Use Case ID:** | UC-7.3 |
| **Use Case Name:** | Calculate Aggregate. |
| **Actors:** | Student |
| **Description:** | User aggregate will be calculated for his selected university. |
| **Trigger:** | User clicks on Calculate Aggregate from One stop Admission Meeting Criteria Check screen. |
| **Preconditions:** | PRE-1. User is already signed in.  PRE-2. User has added his academic record previously into the system. |
| **Post conditions:** | POST-1. User will be displayed with his aggregate for his selected university. |
| **Normal Flow:** | 1. User clicks Calculate Aggregate button after logging into the system.  2. User selects the institute for which he wants his aggregate to be calculated.  2. User clicks on enter.  3. User’s aggregate is displayed. |
| **Alternative Flows:** | N/A |
| **Exceptions:** | * User is not logged in. * User has done O/A levels instead of matric/FSC system. * User has not added his academic details beforehand. |
| **Business Rules** | N/A |
| **Assumptions:** | * User is logged in. * User has completed his education till FSC. * User has added his academic details. |

# Use-Case 7.4 – Display Applicable Institutes

**Table 4.31:** UC-4 Display applicable institutes

|  |  |
| --- | --- |
| **Use Case ID:** | UC-7.4 |
| **Use Case Name:** | Display applicable Institutes. |
| **Actors:** | Student |
| **Description:** | A list of institutes that user can apply for is displayed according to his aggregate. |
| **Trigger:** | User clicks on Display Applicable Institutes from One stop Admission Meeting Criteria Check screen. |
| **Preconditions:** | PRE-1. User is already signed in.  PRE-2. User has added his academic record previously into the system.  PRE-3. User aggregate has been previously calculated by the system. |
| **Post conditions:** | POST-1. List of universities for which user can apply is displayed. |
| **Normal Flow:** | 1. User clicks Display Applicable Institutes button after logging into the system.  2. Applicable Institutes Screen is displayed.  3. Complete list of institutes that user can apply for is displayed. |
| **Alternative Flows:** | N/A  . |
| **Exceptions:** | * User is not logged in. * User has done O/A levels instead of matric/Fsc system. * User has not added his academic details beforehand. |
| **Business Rules** | N/A |
| **Assumptions:** | * User is logged in. * User has completed his education till FSc. * User has added his academic details. |

# Use-Case 7.5 – Get Admission guidelines

**Table 4.32:** UC-5 Get admission guidelines

|  |  |
| --- | --- |
| **Use Case ID:** | UC-7.5 |
| **Use Case Name:** | Get Admission Guidelines |
| **Actors:** | Student |
| **Description:** | User will be displayed with the admission guidelines for his selected university. |
| **Trigger:** | User clicks on Get Admission guidelines from One Stop Admission screen. |
| **Preconditions:** | PRE-1. User is already signed in. |
| **Post conditions:** | POST-1. User will be displayed with the admission guidelines for his selected university. |
| **Normal Flow:** | 1. User clicks Display Admission Guidelines button from One stop Admission Meeting Criteria Check screen after logging into the system.  2. User selects the university for which he wants the admission guidelines.  3. User is prompted to select from to view the admission schedule or how to apply?  4. User selects How to apply option.  5. User is displayed with a screen with complete procedure of How to apply for the specific university. |
| **Alternative Flows:** | 4a. User selects View Admission Schedule option.  4b. User is displayed with admission schedule for the current year of the specified university. |
| **Exceptions:** | * User is not logged in. |
| **Business Rules** | N/A |
| **Assumptions:** | * User is logged in. |

# Use-Case 7.6 – Dashboard

**Table 4.33:** UC-6 Dashboard

|  |  |
| --- | --- |
| **Use Case ID:** | UC-7.6 |
| **Use Case Name:** | Dashboard |
| **Actors:** | Student |
| **Description:** | Dashboard will be displayed containing user selected universities important admission dates i.e., start date of admission, admission deadline and the user selected apply status to give him a reminder. |
| **Trigger:** | User clicks on Get Admission guidelines from One Stop Admission screen. |
| **Preconditions:** | PRE-1. User is already signed in. |
| **Post conditions:** | POST-1. Dashboard will be displayed containing user selected universities important admission dates i.e. start date of admission, admission deadline and the user selected apply status. |
| **Normal Flow:** | 1. User clicks Dashboard button from One stop Admission Meeting Criteria Check screen after logging into the system.  2. User selects the university from the given list to be added onto the dashboard.  3. User clicks on add.  4. Dashboard containing user selected university admission dates and application status is displayed. |
| **Alternative Flows:** | N/A |
| **Exceptions:** | * User is not logged in. |
| **Business Rules** | N/A |
| **Assumptions:** | * User is logged in. |

**Module 8: Job portal**

# Use-Case 8.1 – Display Job Listings

**Table 4.34:** UC-1 Display Job listings

|  |  |
| --- | --- |
| **Use Case ID:** | UC-8.1 |
| **Use Case Name:** | Display Job Listings |
| **Actors:** | Student |
| **Description:** | All the job listings made by the employers will be displayed to the user. |
| **Trigger:** | User clicks on Get Admission guidelines from Job portal screen. |
| **Preconditions:** | PRE-1. User is already signed in.  PRE-2. Employer has added job Listings. |
| **Post conditions:** | POST-1. All the job listings made by the employers will be displayed to the user. |
| **Normal Flow:** | 1. User clicks Display Job Listings button from Job Portal screen after logging into the system.  2. All the job Listings available will be displayed to the user. |
| **Alternative Flows:** | 2a. In case of no available job listings, a prompt appears with message no Listing found. |
| **Exceptions:** | * User is not logged in. * No Job Listing previously added. |
| **Business Rules** | N/A |
| **Assumptions:** | * User is logged in. |

# Use-Case 8.2–Filter Job Listings

**Table 4.35:** UC-2 Filter Job listings

|  |  |
| --- | --- |
| **Use Case ID:** | UC-8.2 |
| **Use Case Name:** | Filter Job Listings |
| **Actors:** | Student |
| **Description:** | Job Listings will be filtered and displayed according to user filter selection. |
| **Trigger:** | User clicks on Get Admission guidelines from job portal screen. |
| **Preconditions:** | PRE-1. User is already signed in.  PRE-2. Employer has added job Listings. |
| **Post conditions:** | POST-1. Job Listings will be filtered and displayed according to user filter selection. |
| **Normal Flow:** | 1. User clicks Filter Job Listings button from Job Portal screen after logging into the system.  2. User selects the filter among the options of filter by latest date, filter by location, and filter by company/employer name.  3. User selects filter by location.  4. User is prompted to enter the location for the job listing.  5. Filter is applied according to user selection and filtered job listings are displayed. |
| **Alternative Flows:** | 3.1.  a. User selects filter by date.  b. Control transferred to step 5 of normal flow.  3.2.  a. User selects filter by employer name.  b. User enters employer’s name.  c. Control transferred to step 5 of normal flow. |
| **Exceptions:** | * User is not logged in. * No Job Listing exists. |
| **Business Rules** | N/A |
| **Assumptions:** | * User is logged in. |

# Use-Case 8.4–Upload Attachments

**Table 4.36:** UC-3 Upload Attachments

|  |  |
| --- | --- |
| **Use Case ID:** | UC-8.4 |
| **Use Case Name:** | Upload Attachments |
| **Actors:** | Student |
| **Description:** | User attachments i.e. CV and educational transcripts are uploaded into the system. |
| **Trigger:** | User clicks on Upload attachments from Job portal screen. |
| **Preconditions:** | PRE-1. User is already signed in.  PE-2. Employer has added job Listings. |
| **Post conditions:** | POST-1. User attachments i.e. CV and educational transcripts are uploaded into the system. |
| **Normal Flow:** | 1. User clicks upload attachments button from Job Portal screen after logging into the system.  2. User selects is prompted to select from choices scan and upload or upload documents.  3. User selects upload option.  4. User Attachments screen is displayed.  5. User clicks on upload CV.  6. User selects his CV.  7. User clicks on upload.  8. User clicks on upload matric transcript.  9. User selects his matric transcript.  10. User clicks on upload.  11. User clicks on upload Fsc transcript.  9. User selects his Fsc transcript.  10. User clicks on upload.  11. User click on proceed. |
| **Alternative Flows:** | 2a. User selects scan and upload option.  2b. User scans his CV.  2c. User scans his matric transcript.  2d. User scans his fsc transcript.  2e. Control transferred to step 3 of normal flow. |
| **Exceptions:** | * User is not logged in. * No Attachment found. |
| **Business Rules** | N/A |
| **Assumptions:** | * User is logged in. * Scanned transcripts exist with user in order to upload them. |

# Use-Case 8.5– Job Apply

**Table 4.37:** UC-5 Job apply

|  |  |
| --- | --- |
| **Use Case ID:** | UC-8.5 |
| **Use Case Name:** | Job apply |
| **Actors:** | Student |
| **Description:** | User applies for the job for the specified listing. |
| **Trigger:** | User clicks on Job apply from Job portal screen. |
| **Preconditions:** | PRE-1. User is already signed in.  PRE-2. User has added his attachments. |
| **Post conditions:** | POST-1. User attachments i.e., CV and educational transcripts are uploaded into the system. |
| **Normal Flow:** | 1. User clicks Job Apply button from Job Portal screen after logging into the system.  2. All the Job Listings are displayed.  3. User hovers to the listing that he wants to apply to.  4. User clicks on apply button besides the listing. |
| **Alternative Flows:** | N/A |
| **Exceptions:** | * User is not logged in. * No Attachments uploaded. |
| **Business Rules** | N/A |
| **Assumptions:** | * User is logged in. * Scanned transcripts exists with user in order to upload them. |

# Use-Case 8.6–Manage Job Listing

**Table 4.38:** UC-6 Manage Job listings

|  |  |
| --- | --- |
| **Use Case ID:** | UC-8.6 |
| **Use Case Name:** | Manage Job Listing |
| **Actors:** | Employer, Admin |
| **Description:** | User adds or updates a job listing to the system. |
| **Trigger:** | User clicks on Manage job Listing from Job portal screen. |
| **Preconditions:** | PRE-1. User is already registered. |
| **Post conditions:** | POST-1. A Job Listing is added or updated to the system. |
| **Normal Flow:** | 1. User clicks Manage Job Listing button from Job Portal screen after logging into the system.  2. User selects add job listing.  3. Job Listing screen appears.  4. User creates title for the listing.  5. User enters the description for the job.  6. User enters the location for the job.  7. User enters Experience required (if any) for the job.  8. User enters status for the listing i.e. open or closed.  9. User clicks on create listing.  10. Changes made are saved. |
| **Alternative Flows:** | N/A |
| **Exceptions:** | * User is not registered. |
| **Business Rules** | N/A |
| **Assumptions:** | * User is registered. |

# Use-Case 8.7–View Applicant Details

**Table 4.39:** UC-7 View applicant details

|  |  |
| --- | --- |
| **Use Case ID:** | UC-8.7 |
| **Use Case Name:** | View Applicant Details |
| **Actors:** | Employer |
| **Description:** | User views applicant Details. |
| **Trigger:** | User clicks on View Applicant’s Details from Job portal screen. |
| **Preconditions:** | PRE-1. User is already registered.  PRE-2. CV is preciously uploaded by the student. |
| **Post conditions:** | POST-1. User views applicant Details. |
| **Normal Flow:** | 1. User clicks View Applicant Details button from Job Portal screen after logging into the system.  2. All the applicants for who applied for the concerned job are displayed.  3. User selects the applicant.  4. Applicant profile appears.  5. User selects view Details.  6. Applicant Details gets displayed. |
| **Alternative Flows:** | 3a. User enters applicant name in the search applicant box.  3b. Matching result is displayed.  3c. Control transferred to step 4 of normal flow. |
| **Exceptions:** | * User is not registered. |
| **Business Rules** | N/A |
| **Assumptions:** | * User is registered. * Applicant has applied for the job. |

# Functional Requirements

Functional requirements of the proposed system are discussed below in detail.

**Module 1: User profiling**

# Enter First name

|  |  |
| --- | --- |
| **Identifier** | FR-1 |
| **Title** | Enter First name |
| **Requirement** | User enters his first name in the Enter first name dialogue box for it to be saved. |
| **Source** | user |
| **Rationale** | User first name to be saved in the app database. |
| **Business Rule (if required)** | BR-1. No special characters allowed |
| **Dependencies** | N/A |
| **Priority** | High |

# Enter Last name

|  |  |
| --- | --- |
| **Identifier** | FR-2 |
| **Title** | Enter Last name |
| **Requirement** | User enters his last name in the Enter last name dialogue box for it to be saved. |
| **Source** | user |
| **Rationale** | User last name to be saved in the app database. |
| **Business Rule (if required)** | BR-1. No special characters allowed |
| **Dependencies** | N/A |
| **Priority** | High |

# Enter email

|  |  |
| --- | --- |
| **Identifier** | FR-3 |
| **Title** | Enter email |
| **Requirement** | User enters his email in the Enter email dialogue box for it to be saved. |
| **Source** | user |
| **Rationale** | User email to be saved in the app database. |
| **Business Rule (if required)** | BR-1. User entered email should be valid. |
| **Dependencies** | N/A |
| **Priority** | High |

# Upload profile picture

|  |  |
| --- | --- |
| **Identifier** | FR-4 |
| **Title** | Upload profile picture |
| **Requirement** | User uploads his profile picture by clicking upload button to add his picture to his account. |
| **Source** | user |
| **Rationale** | User profile picture gets uploaded. |
| **Business Rule (if required)** | N/A |
| **Dependencies** | N/A |
| **Priority** | Low |

# Add contact number

|  |  |
| --- | --- |
| **Identifier** | FR-5 |
| **Title** | Add contact number |
| **Requirement** | User adds his contact number by entering his contact number in the prompted contact number box and clicks on enter to associate it with his account. |
| **Source** | user |
| **Rationale** | User contact number gets added. |
| **Business Rule (if required)** | N/A |
| **Dependencies** | N/A |
| **Priority** | High |

# Enter password

|  |  |
| --- | --- |
| **Identifier** | FR-6 |
| **Title** | Enter password |
| **Requirement** | User enters a valid password by typing it in the Enter password field. |
| **Source** | user |
| **Rationale** | User password gets entered. |
| **Business Rule (if required)** | N/A |
| **Dependencies** | N/A |
| **Priority** | High |

## 

# Confirm password

|  |  |
| --- | --- |
| **Identifier** | FR-7 |
| **Title** | Confirm password |
| **Requirement** | User confirms his password by typing his same previously entered password again in the confirm password field. |
| **Source** | user |
| **Rationale** | User password gets confirmed and added to database. |
| **Business Rule (if required)** | N/A |
| **Dependencies** | FR-6 |
| **Priority** | High |

# Sign-up

|  |  |
| --- | --- |
| **Identifier** | FR-8 |
| **Title** | Sign up |
| **Requirement** | User clicks on sign up button for his account to be created. |
| **Source** | user |
| **Rationale** | User new account gets created. |
| **Business Rule (if required)** | N/A |
| **Dependencies** | FR-3, FR-6 |
| **Priority** | High |

# Sign-in

|  |  |
| --- | --- |
| **Identifier** | FR-9 |
| **Title** | Sign in |
| **Requirement** | User after entering his email and password clicks on sign in button to log into the app. |
| **Source** | user |
| **Rationale** | User gets signed in. |
| **Business Rule (if required)** | BR-1. Valid email and password are entered. |
| **Dependencies** | FR-1, FR-2, FR-3, FR-4, FR-5, FR-6, FR-7 |
| **Priority** | High |

# Forgot password

|  |  |
| --- | --- |
| **Identifier** | FR-10 |
| **Title** | Forget password |
| **Requirement** | User clicks forget password to change his password. |
| **Source** | user |
| **Rationale** | User is helped to recover his password. |
| **Business Rule (if required)** | N/A |
| **Dependencies** | N/A |
| **Priority** | High |

# Enter new password

|  |  |
| --- | --- |
| **Identifier** | FR-11 |
| **Title** | Enter new password |
| **Requirement** | User enters his new password in the new password field. |
| **Source** | user |
| **Rationale** | To update users forgotten password. |
| **Business Rule (if required)** | N/A |
| **Dependencies** | FR-10 |
| **Priority** | High |

# Confirm new password

|  |  |
| --- | --- |
| **Identifier** | FR-12 |
| **Title** | Confirm new password |
| **Requirement** | User enters his new password again in the confirm password field to confirm his password. |
| **Source** | user |
| **Rationale** | To confirm user’s new password. |
| **Business Rule (if required)** | N/A |
| **Dependencies** | FR-11 |
| **Priority** | High |

# Confirm Log out

|  |  |
| --- | --- |
| **Identifier** | FR-13 |
| **Title** | Confirm Log out |
| **Requirement** | User click on confirm logout to be logged out of the app. |
| **Source** | user |
| **Rationale** | To get logged out of the app. |
| **Business Rule (if required)** | N/A |
| **Dependencies** | N/A |
| **Priority** | High |

# Update Account Info

|  |  |
| --- | --- |
| **Identifier** | FR-14 |
| **Title** | Update Account Info |
| **Requirement** | User click on update account info button to update his account information. |
| **Source** | user |
| **Rationale** | To update user account details. |
| **Business Rule (if required)** | N/A |
| **Dependencies** | N/A |
| **Priority** | Medium |

# Enter updated contact number

|  |  |
| --- | --- |
| **Identifier** | FR-15 |
| **Title** | Enter updated contact number |
| **Requirement** | User enters his new contact number to update his previous contact number. |
| **Source** | user |
| **Rationale** | To update user contact number. |
| **Business Rule (if required)** | BR-1. User new contact number should not match with the previous one. |
| **Dependencies** | N/A |
| **Priority** | Medium |

# Update profile picture

|  |  |
| --- | --- |
| **Identifier** | FR-16 |
| **Title** | Update profile picture |
| **Requirement** | User uploads his new profile picture to update it. |
| **Source** | user |
| **Rationale** | To update user profile picture. |
| **Business Rule (if required)** | N/A |
| **Dependencies** | N/A |
| **Priority** | Medium |

# Enter old password

|  |  |
| --- | --- |
| **Identifier** | FR-17 |
| **Title** | Enter old password |
| **Requirement** | User enters his old password in the old password field. |
| **Source** | user |
| **Rationale** | To change user old password. |
| **Business Rule (if required)** | N/A |
| **Dependencies** | N/A |
| **Priority** | Medium |

# Select Role

|  |  |
| --- | --- |
| **Identifier** | FR-18 |
| **Title** | Select Role |
| **Requirement** | User select his role i.e., 0 for student, 1 for Admin and 2 for counselor to gain the respective authority. |
| **Source** | user |
| **Rationale** | To select user’s role from admin, student or counselor. |
| **Business Rule (if required)** | N/A |
| **Dependencies** | N/A |
| **Priority** | High |

**Module 4: Institute Recommendation**

# View University

|  |  |
| --- | --- |
| **Identifier** | FR-19 |
| **Title** | View university |
| **Requirement** | User clicks on view university button to view university details. |
| **Source** | user |
| **Rationale** | To view university Information. |
| **Business Rule (if required)** | N/A |
| **Dependencies** | N/A |
| **Priority** | Medium |

# Select university

|  |  |
| --- | --- |
| **Identifier** | FR-20 |
| **Title** | Select university |
| **Requirement** | User selects university name from the given list to view concerned university details. |
| **Source** | user |
| **Rationale** | To view selected university Information. |
| **Business Rule (if required)** | N/A |
| **Dependencies** | N/A |
| **Priority** | Medium |

# Add preferred field

|  |  |
| --- | --- |
| **Identifier** | FR-21 |
| **Title** | Add preferred field. |
| **Requirement** | User adds his preferred field to get university recommendation. |
| **Source** | user |
| **Rationale** | To add user preferred field to get university recommendation. |
| **Business Rule (if required)** | N/A |
| **Dependencies** | N/A |
| **Priority** | Medium |

# Add preferred Location

|  |  |
| --- | --- |
| **Identifier** | FR-22 |
| **Title** | Add preferred Location. |
| **Requirement** | User adds his preferred location to get university recommendation. |
| **Source** | user |
| **Rationale** | To add user preferred location to get university recommendation. |
| **Business Rule (if required)** | N/A |
| **Dependencies** | N/A |
| **Priority** | Medium |

# Update preferred field

|  |  |
| --- | --- |
| **Identifier** | FR-23 |
| **Title** | Update preferred field. |
| **Requirement** | User updates his preferred field which he previously entered to get university recommendation. |
| **Source** | user |
| **Rationale** | To update user preferred field to get university recommendation. |
| **Business Rule (if required)** | N/A |
| **Dependencies** | FR-21 |
| **Priority** | Medium |

# Update preferred Location

|  |  |
| --- | --- |
| **Identifier** | FR-24 |
| **Title** | Update preferred Location. |
| **Requirement** | User updates his preferred location to get university recommendation. |
| **Source** | user |
| **Rationale** | To update user preferred location to get university recommendation. |
| **Business Rule (if required)** | N/A |
| **Dependencies** | FR-22 |
| **Priority** | Medium |

# Show University recommendation

|  |  |
| --- | --- |
| **Identifier** | FR-25 |
| **Title** | Show university recommendation. |
| **Requirement** | User click on get university recommendation button to get his result displayed. |
| **Source** | user |
| **Rationale** | To view university recommendation. |
| **Business Rule (if required)** | N/A |
| **Dependencies** | FR-21, FR-22 |
| **Priority** | Medium |

# Enter first university name

|  |  |
| --- | --- |
| **Identifier** | FR-26 |
| **Title** | Enter first university name. |
| **Requirement** | User enters first university name in the specified field. |
| **Source** | user |
| **Rationale** | To enter university name that is to be compared. |
| **Business Rule (if required)** | N/A |
| **Dependencies** | N/A |
| **Priority** | Medium |

# Enter second university name

|  |  |
| --- | --- |
| **Identifier** | FR-27 |
| **Title** | Enter second university name. |
| **Requirement** | User enters second university name in the specified field. |
| **Source** | user |
| **Rationale** | To enter university name that is to be compared with. |
| **Business Rule (if required)** | N/A |
| **Dependencies** | N/A |
| **Priority** | Medium |

# Compare university

|  |  |
| --- | --- |
| **Identifier** | FR-28 |
| **Title** | Compare university. |
| **Requirement** | User click on compare university button. |
| **Source** | user |
| **Rationale** | To compare user selected universities. |
| **Business Rule (if required)** | N/A |
| **Dependencies** | FR-26, FR-27 |
| **Priority** | Medium |

# Compare universities courses

|  |  |
| --- | --- |
| **Identifier** | FR-29 |
| **Title** | Compare universities courses. |
| **Requirement** | User click on compare universities courses button. |
| **Source** | user |
| **Rationale** | To compare user selected universities courses offered. |
| **Business Rule (if required)** | BR-1. User has selected both the universities before clicking on comparing button. |
| **Dependencies** | FR-26, FR-27 |
| **Priority** | Medium |

# Compare universities statistics

|  |  |
| --- | --- |
| **Identifier** | FR-30 |
| **Title** | Compare universities statistics. |
| **Requirement** | User click on compare universities statistics button. |
| **Source** | user |
| **Rationale** | To compare user selected universities courses statistics. |
| **Business Rule (if required)** | BR-1. User has selected both the universities before clicking on comparing button. |
| **Dependencies** | FR-26, FR-27 |
| **Priority** | Medium |

# Send email

|  |  |
| --- | --- |
| **Identifier** | FR-31 |
| **Title** | Send email. |
| **Requirement** | User click on send email button form contact university via email screen. |
| **Source** | user |
| **Rationale** | To send university email. |
| **Business Rule (if required)** | N/A |
| **Dependencies** | FR-26, FR-27 |
| **Priority** | Medium |

# Ask Question

|  |  |
| --- | --- |
| **Identifier** | FR-32 |
| **Title** | Ask Question. |
| **Requirement** | User click on ask question button form FAQ main screen. |
| **Source** | user |
| **Rationale** | To ask a question related to a university from other users. |
| **Business Rule (if required)** | N/A |
| **Dependencies** | N/A |
| **Priority** | Medium |

# View FAQ form

|  |  |
| --- | --- |
| **Identifier** | FR-33 |
| **Title** | View FAQ form. |
| **Requirement** | User click on view FAQ from button from FAQ’s main screen. |
| **Source** | User |
| **Rationale** | To view frequently asked questions page by other users related to a university. |
| **Business Rule (if required)** | N/A |
| **Dependencies** | N/A |
| **Priority** | Medium |

**Module 7: One Stop Admission meeting criteria check**

# Add matric marks

|  |  |
| --- | --- |
| **Identifier** | FR-34 |
| **Title** | Add matric marks. |
| **Requirement** | User click on add matric marks button from Academic details screen. |
| **Source** | user |
| **Rationale** | To add user Matric marks. |
| **Business Rule (if required)** | N/A |
| **Dependencies** | N/A |
| **Priority** | Medium |

# Add Fsc marks

|  |  |
| --- | --- |
| **Identifier** | FR-35 |
| **Title** | Add Fsc marks. |
| **Requirement** | User click on add Fsc marks button from Academic details screen. |
| **Source** | user |
| **Rationale** | To add user Fsc marks. |
| **Business Rule (if required)** | N/A |
| **Dependencies** | FR-43, FR-44 |
| **Priority** | Medium |

# Add School

|  |  |
| --- | --- |
| **Identifier** | FR-36 |
| **Title** | Add School. |
| **Requirement** | User click on add School button from Academic details screen. |
| **Source** | user |
| **Rationale** | To add user Secondary school. |
| **Business Rule (if required)** | N/A |
| **Dependencies** | FR-43, FR-44 |
| **Priority** | Medium |

# Add College

|  |  |
| --- | --- |
| **Identifier** | FR-37 |
| **Title** | Add College. |
| **Requirement** | User click on add College button from Academic details screen. |
| **Source** | user |
| **Rationale** | To add user College. |
| **Business Rule (if required)** | N/A |
| **Dependencies** | FR-43, FR-44 |
| **Priority** | Medium |

# Save Academic Details

|  |  |
| --- | --- |
| **Identifier** | FR-38 |
| **Title** | Save Academic Details. |
| **Requirement** | User click on Save Academic Details button. |
| **Source** | user |
| **Rationale** | To save user academic details in database. |
| **Business Rule (if required)** | N/A |
| **Dependencies** | N/A |
| **Priority** | Medium |

# Edit Matric marks

|  |  |
| --- | --- |
| **Identifier** | FR-39 |
| **Title** | Edit matrix marks. |
| **Requirement** | User click on edit matric marks button. |
| **Source** | user |
| **Rationale** | To update user matric marks in database. |
| **Business Rule (if required)** | N/A |
| **Dependencies** | FR-34 |
| **Priority** | Medium |

# Edit FSC Marks

|  |  |
| --- | --- |
| **Identifier** | FR-40 |
| **Title** | Edit FSC marks. |
| **Requirement** | User click on edit FSC marks button. |
| **Source** | user |
| **Rationale** | To update user FSC marks in database. |
| **Business Rule (if required)** | N/A |
| **Dependencies** | FR-35 |
| **Priority** | Medium |

# Edit school

|  |  |
| --- | --- |
| **Identifier** | FR-41 |
| **Title** | Edit school. |
| **Requirement** | User click on edit school button. |
| **Source** | user |
| **Rationale** | To update user school in database. |
| **Business Rule (if required)** | N/A |
| **Dependencies** | FR-36 |
| **Priority** | Medium |

# Edit college

|  |  |
| --- | --- |
| **Identifier** | FR-42 |
| **Title** | Edit college. |
| **Requirement** | User click on edit college button. |
| **Source** | user |
| **Rationale** | To edit user college name in database. |
| **Business Rule (if required)** | N/A |
| **Dependencies** | FR-37 |
| **Priority** | Medium |

# Compute Aggregate

|  |  |
| --- | --- |
| **Identifier** | FR-43 |
| **Title** | Compute Aggregate. |
| **Requirement** | User click on Compute Aggregate button. |
| **Source** | user |
| **Rationale** | To compute user aggregate for a specific university. |
| **Business Rule (if required)** | N/A |
| **Dependencies** | FR-34, FR-35, FR-36, FR-37 |
| **Priority** | Medium |

# Show Applicable Institutes

|  |  |
| --- | --- |
| **Identifier** | FR-44 |
| **Title** | Show Applicable Institutes. |
| **Requirement** | User click on Show Applicable Institutes button. |
| **Source** | user |
| **Rationale** | To view List of Institutes in which user can apply for. |
| **Business Rule (if required)** | N/A |
| **Dependencies** | N/A |
| **Priority** | Medium |

# Select University

|  |  |
| --- | --- |
| **Identifier** | FR-45 |
| **Title** | Select university. |
| **Requirement** | User click on select university button. |
| **Source** | user |
| **Rationale** | To select university for which user can later see guidelines. |
| **Business Rule (if required)** | N/A |
| **Dependencies** | N/A |
| **Priority** | Medium |

# View Admission Schedule

|  |  |
| --- | --- |
| **Identifier** | FR-46 |
| **Title** | View Admission Schedule. |
| **Requirement** | User click on View Admission Schedule button. |
| **Source** | user |
| **Rationale** | To view admission schedule for the concerned university. |
| **Business Rule (if required)** | BR-1. Current year admission schedule is displayed. |
| **Dependencies** | FR-45 |
| **Priority** | Medium |

# How to Apply?

|  |  |
| --- | --- |
| **Identifier** | FR-47 |
| **Title** | How to Apply? |
| **Requirement** | User click on How to Apply button. |
| **Source** | user |
| **Rationale** | To view procedure for the admission steps for concerned university. |
| **Business Rule (if required)** | N/A |
| **Dependencies** | FR-45 |
| **Priority** | Medium |

**Module 8: Job portal**

# Show Job Listings

|  |  |
| --- | --- |
| **Identifier** | FR-48 |
| **Title** | Show Job Listings |
| **Requirement** | User click on Show Job Listings button. |
| **Source** | user |
| **Rationale** | To view Job Listings. |
| **Business Rule (if required)** | N/A |
| **Dependencies** | N/A |
| **Priority** | High |

# Filter Job Listings by date

|  |  |
| --- | --- |
| **Identifier** | FR-49 |
| **Title** | Filter Job Listings by date |
| **Requirement** | User click on Filter Job Listings by Date option. |
| **Source** | user |
| **Rationale** | To Filter Job Listings by date. |
| **Business Rule (if required)** | N/A |
| **Dependencies** | N/A |
| **Priority** | Low |

# Filter Job Listings by location

|  |  |
| --- | --- |
| **Identifier** | FR-50 |
| **Title** | Filter Job Listings by location |
| **Requirement** | User click on Filter Job Listings by Location option. |
| **Source** | user |
| **Rationale** | To Filter Job Listings by Location. |
| **Business Rule (if required)** | N/A |
| **Dependencies** | N/A |
| **Priority** | Low |

# Filter Job Listings by Employer Name

|  |  |
| --- | --- |
| **Identifier** | FR-51 |
| **Title** | Filter Job Listings by Employer Name |
| **Requirement** | User click on Filter Job Listings by Employer Name option. |
| **Source** | user |
| **Rationale** | To Filter Job Listings by Employer Name. |
| **Business Rule (if required)** | N/A |
| **Dependencies** | N/A |
| **Priority** | Low |

# Upload CV

|  |  |
| --- | --- |
| **Identifier** | FR-52 |
| **Title** | Upload CV |
| **Requirement** | User click on upload CV option. |
| **Source** | user |
| **Rationale** | To upload user’s CV. |
| **Business Rule (if required)** | N/A |
| **Dependencies** | N/A |
| **Priority** | Medium |

# Upload Matric Transcript

|  |  |
| --- | --- |
| **Identifier** | FR-53 |
| **Title** | Upload Matric Transcript |
| **Requirement** | User clicks on upload Matric Transcript option. |
| **Source** | user |
| **Rationale** | To upload user’s Matric Transcript. |
| **Business Rule (if required)** | N/A |
| **Dependencies** | N/A |
| **Priority** | Medium |

# 

# Upload Fsc Transcript

|  |  |
| --- | --- |
| **Identifier** | FR-54 |
| **Title** | Upload Fsc Transcript |
| **Requirement** | User click on upload Fsc Transcript option. |
| **Source** | User |
| **Rationale** | To upload user’s Fsc Transcript. |
| **Business Rule (if required)** | N/A |
| **Dependencies** | N/A |
| **Priority** | Medium |

# Select Job Listing

|  |  |
| --- | --- |
| **Identifier** | FR-55 |
| **Title** | Select Job Listing |
| **Requirement** | User selects Job Listing from the listings displayed. |
| **Source** | User |
| **Rationale** | To select job listing to apply for it. |
| **Business Rule (if required)** | N/A |
| **Dependencies** | N/A |
| **Priority** | Medium |

# Apply Job

|  |  |
| --- | --- |
| **Identifier** | FR-56 |
| **Title** | Apply Job |
| **Requirement** | User click on Apply Job option. |
| **Source** | User |
| **Rationale** | To apply for a job from the list of job listings. |
| **Business Rule (if required)** | N/A |
| **Dependencies** | FR-55 |
| **Priority** | Medium |

# Add Job title

|  |  |
| --- | --- |
| **Identifier** | FR-57 |
| **Title** | Add Job title |
| **Requirement** | User enters Job title for which listing is to be created. |
| **Source** | User (Employer) |
| **Rationale** | To add a job title for which listing is to be created. |
| **Business Rule (if required)** | N/A |
| **Dependencies** | N/A |
| **Priority** | High |

# Add Job Location

|  |  |
| --- | --- |
| **Identifier** | FR-58 |
| **Title** | Add Job Location |
| **Requirement** | User enters Job Location for which listing is to be created. |
| **Source** | User (Employer) |
| **Rationale** | To add a job Location for which listing is to be created. |
| **Business Rule (if required)** | N/A |
| **Dependencies** | N/A |
| **Priority** | High |

# Add Job experience required

|  |  |
| --- | --- |
| **Identifier** | FR-59 |
| **Title** | Add Job experience required. |
| **Requirement** | User enters minimum Job experience for which listing is to be created |
| **Source** | User (Employer) |
| **Rationale** | To add a job required experience for which listing is to be created. |
| **Business Rule (if required)** | N/A |
| **Dependencies** | N/A |
| **Priority** | High |

# Enter Job Description

|  |  |
| --- | --- |
| **Identifier** | FR-60 |
| **Title** | Enter Job description. |
| **Requirement** | User enters Job description for which listing is to be created. |
| **Source** | User (Employer) |
| **Rationale** | To add a job description for which listing is to be created. |
| **Business Rule (if required)** | N/A |
| **Dependencies** | N/A |
| **Priority** | High |

# Job availability status

|  |  |
| --- | --- |
| **Identifier** | FR-61 |
| **Title** | Job availability status. |
| **Requirement** | User enters job availability status for concerned listing. |
| **Source** | User (Employer) |
| **Rationale** | To update job availability status (open/closed) for concerned listing. |
| **Business Rule (if required)** | N/A |
| **Dependencies** | N/A |
| **Priority** | High |

# Cancel Job Listing

|  |  |
| --- | --- |
| **Identifier** | FR-62 |
| **Title** | Cancel Job Listing. |
| **Requirement** | User cancels job listing which he previously created. |
| **Source** | User (Employer) |
| **Rationale** | To cancels job listing which he previously created. |
| **Business Rule (if required)** | N/A |
| **Dependencies** | N/A |
| **Priority** | High |

# View Applicant CV

|  |  |
| --- | --- |
| **Identifier** | FR-63 |
| **Title** | View Applicant CV. |
| **Requirement** | User selects View Applicant CV option. |
| **Source** | User (Employer) |
| **Rationale** | To view Applicant CV. |
| **Business Rule (if required)** | N/A |
| **Dependencies** | N/A |
| **Priority** | High |

**Module 2: Virtual Sessions**

# Mic request

|  |  |
| --- | --- |
| **Identifier** | FR-64 |
| **Title** | Mic request |
| **Requirement** | System request mic access to place audio call. |
| **Source** | system |
| **Rationale** | System permission of mic to be saved for app. |
| **Business Rule (if required)** | BR-1. Mic device to be installed on system. |
| **Dependencies** | N/A |
| **Priority** | High |

# Allow mic access

|  |  |
| --- | --- |
| **Identifier** | FR-65 |
| **Title** | Allow mic access |
| **Requirement** | User allow app to use system mic |
| **Source** | user |
| **Rationale** | User response is saved in device permissions |
| **Business Rule (if required)** | BR-1. User to allow mic to be used for calls |
| **Dependencies** | N/A |
| **Priority** | High |

# Allow camera access

|  |  |
| --- | --- |
| **Identifier** | FR-76 |
| **Title** | Allow camera access |
| **Requirement** | User allow app to use system camera |
| **Source** | User |
| **Rationale** | User response is saved in device permissions |
| **Business Rule (if required)** | BR-1. User to allow camera to be used for video calls |
| **Dependencies** | N/A |
| **Priority** | High |

# Place audio call

|  |  |
| --- | --- |
| **Identifier** | FR-67 |
| **Title** | Place audio call |
| **Requirement** | User click on green call button to place call |
| **Source** | user |
| **Rationale** | User call is connected |
| **Business Rule (if required)** | N/A |
| **Dependencies** | FR-73, FR-74 |
| **Priority** | High |

# Place video call

|  |  |
| --- | --- |
| **Identifier** | FR-68 |
| **Title** | Place video call |
| **Requirement** | User click on green camera button to place video call |
| **Source** | user |
| **Rationale** | User video call is connected |
| **Business Rule (if required)** | N/A |
| **Dependencies** | FR-73,FR-78 |
| **Priority** | High |

# Decline audio call

|  |  |
| --- | --- |
| **Identifier** | FR-69 |
| **Title** | Decline audio call |
| **Requirement** | User slides left to decline voice call |
| **Source** | user |
| **Rationale** | User call is disconnected |
| **Business Rule (if required)** | N/A |
| **Dependencies** | FR-67 |
| **Priority** | High |

# Decline video call

|  |  |
| --- | --- |
| **Identifier** | FR-70 |
| **Title** | Decline video call |
| **Requirement** | User slides left to decline video call |
| **Source** | user |
| **Rationale** | User call is disconnected |
| **Business Rule (if required)** | N/A |
| **Dependencies** | FR-68 |
| **Priority** | High |

# Call Feedback

|  |  |
| --- | --- |
| **Identifier** | FR-71 |
| **Title** | Call Feedback |
| **Requirement** | User gives feedback about call session when session ends |
| **Source** | user |
| **Rationale** | User feedback is sent to admin for performance record. |
| **Business Rule (if required)** | N/A |
| **Dependencies** | FR-67, FR-68 |
| **Priority** | Low |

# Display Call log

|  |  |
| --- | --- |
| **Identifier** | FR-72 |
| **Title** | Display Call Log |
| **Requirement** | User selects display call log button |
| **Source** | user |
| **Rationale** | Users call log for audio and video calls are displayed. |
| **Business Rule (if required)** | N/A |
| **Dependencies** | N/A |
| **Priority** | Low |

# Generate call Id

|  |  |
| --- | --- |
| **Identifier** | FR-73 |
| **Title** | Generate Call Id |
| **Requirement** | User selects Generate Call Id button |
| **Source** | user |
| **Rationale** | Call Id for the meeting gets generated through which another user can join. |
| **Business Rule (if required)** | N/A |
| **Dependencies** | N/A |
| **Priority** | High |

# Copy call Id

|  |  |
| --- | --- |
| **Identifier** | FR-74 |
| **Title** | Copy Call Id |
| **Requirement** | User selects Copy Call Id button |
| **Source** | user |
| **Rationale** | Call Id for the meeting gets copied to help other user in order to join the meeting. |
| **Business Rule (if required)** | N/A |
| **Dependencies** | N/A |
| **Priority** | High |

**Module 3: Aptitude Test**

# Start Test

|  |  |
| --- | --- |
| **Identifier** | FR-75 |
| **Title** | Start Test |
| **Requirement** | User clicks start test button to generate test. |
| **Source** | user |
| **Rationale** | MCQs based aptitude test is displayed |
| **Business Rule (if required)** | N/A |
| **Dependencies** | N/A |
| **Priority** | Medium |

# End Test

|  |  |
| --- | --- |
| **Identifier** | FR-76 |
| **Title** | End Test |
| **Requirement** | User clicks end test button to end test. |
| **Source** | user |
| **Rationale** | Test is ended |
| **Business Rule (if required)** | N/A |
| **Dependencies** | FR-9 |
| **Priority** | High |

# Select MCQs Option

|  |  |
| --- | --- |
| **Identifier** | FR-77 |
| **Title** | Select MCQs Option |
| **Requirement** | User to select any option from available 4 options |
| **Source** | user |
| **Rationale** | MCQs option response is recorded. |
| **Business Rule (if required)** | N/A |
| **Dependencies** | N/A |
| **Priority** | High |

# Deselect MCQs Option

|  |  |
| --- | --- |
| **Identifier** | FR-78 |
| **Title** | Deselect MCQs Options |
| **Requirement** | User to deselect any option selected. |
| **Source** | user |
| **Rationale** | MCQs option response is recorded. |
| **Business Rule (if required)** | N/A |
| **Dependencies** | N/A |
| **Priority** | High |

# Submit Test

|  |  |
| --- | --- |
| **Identifier** | FR-79 |
| **Title** | Submit Test |
| **Requirement** | User to select submit test button. |
| **Source** | User |
| **Rationale** | Aptitude test attempted by user gets submitted. |
| **Business Rule (if required)** | N/A |
| **Dependencies** | FR-77 |
| **Priority** | High |

# View Results

|  |  |
| --- | --- |
| **Identifier** | FR-80 |
| **Title** | View Results |
| **Requirement** | User to click view result button. |
| **Source** | user |
| **Rationale** | Aptitude test result is displayed as per user responses. |
| **Business Rule (if required)** | N/A |
| **Dependencies** | FR-79 |
| **Priority** | Medium |

# View Test standing

|  |  |
| --- | --- |
| **Identifier** | FR-81 |
| **Title** | View Test standing |
| **Requirement** | User to click test standing button. |
| **Source** | user |
| **Rationale** | To allow user to check his score among other users attempts and his own previous attempts. |
| **Business Rule (if required)** | N/A |
| **Dependencies** | FR-79 |
| **Priority** | Medium |

# View detailed results

|  |  |
| --- | --- |
| **Identifier** | FR-82 |
| **Title** | View detailed Result |
| **Requirement** | User to click View detailed Results button. |
| **Source** | user |
| **Rationale** | To allow user to View detailed result in graph form. |
| **Business Rule (if required)** | N/A |
| **Dependencies** | FR-79 |
| **Priority** | Medium |

**Module 6: AI Chatbot**

# Enter Query

|  |  |
| --- | --- |
| **Identifier** | FR-83 |
| **Title** | Enter Query |
| **Requirement** | User type his query for the chatbot |
| **Source** | user |
| **Rationale** | User to be able to type text to make a query. |
| **Business Rule (if required)** | N/A |
| **Dependencies** | N/A |
| **Priority** | High |

# Show university rankings

|  |  |
| --- | --- |
| **Identifier** | FR-84 |
| **Title** | Show university rankings |
| **Requirement** | User asks chatbot to show university rankings. |
| **Source** | User |
| **Rationale** | Chatbot to display University rankings to the user. |
| **Business Rule (if required)** | N/A |
| **Dependencies** | N/A |
| **Priority** | Medium |

# Show top careers

|  |  |
| --- | --- |
| **Identifier** | FR-85 |
| **Title** | Show Top careers |
| **Requirement** | User asks chatbot to show top careers. |
| **Source** | User |
| **Rationale** | Chatbot to display top careers to the user. |
| **Business Rule (if required)** | N/A |
| **Dependencies** | N/A |
| **Priority** | Medium |

# Submit feedback

|  |  |
| --- | --- |
| **Identifier** | FR-86 |
| **Title** | Show Top careers |
| **Requirement** | User selects submit feedback button after entering it. |
| **Source** | User |
| **Rationale** | To receive user’s feedback. |
| **Business Rule (if required)** | N/A |
| **Dependencies** | N/A |
| **Priority** | Medium |

**Module 5: Entry test preparation**

# Select institution

|  |  |
| --- | --- |
| **Identifier** | FR-87 |
| **Title** | Select institution |
| **Requirement** | User will click on select preferred institution from the displayed list. |
| **Source** | User |
| **Rationale** | To select institution from the displayed list. |
| **Business Rule (if required)** | N/A |
| **Dependencies** | N/A |
| **Priority** | Medium |

# View Past Papers

|  |  |
| --- | --- |
| **Identifier** | FR-88 |
| **Title** | View past papers |
| **Requirement** | User will click on View past papers button. |
| **Source** | user |
| **Rationale** | User will be able view past papers of the selected university. |
| **Business Rule (if required)** | N/A |
| **Dependencies** | FR-87, FR-89 |
| **Priority** | Medium |

# Select Year

|  |  |
| --- | --- |
| **Identifier** | FR-89 |
| **Title** | Select Year |
| **Requirement** | User shall select specific year for desired past papers |
| **Source** | user |
| **Rationale** | User will be able to select year for past papers from the year list. |
| **Business Rule (if required)** | User need to select any year to view past papers. |
| **Dependencies** | N/A |
| **Priority** | High |

# Start Mock test

|  |  |
| --- | --- |
| **Identifier** | FR-90 |
| **Title** | Start Mock test |
| **Requirement** | User shall select start Mock Test button in Test preparation section. |
| **Source** | user |
| **Rationale** | User will be able to attempt mock test. |
| **Business Rule (if required)** | N/A |
| **Dependencies** | FR-87 |
| **Priority** | High |

# View Mock Test result

|  |  |
| --- | --- |
| **Identifier** | FR-91 |
| **Title** | View mock test result |
| **Requirement** | User will select View mock test result option. |
| **Source** | User |
| **Rationale** | Mock test results attempted by the user will be displayed. |
| **Business Rule (if required)** | N/A |
| **Dependencies** | FR-92 |
| **Priority** | High |

# Submit test

|  |  |
| --- | --- |
| **Identifier** | FR-92 |
| **Title** | Submit test |
| **Requirement** | User will select submit test option after attempting the test. |
| **Source** | User |
| **Rationale** | Test attempted by the user shall be submitted. |
| **Business Rule (if required)** | N/A |
| **Dependencies** | FR-90 |
| **Priority** | High |

# End test

|  |  |
| --- | --- |
| **Identifier** | FR-93 |
| **Title** | End test |
| **Requirement** | User will need to click on End test button to End test |
| **Source** | user |
| **Rationale** | User will be able to end the test. |
| **Business Rule (if required)** | N/A |
| **Dependencies** | FR-90 |
| **Priority** | High |

# Show Results

|  |  |
| --- | --- |
| **Identifier** | FR-94 |
| **Title** | Show Results |
| **Requirement** | User clicks on Show result button after attempting test. |
| **Source** | user |
| **Rationale** | User will be able to view result of attempted mock test. |
| **Business Rule (if required)** | N/A |
| **Dependencies** | FR-92 |
| **Priority** | Medium |

# View discussion form

|  |  |
| --- | --- |
| **Identifier** | FR-95 |
| **Title** | View discussion form |
| **Requirement** | User click on view discussion form option. |
| **Source** | User |
| **Rationale** | User will be able to view discussion form congaing queries (if any) of other users regarding entry test preparation. |
| **Business Rule (if required)** | N/A |
| **Dependencies** | N/A |
| **Priority** | Medium |

# Enter question

|  |  |
| --- | --- |
| **Identifier** | FR-96 |
| **Title** | Enter question |
| **Requirement** | User enters his question in the given space provided. |
| **Source** | User |
| **Rationale** | User will be able to enter his question that he possesses in the discussion form. |
| **Business Rule (if required)** | N/A |
| **Dependencies** | N/A |
| **Priority** | Medium |

# Submit answer

|  |  |
| --- | --- |
| **Identifier** | FR-97 |
| **Title** | Submit answer |
| **Requirement** | User answers the question(s) ask by other users and clicks on submit button in discussion form. |
| **Source** | User |
| **Rationale** | User answer to the question(s) ask by other users gets submitted and displayed in discussion form. |
| **Business Rule (if required)** | N/A |
| **Dependencies** | FR-95 |
| **Priority** | Medium |

# Add Video (learning material)

|  |  |
| --- | --- |
| **Identifier** | FR-98 |
| **Title** | Add video |
| **Requirement** | User adds video as the learning material by entering its URL. |
| **Source** | User |
| **Rationale** | Video gets added and displayed in learning materials section. |
| **Business Rule (if required)** | N/A |
| **Dependencies** | N/A |
| **Priority** | Medium |

# Add notes (learning material)

|  |  |
| --- | --- |
| **Identifier** | FR-99 |
| **Title** | Add notes |
| **Requirement** | User adds notes as the learning material by selecting add notes option and uploading pdf file of notes. |
| **Source** | User |
| **Rationale** | To add Pdf of notes to learning material section. |
| **Business Rule (if required)** | N/A |
| **Dependencies** | N/A |
| **Priority** | Medium |

# View video

|  |  |
| --- | --- |
| **Identifier** | FR-97 |
| **Title** | View video |
| **Requirement** | User clicks on view video option in learning material section. |
| **Source** | User |
| **Rationale** | To view video of the required learning material. |
| **Business Rule (if required)** | N/A |
| **Dependencies** | FR-98 |
| **Priority** | Medium |

# View notes

|  |  |
| --- | --- |
| **Identifier** | FR-97 |
| **Title** | View notes |
| **Requirement** | User clicks on view notes option in learning material section. |
| **Source** | User |
| **Rationale** | To view notes of the required learning material. |
| **Business Rule (if required)** | N/A |
| **Dependencies** | FR-99 |
| **Priority** | Medium |

# Non-Functional Requirements

Non-functional requirements are listed below.

# Reliability

*RLE-1: System remains operational as long as it has an active internet connection.*

# Usability

*USE-1: The CDB shall allow the user to view all the universities in which he can apply for in a single screen along with their respective information.*

*USE-1: System will be very user friendly because of AI Chabot assisting him with all his queries.*

*USE-2: System will allow user to finalize job listings and apply for the desired one all from a single screen.*

# Performance

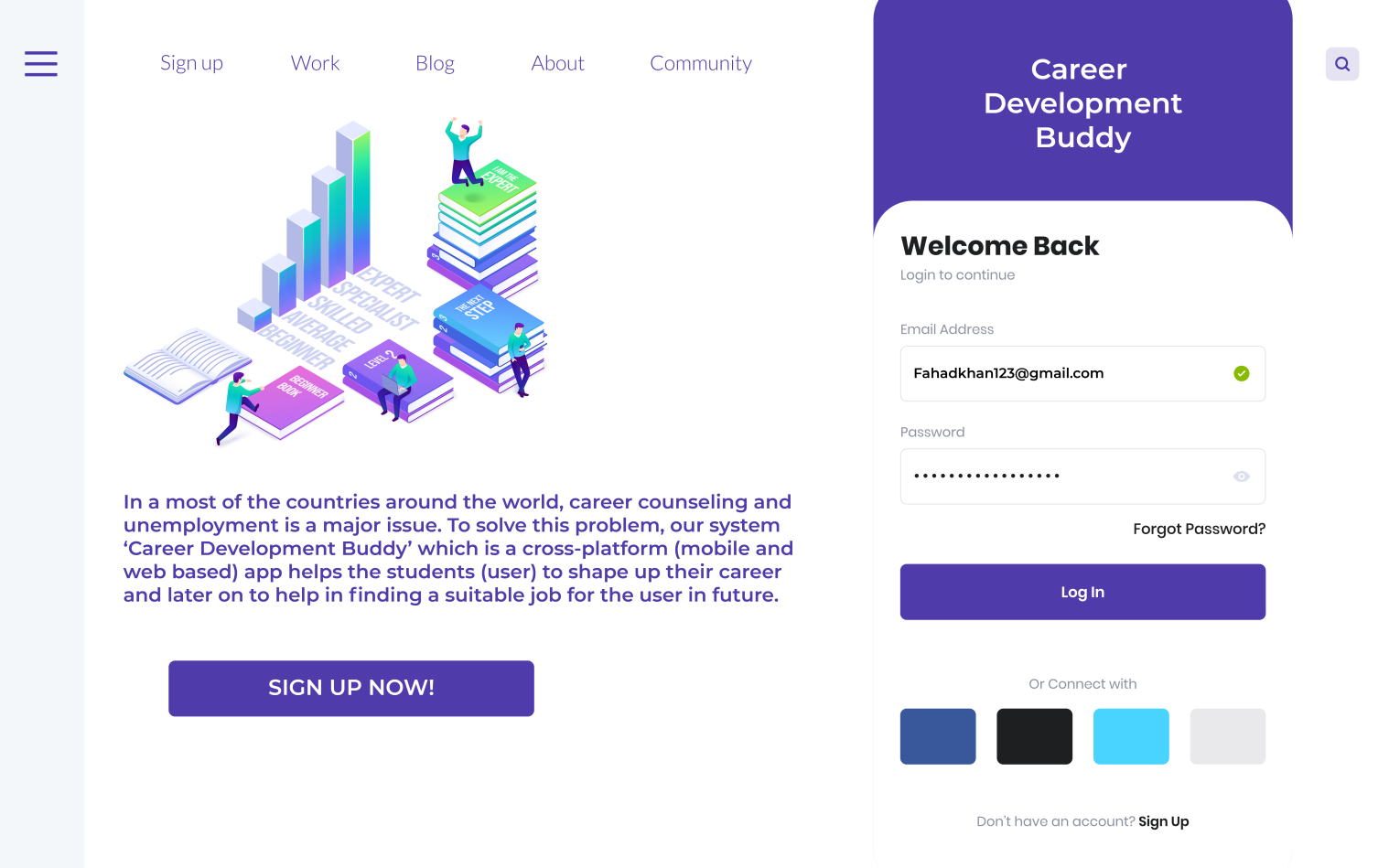
*PER-1: System shall accommodate up to 4000 user’s data.*

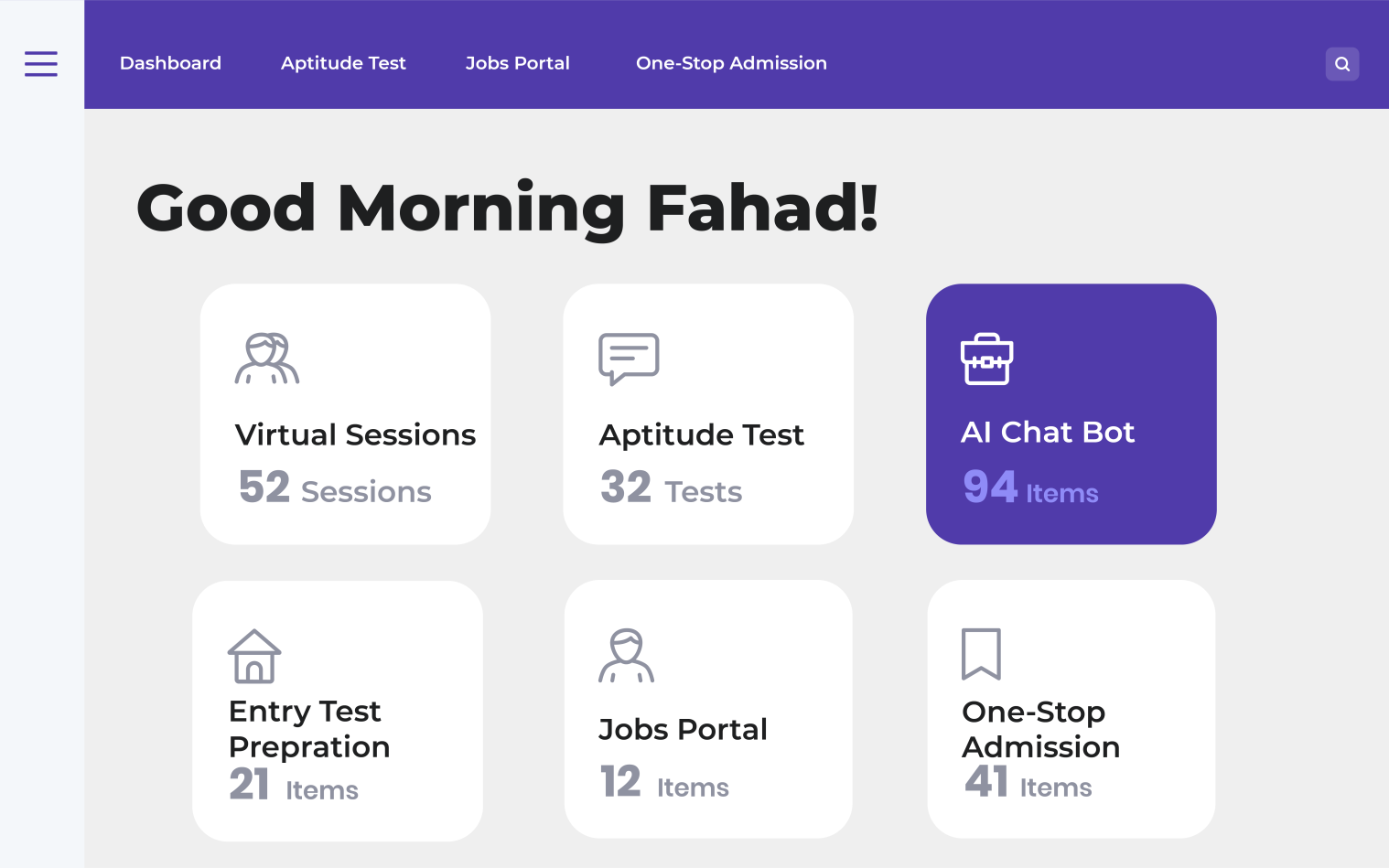
# Security

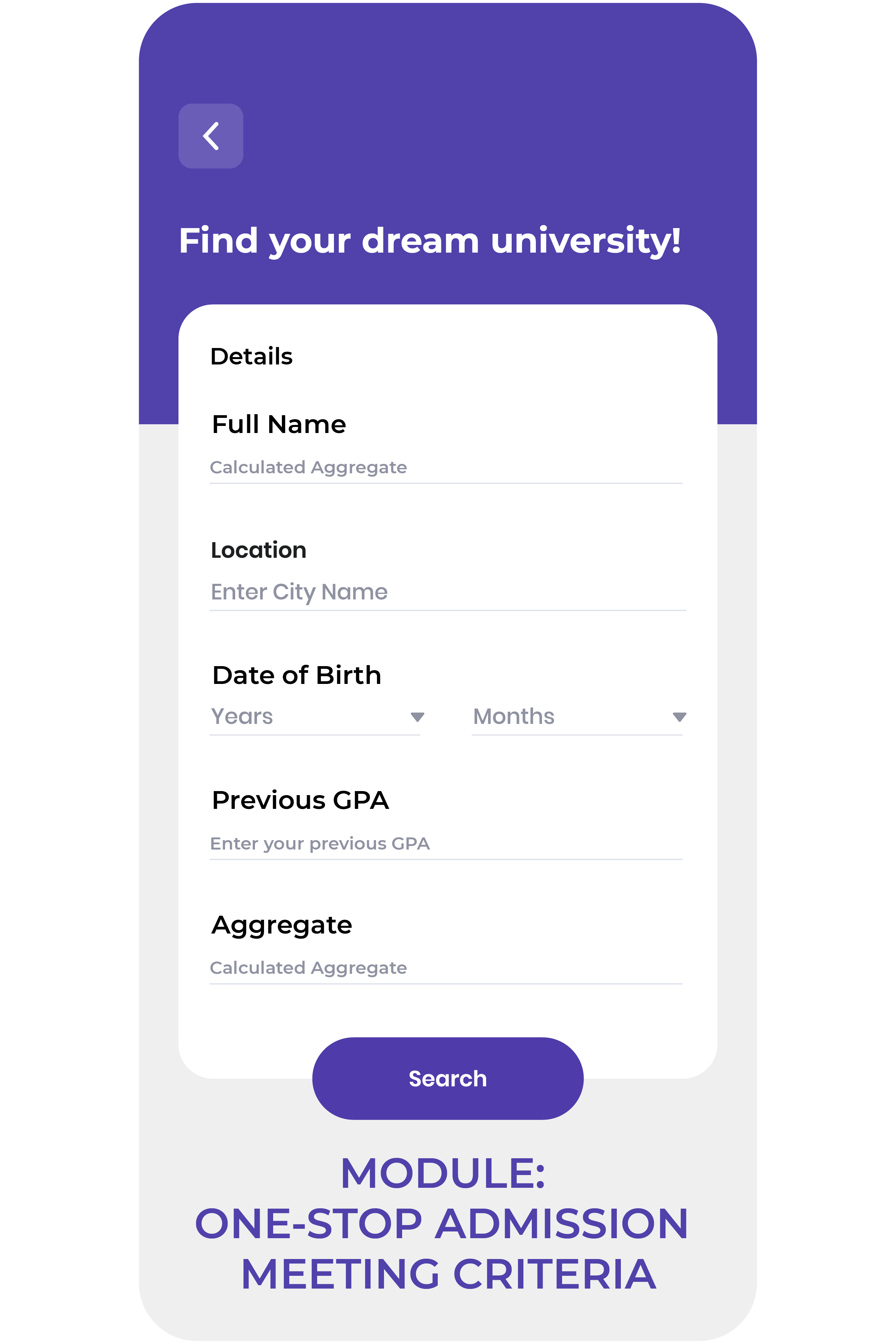
*SE-1: System can be breached by a hacker with medium skill level.*

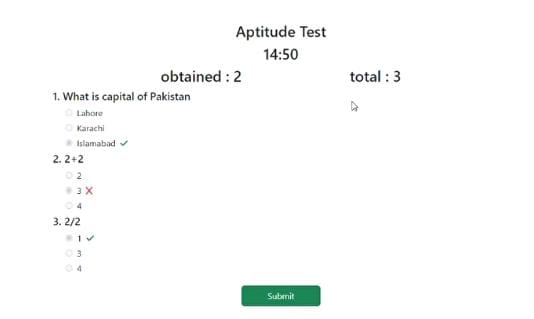
# External Interface Requirements

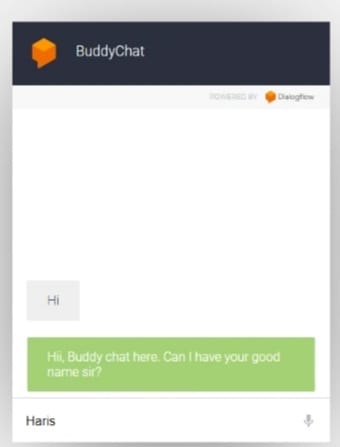
# Sample user Interfaces











.

# Software interfaces

*SI-1.1: The CDB shall store the entire user’s information using MongoDB database.*

*SI-1.2: The application must use React.*

.

# Communications interfaces

*CI-1: The CDB shall prompt the user to verify his email when during registration process.*

*CL-2: The CDB shall prompt the user to verify his himself when he changing his password.*

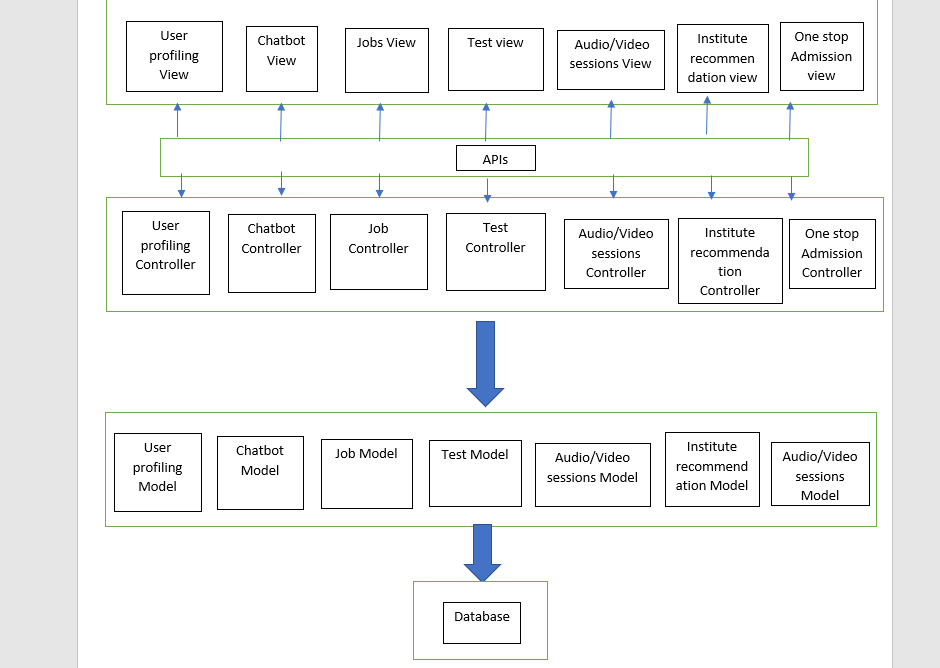
# Design and Architecture

Design and architecture of proposed system is described as follow.



# Architectural Design

System follows MVC architecture as:

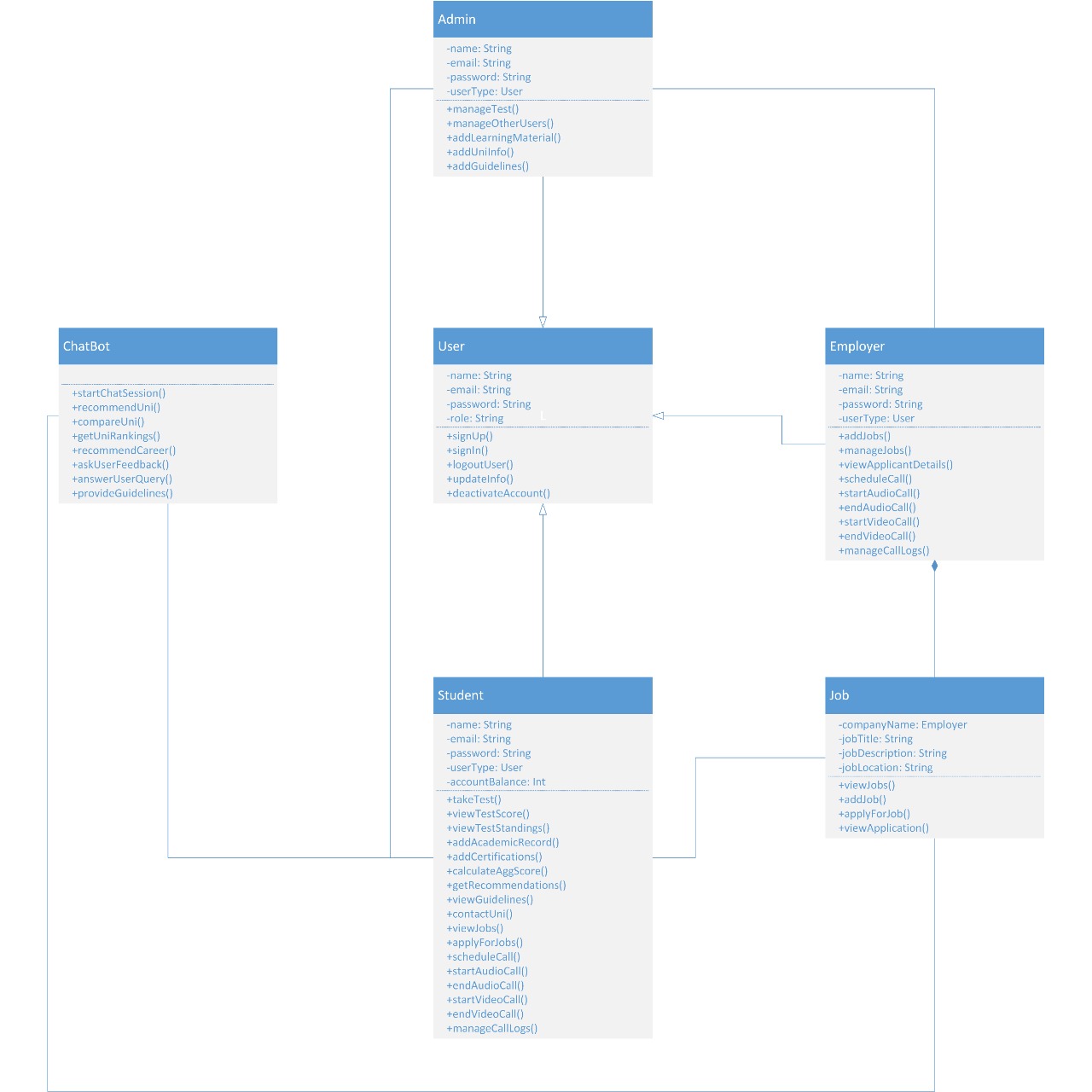


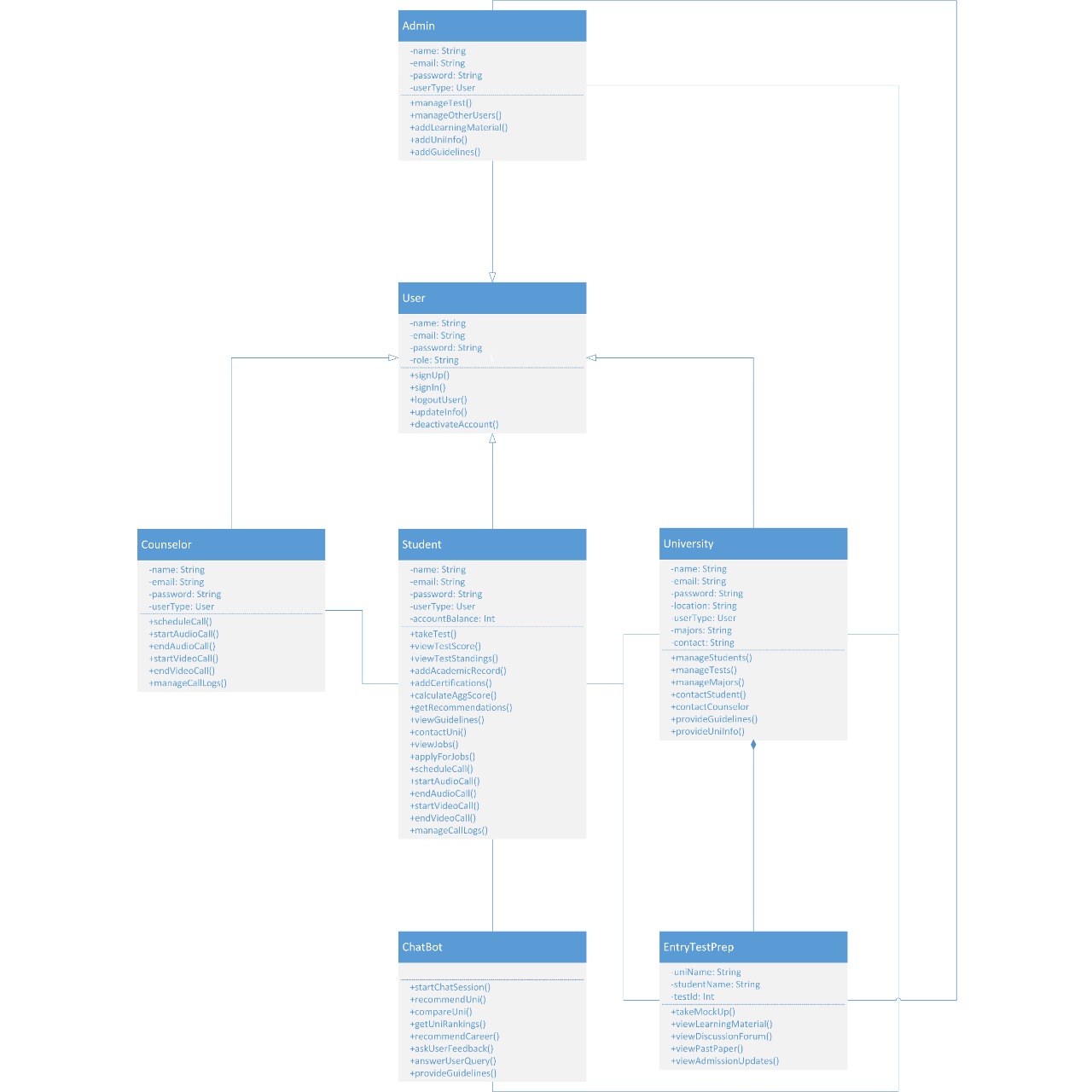
.

# Design Models

As design model is OOP so design models are as follows:

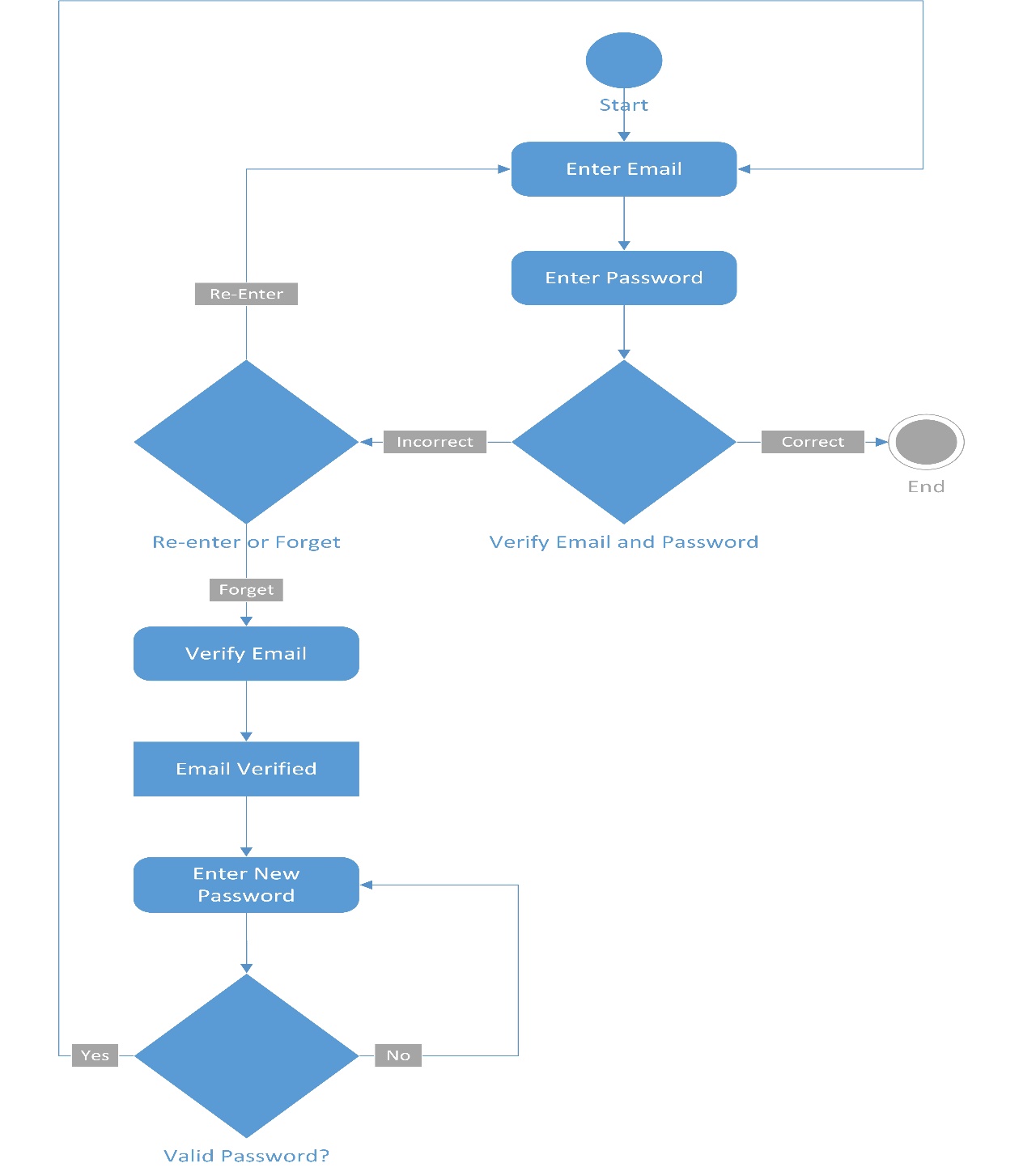
# Class Diagram



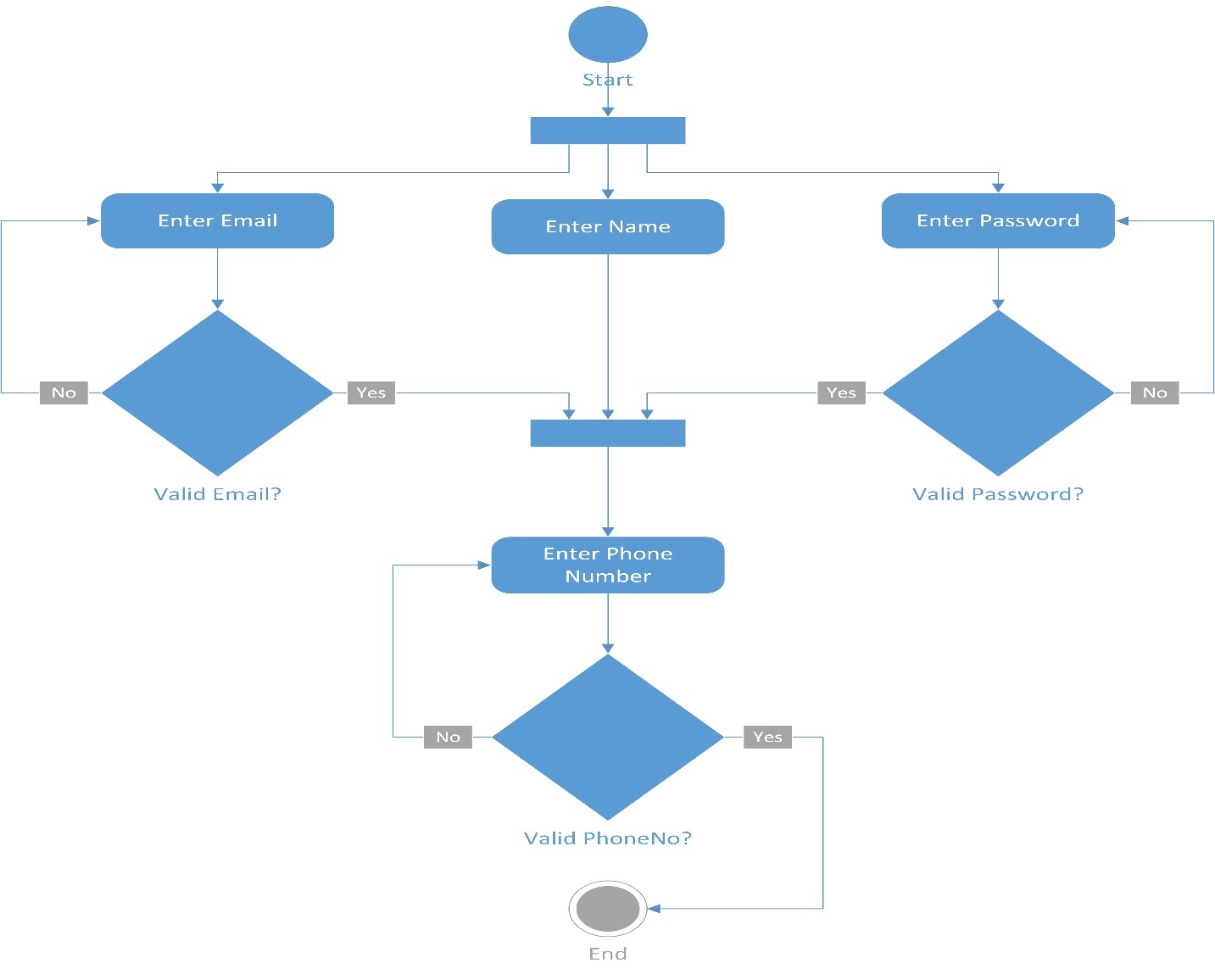


# Activity Diagrams

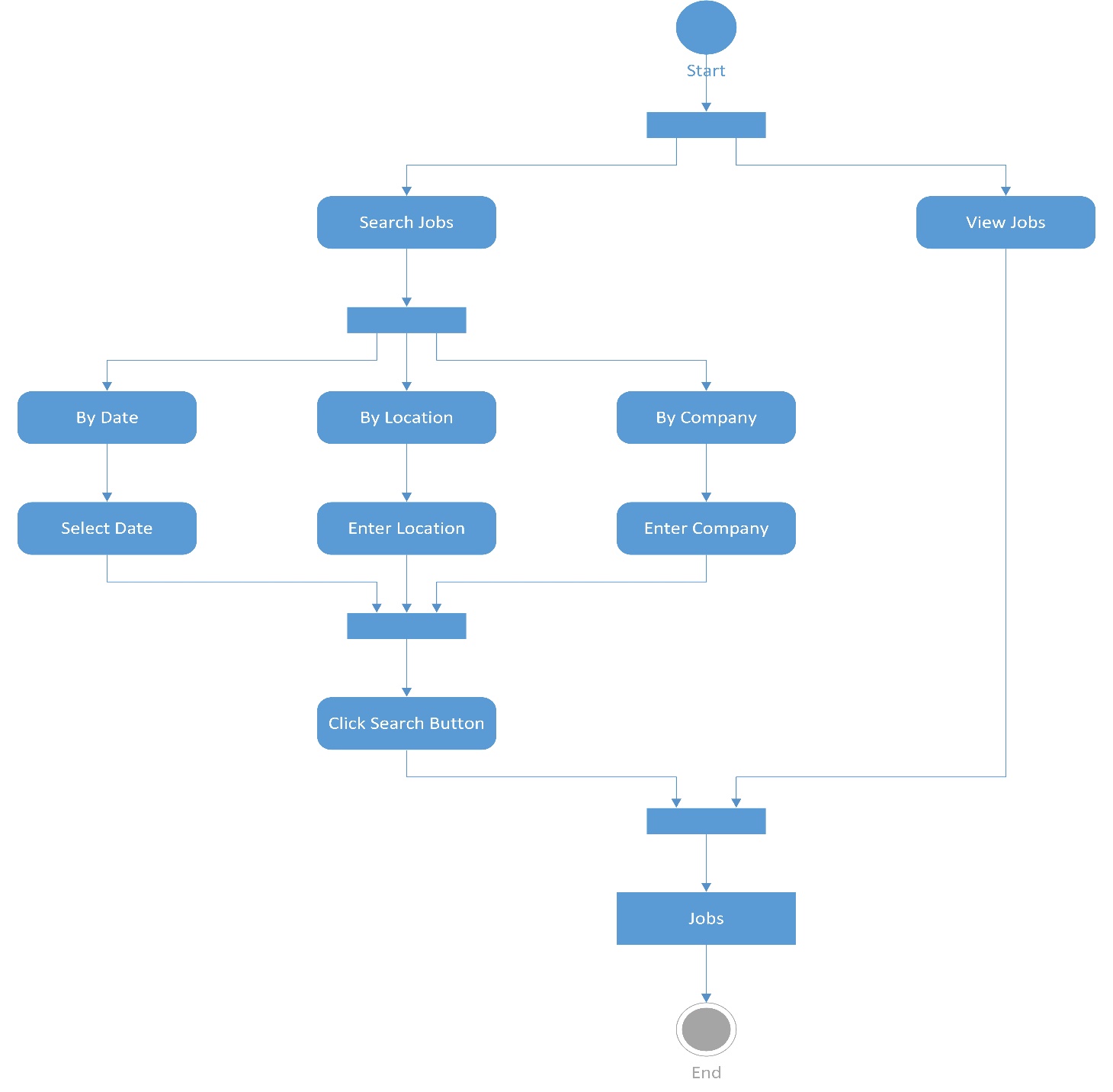
# Sign-in



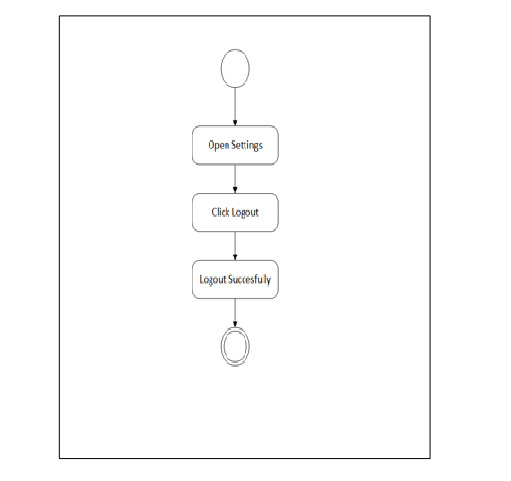
# Sign-up



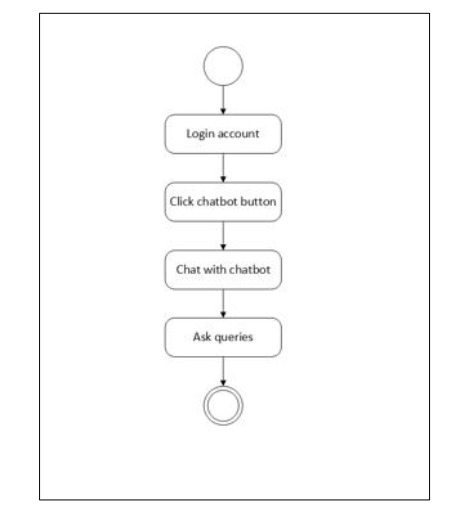
# View/ Search Job Listing



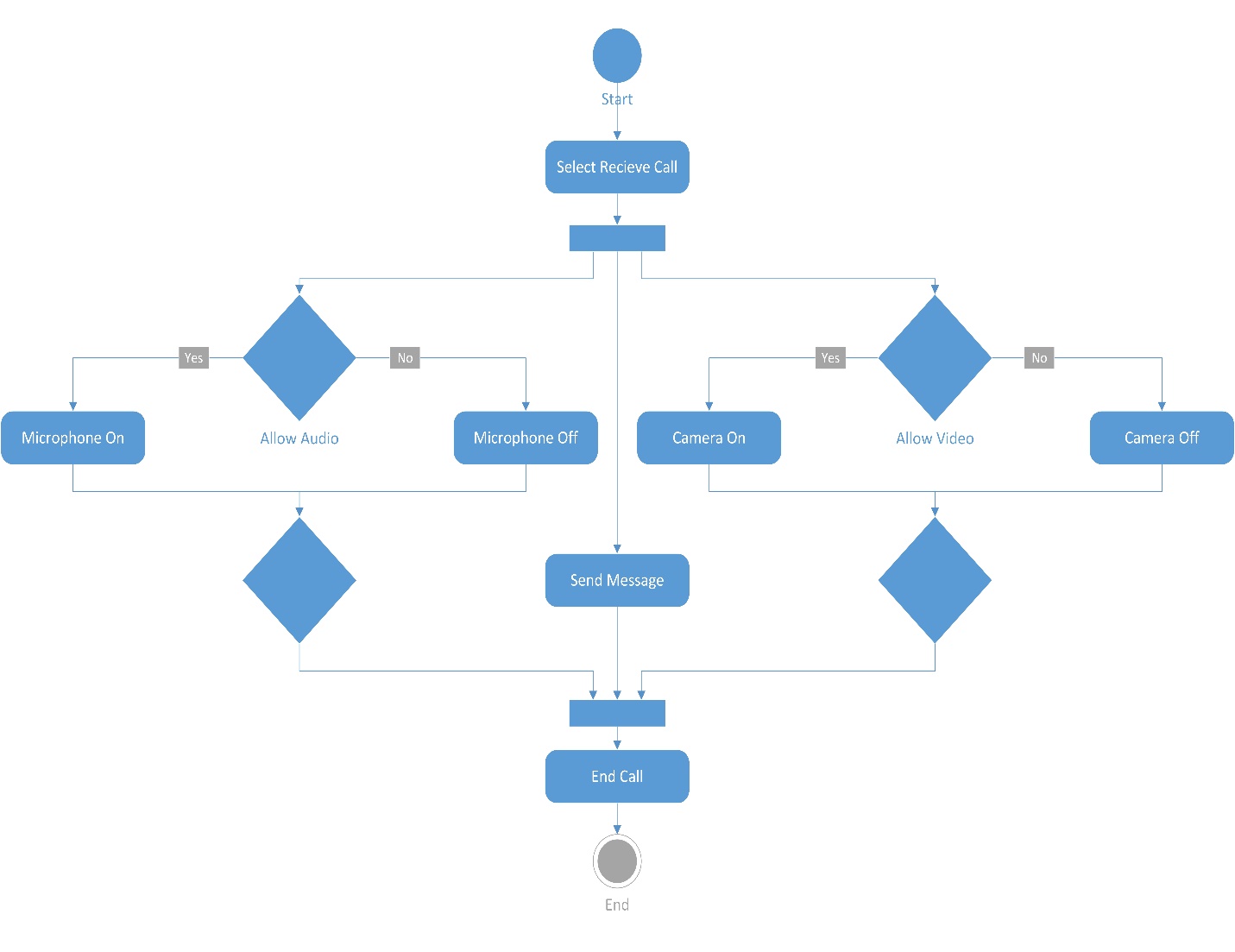
# Logout



# Chatbot

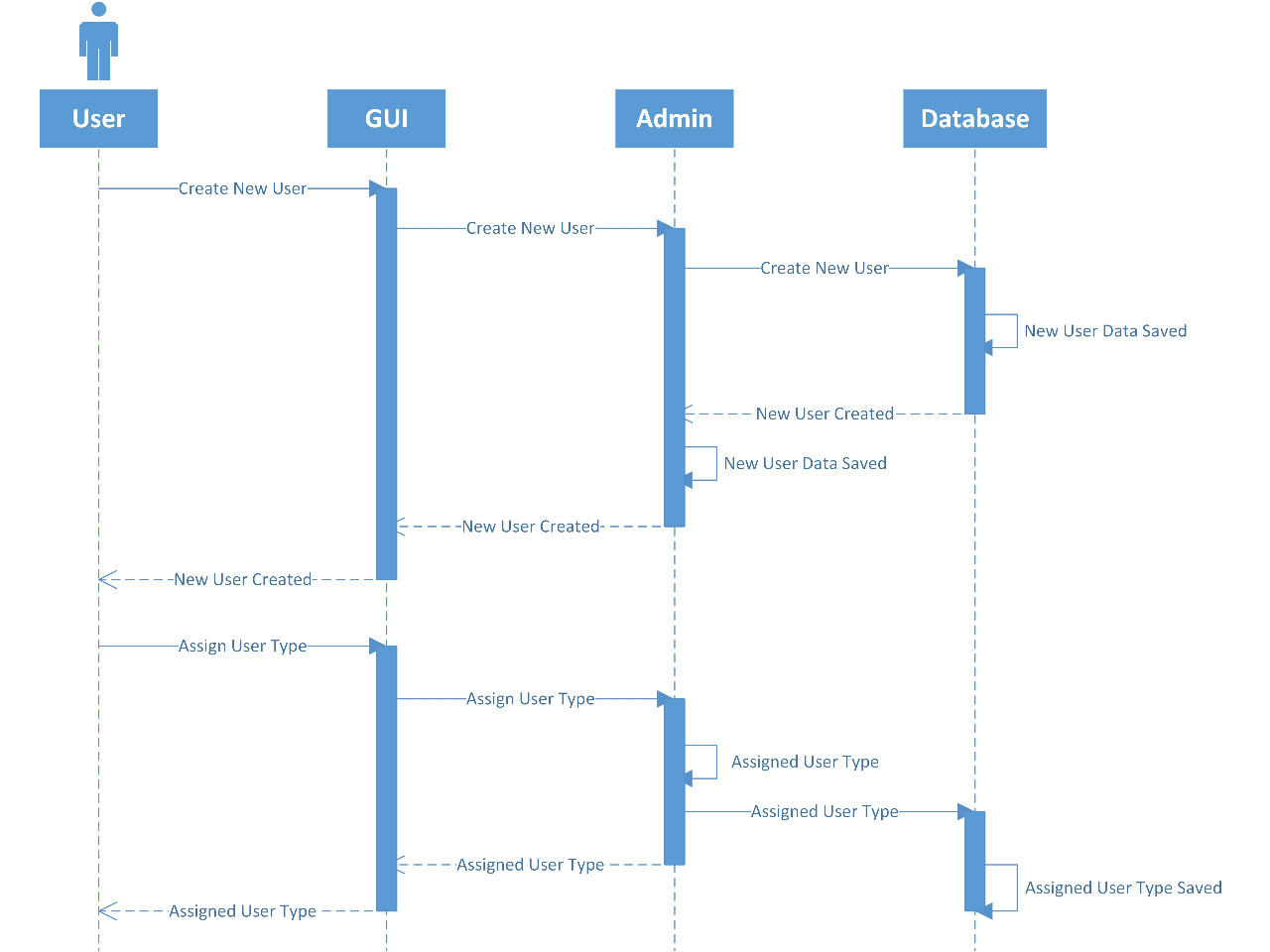


# Join call

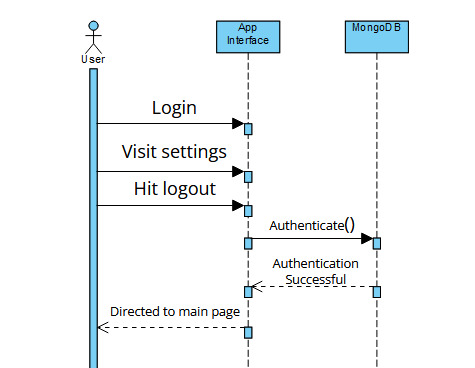


# Sequence Diagrams

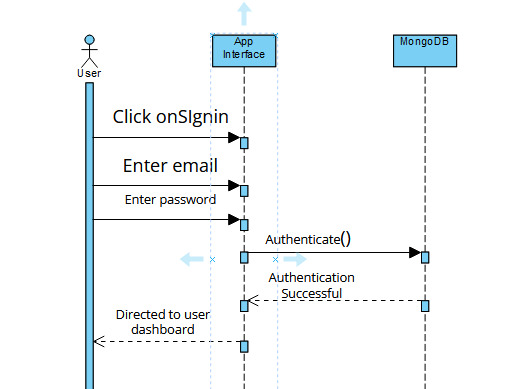
**Signup**



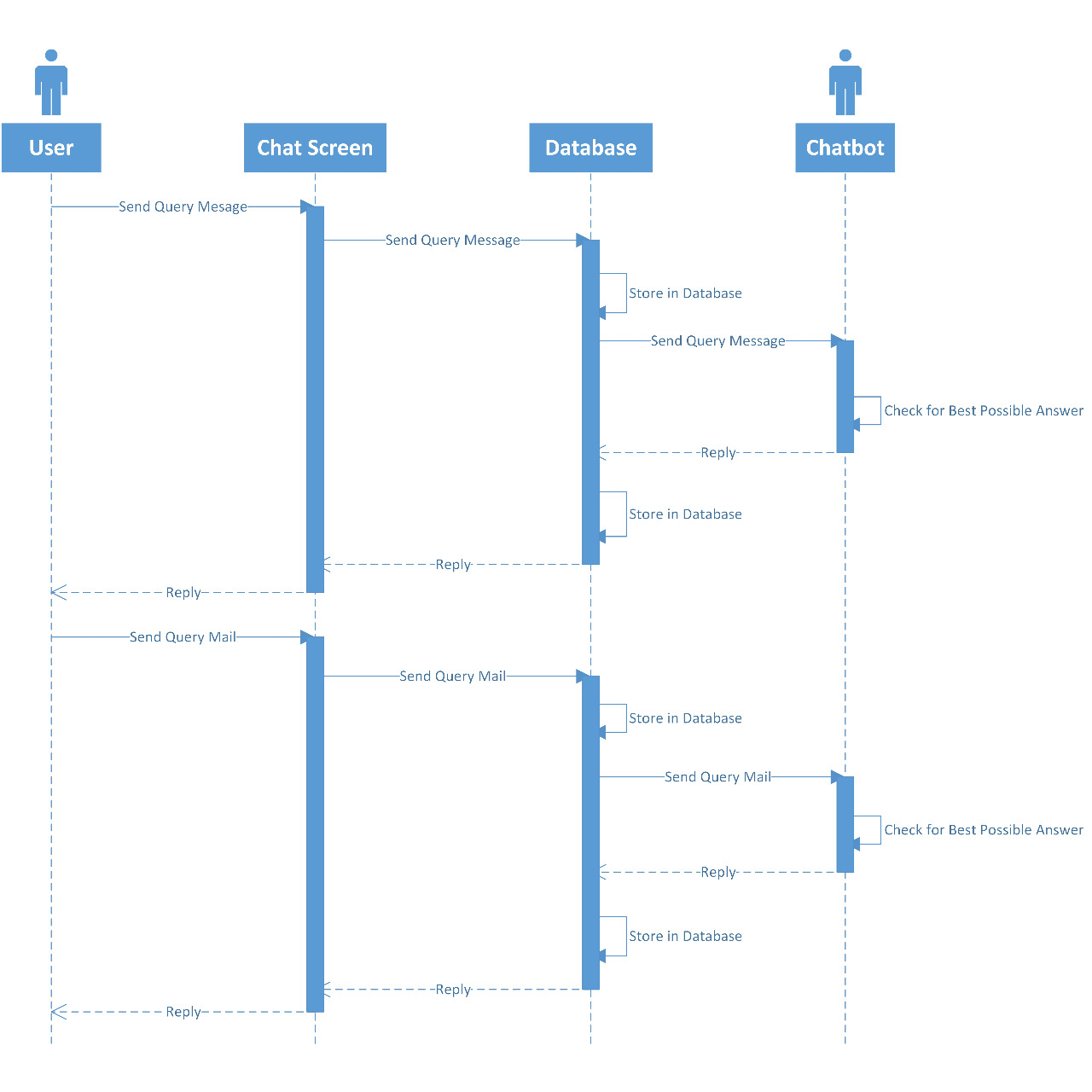
**Logout**



**Login**



**Chatbot**



# Data Design

MongoDB will be used. Following is the Mongo schema:

**Academics:**

const academicSchema = new mongoose.Schema(

{

full\_name: {

type: String,

required: ["Full name not provided"],

},

matric\_marks: {

type: String,

required: ["Metric marks not provided"],

},

school: {

type: String,

required: ["School not provided"],

},

fsc\_marks: {

type: String,

required: ["Fsc. marks not provided"],

},

college: {

type: String,

required: ["College not provided"],

},

user: {

type: mongoose.Types.ObjectId,

ref: "users",

required: true,

},

},

{

timestamps: true,

}

);

**CallLog:**

const callLogSchema = new mongoose.Schema(

{

call\_id: {

type: "string",

},

caller: {

type: "ObjectId",

ref: "User",

},

receiver: {

type: "ObjectId",

ref: "User",

},

start\_time: {

type: Date,

required: true,

},

end\_time: {

type: Date,

required: true,

},

},

{

timestamps: true,

}

);

**Discussion:**

const fileSchema = new mongoose.Schema({

query: {

type: String,

trim: true

},

response: {

type: [String]

}

}, {

timestamps: true

})

**Files:**

const fileSchema = new mongoose.Schema(

{

name: {

type: String,

trim: true,

},

email: {

type: String,

trim: true,

},

cv: {

type: String,

trim: true,

},

fsc: {

type: String,

trim: true,

},

matric: {

type: String,

},

job\_id: {

type: mongoose.Schema.Types.ObjectId,

ref: "jobs",

},

},

{

timestamps: true,

}

);

**Job:**

const jobSchema = new mongoose.Schema({

companyName: {

type: String,

trim: true

},

posted\_by: {

type: mongoose.Types.ObjectId,

required: true

},

title: {

type: String,

required: true,

trim: true

},

description: {

type: String,

required: true,

trim: true,

},

location: {

type: String,

required: true,

},

experience: {

type: String,

},

}, {

timestamps: true

})

**Paper:**

const fileSchema = new mongoose.Schema({

university: {

type: String,

trim: true

},

year: {

type: String,

trim: true

},

pdf: {

type: String,

trim: true

},

}, {

timestamps: true

})

**PDF:**

const fileSchema = new mongoose.Schema({

pdf: {

type: String,

trim: true

},

}, {

timestamps: true

})

**Question:**

const questionsSchema = new mongoose.Schema(

{

category: {

type: String,

trim: true,

required: ["Catagory not provided"],

},

question: {

type: String,

trim: true,

required: ["Question not provided"],

},

options: {

type: Array,

default: [],

},

correct\_answer: {

type: String,

trim: true,

required: ["Correct Answer not provided"],

},

},

{

timestamps: true,

}

);

**Result:**

const categoryWiseMarks = {

categoryType: {

type: mongoose.Schema.Types.String,

required: true,

},

marks: {

type: mongoose.Schema.Types.Number,

required: true,

},

};

const resultsSchema = new mongoose.Schema(

{

user: {

type: "ObjectId",

ref: "User",

required: true,

},

category: {

type: [categoryWiseMarks], // array of category marks

default: [],

},

total: {

type: Number,

default: 0,

},

obtained: {

type: Number,

default: 0,

},

},

{

timestamps: true,

}

);

**Test:**

const testSchema = new mongoose.Schema({

university: {

type: String,

trim: true,

required: true,

},

question: {

type: String,

required: true,

trim: true

},

A: {

type: String,

required: true,

trim: true,

},

B: {

type: String,

required: true,

},

C: {

type: String,

required: true,

},

D: {

type: String,

required: true,

},

correctOption: {

type: String,

required: true,

},

}, {

timestamps: true

})

**University:**

const universitySchema = new mongoose.Schema(

{

name: {

type: String,

required: ["Catagory not provided"],

},

merit: {

type: String,

required: ["Question not provided"],

},

speciality: {

type: String,

required: ["Speciality not provided"],

},

city: {

type: String,

required: ["Correct Answer not provided"],

},

university\_page: {

type: String,

default: "https://www.google.com",

},

},

{

timestamps: true,

}

);

**User:**

const userSchema = new mongoose.Schema(

{

name: {

type: String,

required: true,

trim: true,

},

email: {

type: String,

required: true,

trim: true,

unique: true,

},

password: {

type: String,

required: true,

},

role: {

type: Number,

default: 0,

},

image: {

type: String,

},

},

{

timestamps: true,

}

);

**Video:**

const videoSchema = new mongoose.Schema({

video: {

type: String,

trim: true

},

name: {

type: String,

trim: true

}

}, {

timestamps: true

})

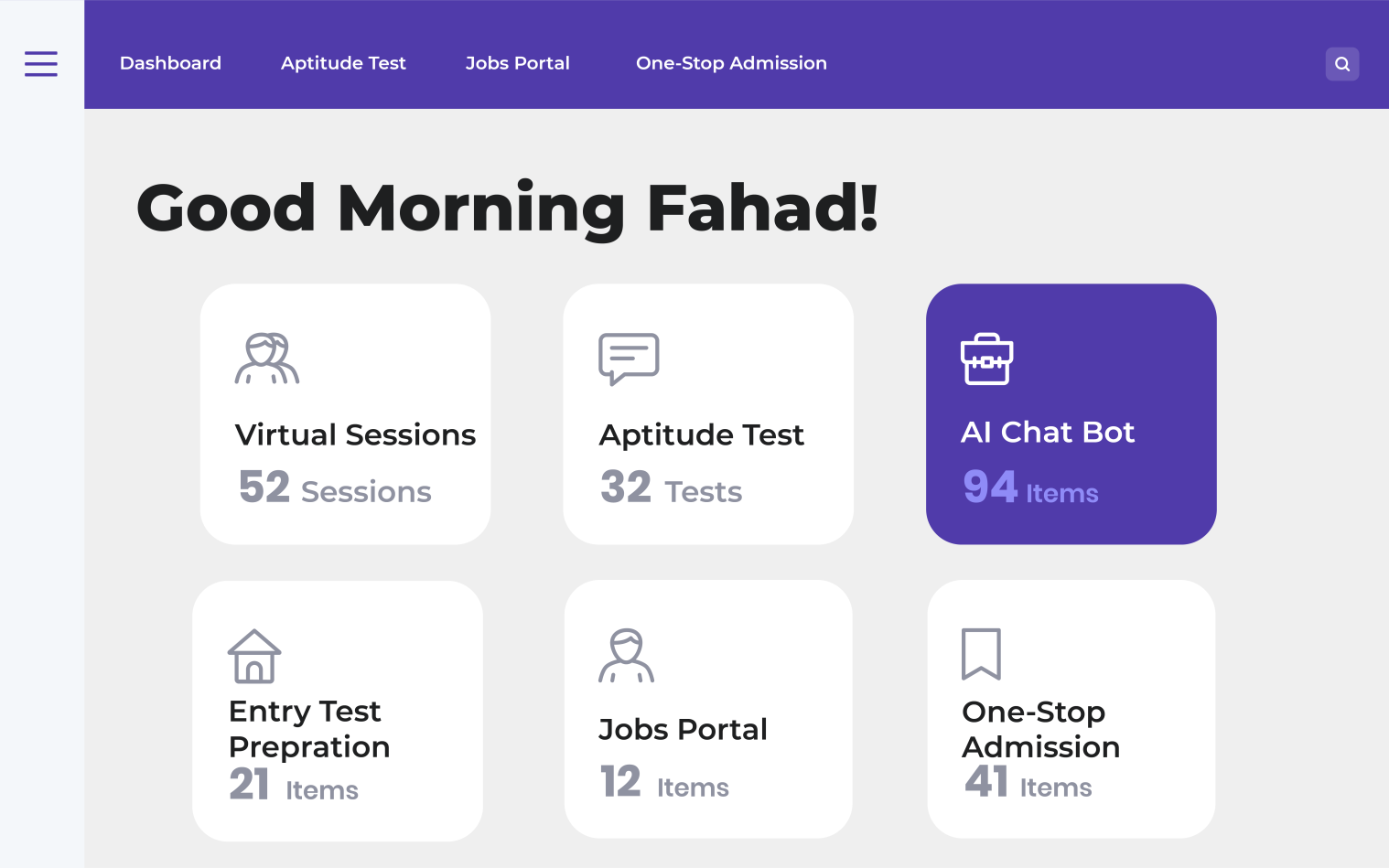
# Data Dictionary

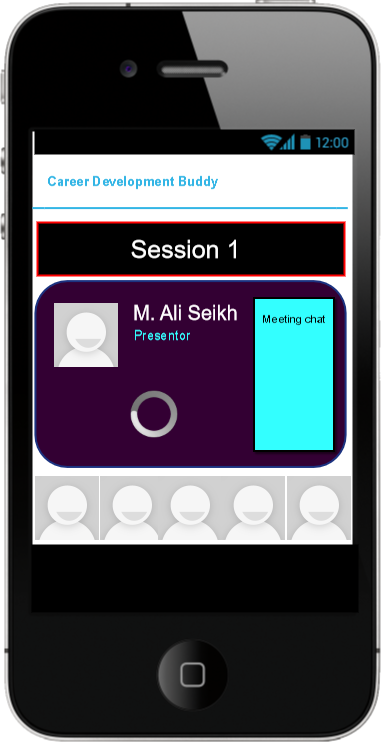
|  |  |  |  |
| --- | --- | --- | --- |
| **Collections** | **Property** | **Data Type** | **Description** |
| Users | Full name | string | Name of the user |
| Users | email | string | Email of user |
| Users | Role | int | Role of user |
| Users | Password | string | Password of user |
| Users | image | string | image of user |
| Videos | video | string | Video url |
| Videos | name | string | Video title |
| Test | question | string | Question |
| Test | university | String | University name |
| Test | A | String | A option |
| Test | B | String | B option |
| Test | C | String | C option |
| Test | D | String | D option |
| Test | Correct option | String | Correct option |
| PDF | pdf | String | Pdf file |
| Paper | university | String | University name for the paper |
| Paper | Year | String | Paper required year |
| Paper | Pdf | String | Pdf doc url |
| Discussion | Query | String | Discussion query |
| Discussion | Response | string | Discussion response |

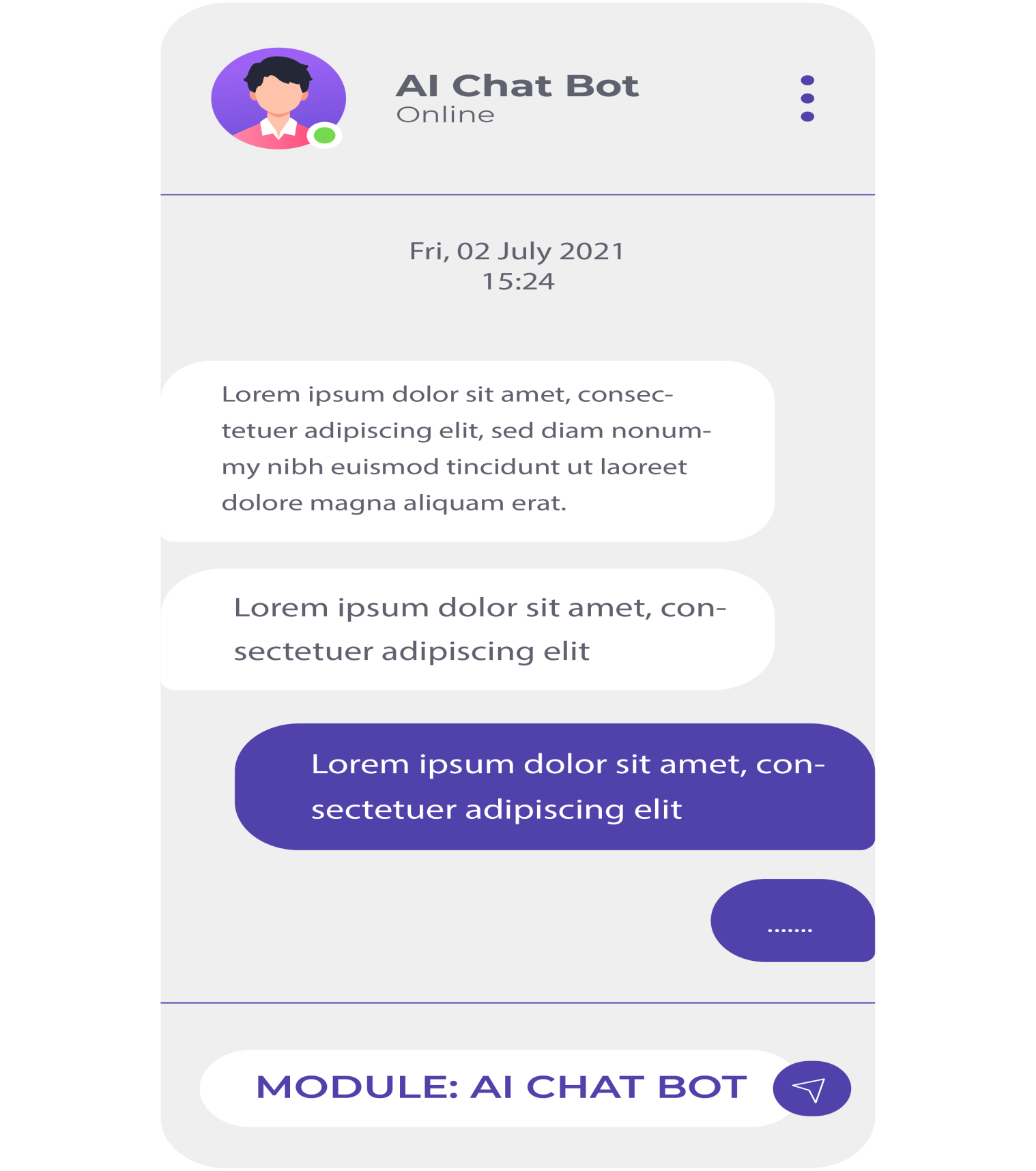
# Human Interface Design

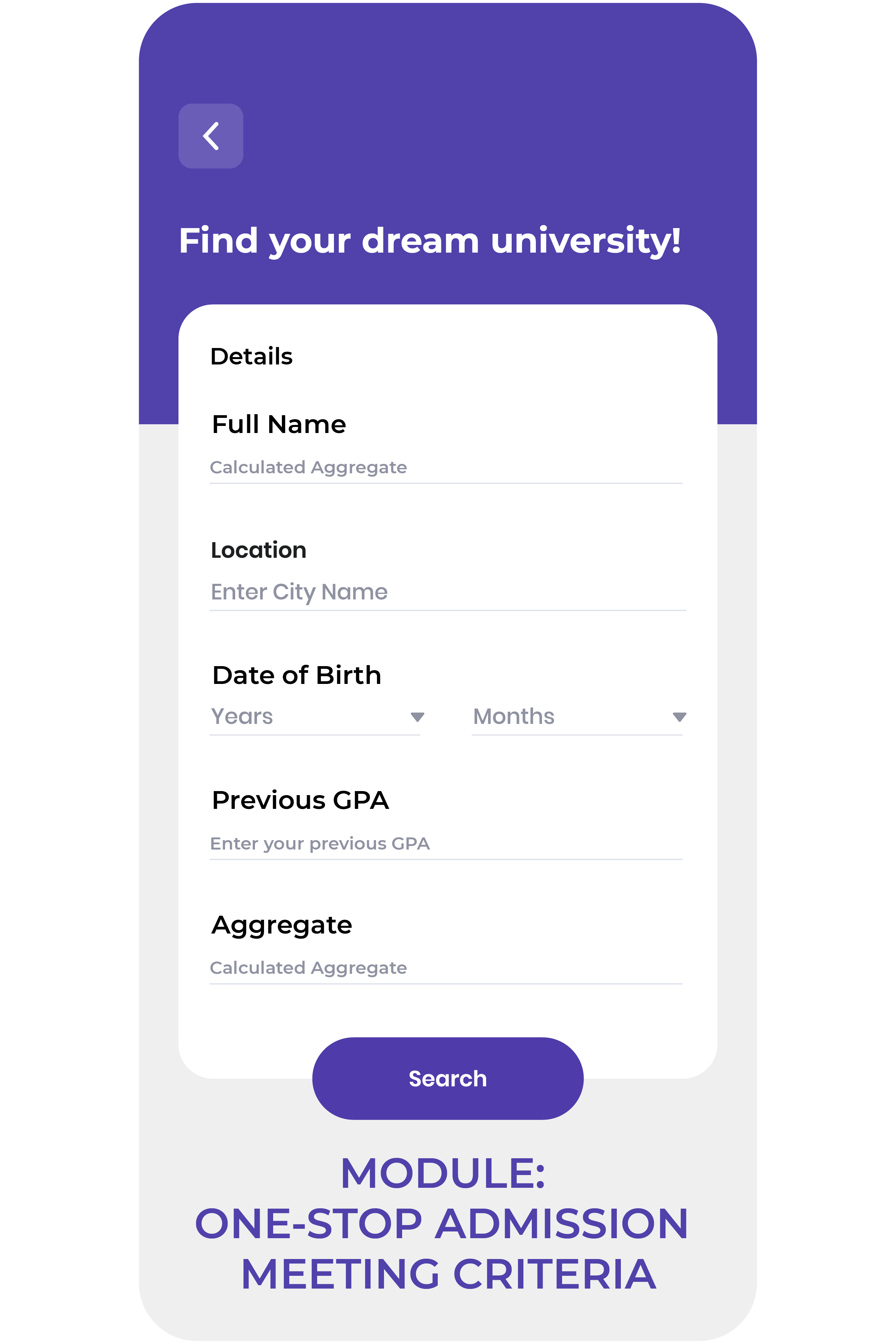
Interfaces displayed to the user are shown below as mockups.

# Screen Images









# Implementation

Implementation of the proposed system along with the API`s used and algorithms of the core functionalities are discussed below.

.



## Algorithm

|  |
| --- |
| **Algorithm 1: Chabot** |
| **Input:** User inputs a statement |
| **Output:** Chabot gives the reply |
| * Start by receiving the user's input or query regarding their career aspirations, interests, skills, or any specific questions they have. * Construct a prompt that includes the user's input and provides context for the OpenAI API. * Send the prompt to the OpenAI API and receive the generated response by using "openai.ChatCompletion.create()" method. * Extract the bot's response from the API's generated output. This could be done by retrieving the last message from the API's response. * Present the bot's response to the user in a chat-like format, making it easy to read and understand. * Allow the user to continue the conversation by providing additional input or follow-up questions. Loop back to step 1 to collect the new input. * Implement logic to identify specific user queries related to career counseling by using keyword matching to recognize keywords or intents. * Based on the user's queries, generate tailored responses using the OpenAI API. |

|  |
| --- |
| **Algorithm 2: Recommendation System** |
| **Input:** User's preferences |
| **Output:** Recommend universities to users based on their input |
| * Import the necessary libraries: pandas, numpy, Flask, and flask\_cors. * Read and preprocess the data from the CSV file into a pandas DataFrame (df) by dropping columns containing the word "unnamed". * Define the Euclidean distance function (euclidean\_dist) to calculate the distance between two data instances. * Implement the k-nearest neighbors algorithm (knn):  1. Take the training set, test instance, and value of k as input. 2. Calculate the distances between the test instance and each instance in the training set using euclidean\_dist. 3. Sort the distances and select the indices of the k nearest neighbors. 4. Count the occurrences of each class label among the neighbors. 5. Sort the class labels based on their counts and return them along with the neighbor indices.  * Extract the input features (matric marks,fsc marks, prefered field, location) from the JSON request body. * Create a test DataFrame (testSet) using the input features. * Set the value of k. * Apply the KNN algorithm (knn) to the test set and store the results in ‘result’ and ‘neigh’. * Extract the class labels from result and store them. * Return a JSON response containing the predicted universities (universities) and the weights (indices of the nearest neighbors). |
|  |

## External APIs/SDKs

APIs/SDKs used in our project implementation are show in the following table.

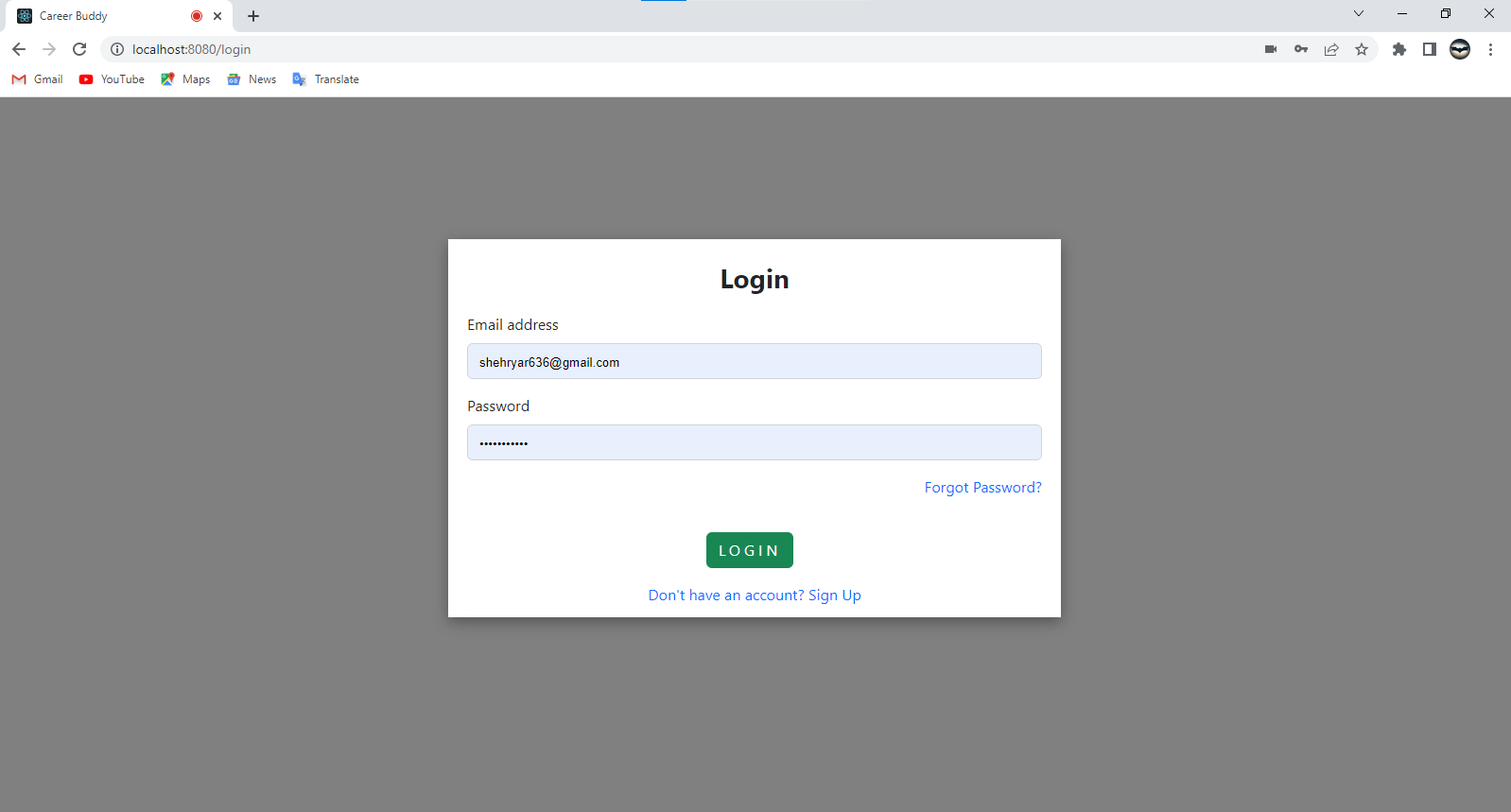
**Table 5:** Details of APIs used in the project

|  |  |  |
| --- | --- | --- |
| **Name of API and version** | **Description of API** | **Purpose of usage** |
| Rest API | It conforms to the constraints of REST architectural style and allows for interaction with RESTful web services. | RestAPIs are used to interact between front end and backend. They are responsible for data transfer. |
| WebRTC API | WebRTC enables real time communication over peer to peer connection. | Web Sockets are responsible for chatting. Their purpose is to facilitate audio and video call functionality |
|  |  |  |

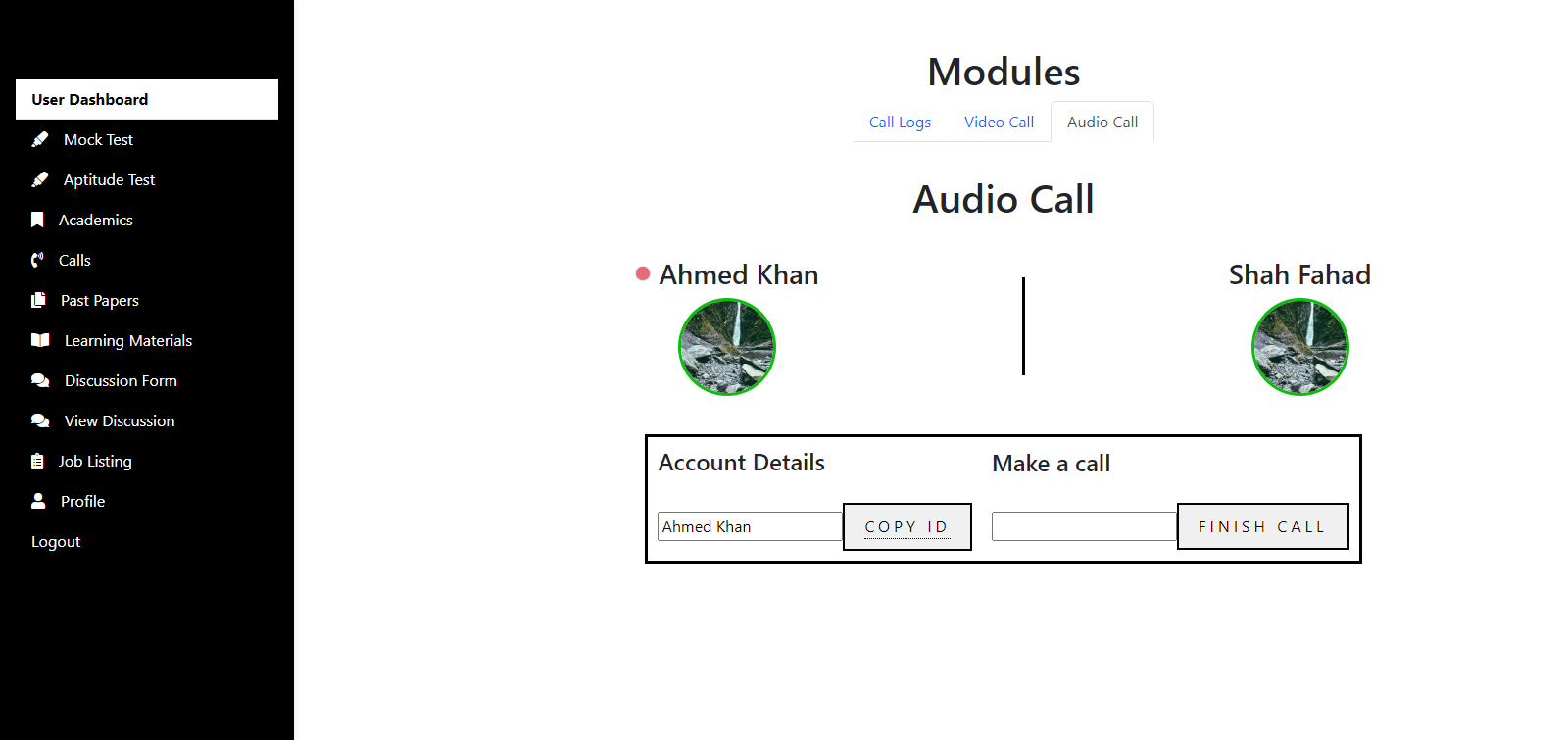
## User Interface

UI of the proposed system is shown below:

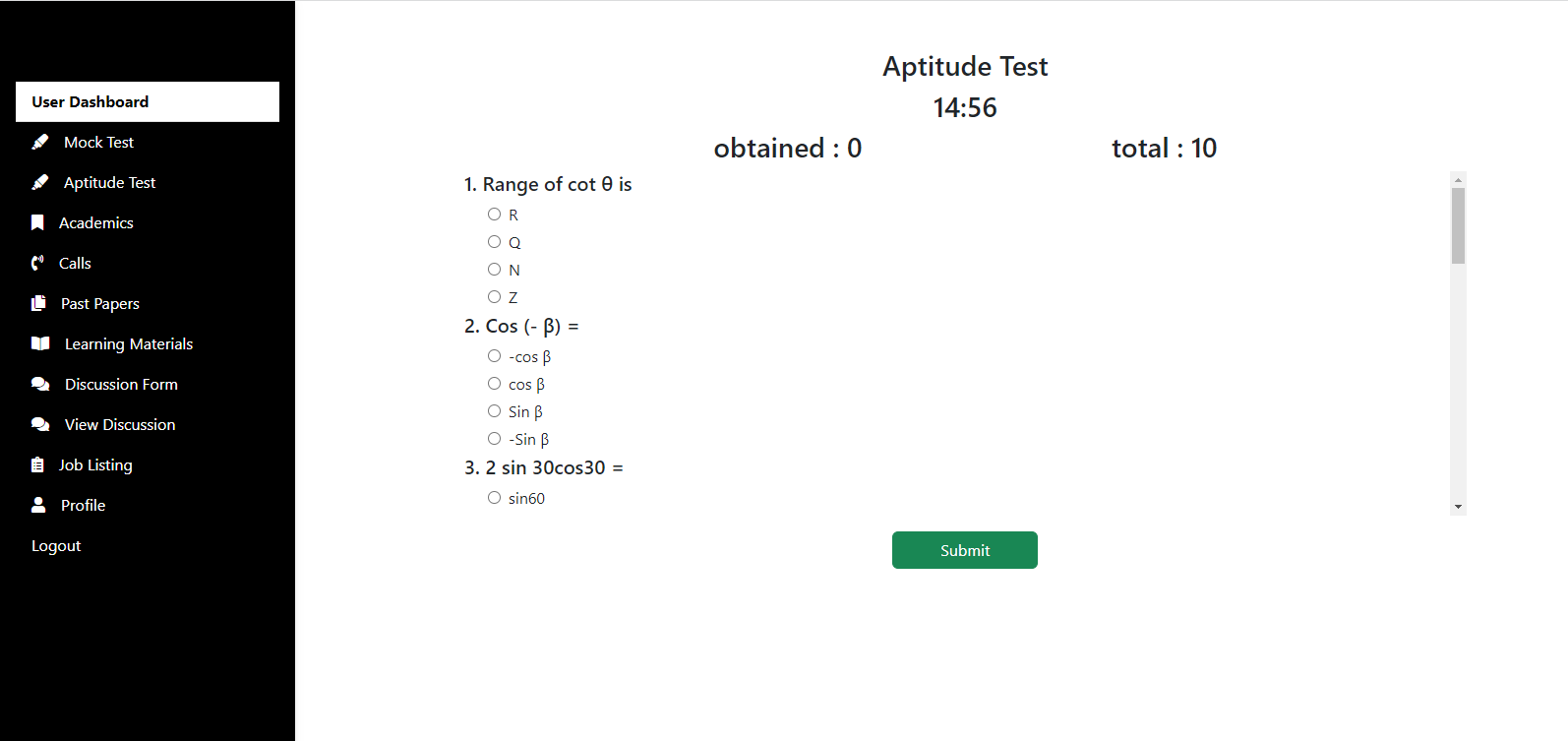
### Login Screen

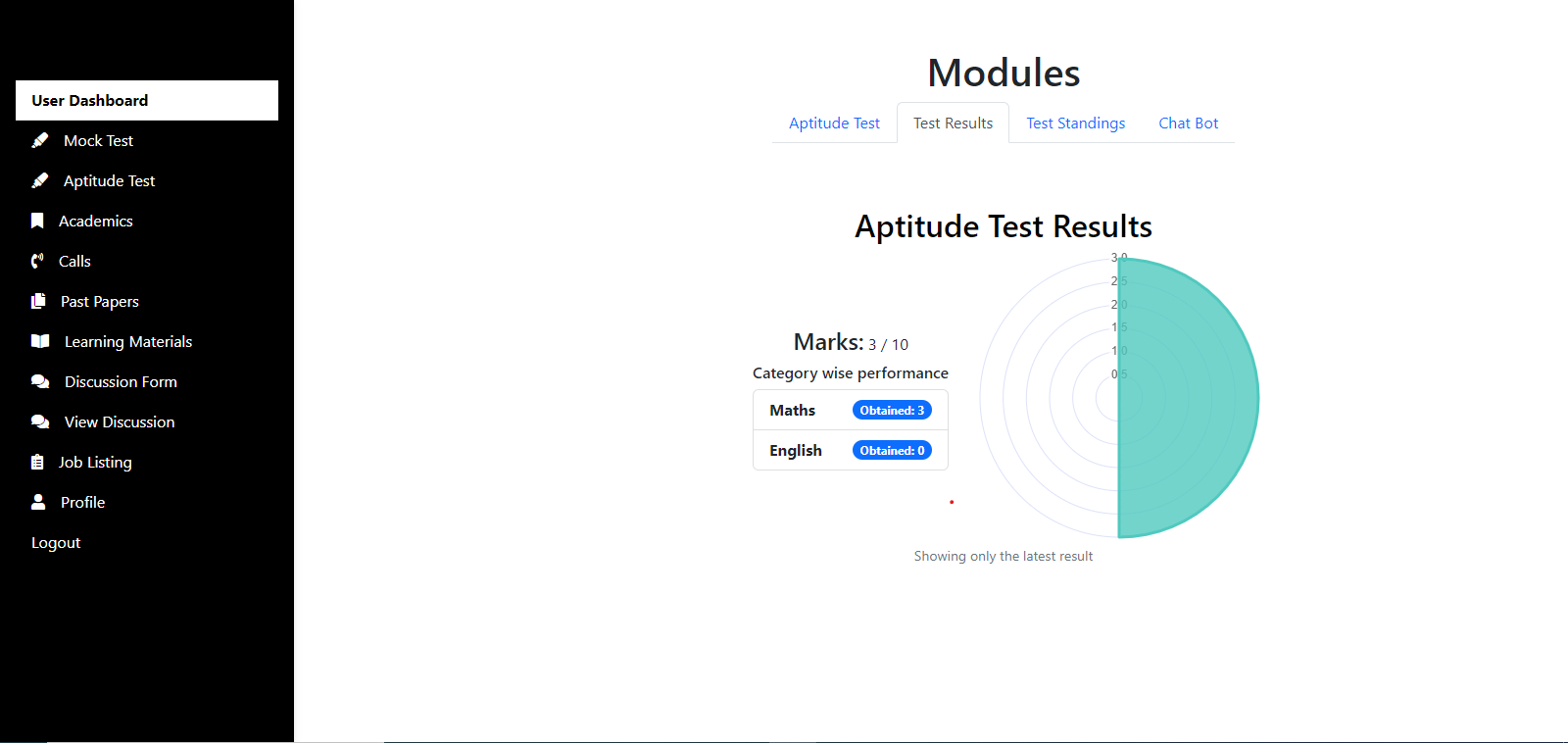


### Virtual Session

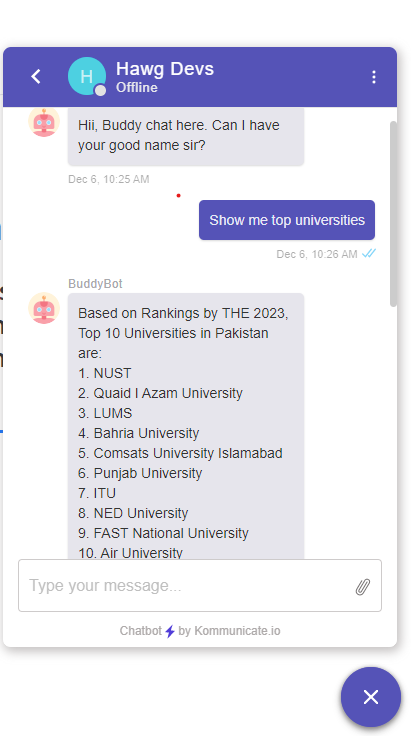


### Aptitude test

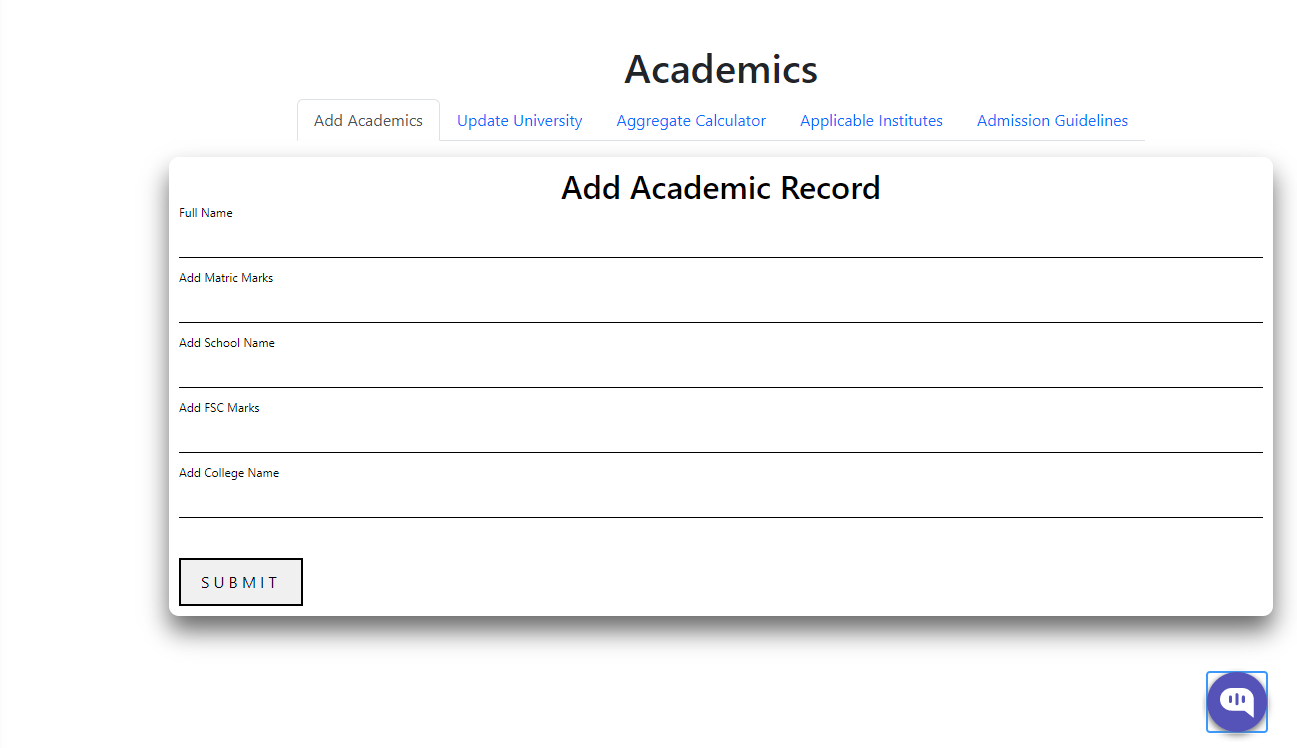




### Chatbot



### One-Stop admission check



## Deployment

Environment conditions used currently are local host; project shall be deployed later and main features are tested internally.

# Testing and Evaluation



Types of testing done along with respective test cases are discussed below in detail.

.

## Unit Testing

It’s a level of software testing where individual units of a software/component are tested. The purpose is to validate that each unit of the software performs as designed.

**Unit Testing 1:** Enter email

**Testing Objective:** To ensure a valid email is entered.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No.** | **Test case/Test script** | **Attribute and value** | **Expected result** | **Result** |
| 1 | Enter correct email to check if is being validated. | Email: shehryar@gmail.com | Email valid. | Pass |
| 2 | Enter incorrect email to check if error message is shown. | Email: daemon | Invalid email | Pass |

**Unit Testing 2:** Enter password

**Testing Objective:** To ensure a valid password is entered.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No.** | **Test case/Test script** | **Attribute and value** | **Expected result** | **Result** |
| 1 | Enter correct password to check if is being validated. | password: shehryar123 | Valid password. | Pass |
| 2 | Enter incorrect password to check if error message is shown. | Password: 123 | Invalid password (Should be 8 characters long). | Pass |

**Unit Testing 3:** Forget password

**Testing Objective:** To ensure the user new password gets updated.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No.** | **Test case/Test script** | **Attribute and value** | **Expected result** | **Result** |
| 1 | Enter new password and verify if it’s updated or not. | New password: shehryar123 | Password gets updated. | Pass |

**Unit Testing 4:** Update profile picture

**Testing Objective:** To ensure the profile picture gets updated for the respected user.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No.** | **Test case/Test script** | **Attribute and value** | **Expected result** | **Result** |
| 1 | Upload new profile picture and check whether its updated or not. | New profile pic: test.png | Profile picture successfully updated. | Pass |
| 2 | Upload invalid file and check whether it’s uploaded or not. | New profile pic: abc.txt | Invalid file format. | Pass |

**Unit Testing 5:** Logout

**Testing Objective:** To ensure user log out of the system.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No.** | **Test case/Test script** | **Attribute and value** | **Expected result** | **Result** |
| 1 | Check user logs out from the system by clicking logout button. | Confirm logout?  Yes | User logged out. | Pass |

**Unit Testing 6:** Allow mic access.

**Testing Objective:** To ensure mic access is provided to the system.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No.** | **Test case/Test script** | **Attribute and value** | **Expected result** | **Result** |
| 1 | Upon the message prompting mic access, allow the system to access it to check its working. | Allow mic access?  Yes | Functional Mic access provided to the system. | Pass |

**Unit Testing 7:** Allow camera access.

**Testing Objective:** To ensure camera access is provided to the system.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No.** | **Test case/Test script** | **Attribute and value** | **Expected result** | **Result** |
| 1 | Upon the message prompting camera access, allow the system to access it to check it’s working. | Allow camera access?  Yes | Functional camera access provided to the system. | Pass |

**Unit Testing 8:** Generate call ID.

**Testing Objective:** To ensure call ID gets generated when user place a call.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No.** | **Test case/Test script** | **Attribute and value** | **Expected result** | **Result** |
| 1 | Upon placing the call check whether the call ID is generated or not. | Call ID:  234GFU1 | Call ID gets generated. | Pass |

**Unit Testing 9:** Connect to call.

**Testing Objective:** To ensure second user connects to call via call ID.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No.** | **Test case/Test script** | **Attribute and value** | **Expected result** | **Result** |
| 1 | Check user connects to call via call ID. | Join Call ID:  234GFU1 | User connected to call successfully. | Pass |

**Unit Testing 10:** Finish call.

**Testing Objective:** To ensure call gets ended.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No.** | **Test case/Test script** | **Attribute and value** | **Expected result** | **Result** |
| 1 | Check call gets ended when clicked on end call button. | End call:  True | Call successfully ended. | Pass |

**Unit Testing 11:** Receive call.

**Testing Objective:** To ensure call gets received when clicked on receive call button.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No.** | **Test case/Test script** | **Attribute and value** | **Expected result** | **Result** |
| 1 | Check call gets received when clicked on receive call button. | Receive incoming call:  True | Call successfully received. | Pass |

**Unit Testing 12:** Show Call logs

**Testing Objective:** To ensure updated call logs gets displayed when clicked on view call logs option.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No.** | **Test case/Test script** | **Attribute and value** | **Expected result** | **Result** |
| 1 | Check user call logs gets updated in database after making a new call. | ViewCallLogs:  true | Updated call logs displayed with following new values in database:  Call ID: 234GFU1  Time: 11:50 pm  Date: 12/6/2022  Duration: 00:03:20 | Pass |

**Unit Testing 13:** Begin Aptitude Test.

**Testing Objective:** To ensure aptitude test gets started when clicked on begin test button.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No.** | **Test case/Test script** | **Attribute and value** | **Expected result** | **Result** |
| 1 | Check aptitude test gets started when clicked on begin test button. | Begin Test:  True | Test successfully started and displayed. | Pass |

**Unit Testing 14:** Attempt Aptitude Test.

**Testing Objective:** To ensure aptitude test gets started when clicked on begin test button.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No.** | **Test case/Test script** | **Attribute and value** | **Expected result** | **Result** |
| 1 | Check whether aptitude test is functional by attempting given mcqs. | Which figure from the options on the right could occupy the highlighted square to complete the sequence?  B | Incorrect. | Pass |
| 2 | Check whether aptitude test is functional by attempting given mcqs. | What is the average of first five multiples of 12?  C | Correct. | Pass |

**Unit Testing 15:** Show Test results.

**Testing Objective:** To ensure aptitude test results gets displayed after attempting the test when clicked on show results button test button.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No.** | **Test case/Test script** | **Attribute and value** | **Expected result** | **Result** |
| 1 | Check aptitude test results gets displayed after attempting the test when clicked on show results button test button. | Show Test results:  True | Test results displayed. | Pass |

**Unit Testing 16:** View University details.

**Testing Objective:** To ensure university detail gets displayed when user selects a university.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No.** | **Test case/Test script** | **Attribute and value** | **Expected result** | **Result** |
| 1 | Check university detail gets displayed when user selects university. | University name:  Comsats | Details get displayed. | Pass |

**Unit Testing 17:** Compare University.

**Testing Objective:** To ensure selected universities detail gets compared and displayed.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No.** | **Test case/Test script** | **Attribute and value** | **Expected result** | **Result** |
| 1 | Check selected universities detail gets compared and displayed when clicked on compare university. | First University name:  Comsats  Second University name:  NUST | University statistics (courses offered, campuses, enrollment ratios etc) get compared. | Pass |

**Unit Testing 18:** University Recommendation.

**Testing Objective:** To ensure user gets recommended a suitable university when clicked on university recommendation tab.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No.** | **Test case/Test script** | **Attribute and value** | **Expected result** | **Result** |
| 1 | Check user gets recommended a suitable university after he enters his marks obtained and desired tuition fee when clicked on university recommendation tab. | Marks obtained:  762  Tuition fee:  70000 | Suitable list of universities recommended corresponding to the user entered parameters. | Pass |

**Unit Testing 19:** Upload learning material (pdf).

**Testing Objective:** To ensure pdf file gets uploaded for entry test learning material.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No.** | **Test case/Test script** | **Attribute and value** | **Expected result** | **Result** |
| 1 | Ensures pdf file gets uploaded from computer. | pdf: test.txt | Pdf gets uploaded. | Pass |
| 2 | Ensures error message is displayed if incorrect file gets uploaded from computer. | file: abc.mp4 | Invalid file. | Pass |
| 3 | Ensures video link gets uploaded. | Video link: https://www.youtube.com/watch?v=Q33KBiDriJY | Video gets uploaded. | Pass |

**Unit Testing 20:** Add mock test.

**Testing Objective:** To ensure mock test file gets uploaded by Admin.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No.** | **Test case/Test script** | **Attribute and value** | **Expected result** | **Result** |
| 1 | Ensures admin added mock test gets displayed to user. | pdf: mock.txt | File gets added and displayed to user. | Pass |

**Unit Testing 21:** Discussion form.

**Testing Objective:** To ensure user added queries gets added and displayed to other users in discussion form.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No.** | **Test case/Test script** | **Attribute and value** | **Expected result** | **Result** |
| 1 | Ensures user added query gets displayed to other user on the discussion form. | Question:  What is ReactJS | Question shown on the discussion form. | Pass |
| 2 | Ensures error message is shown when user inputs nothing and clicks submit on the discussion form. | Question:  NULL | Error! Please enter a question. | Pass |

**Unit Testing 22:** Past papers.

**Testing Objective:** To ensure past papers gets added and displayed.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No.** | **Test case/Test script** | **Attribute and value** | **Expected result** | **Result** |
| 1 | Ensures admin added past paper gets displayed to the user. | Paper:  FAST.txt  Year:  2019 | Paper added and displayed to other users. | Pass |
| 2 | Ensures admin added past paper gets displayed to the user. | Paper:  Comsats.txt  Year:  2021 | Paper added and displayed to other users. | Pass |
| 3 | Ensures 2020 past papers get filtered and displayed when clicked on filter past papers by year (2020) | Year:  2019 | Paper: FAST  Above paper gets displayed. | Pass |

**Unit Testing 23:** Ask Chatbot query.

**Testing Objective:** To ensure chatbot responds to the posed query.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No.** | **Test case/Test script** | **Attribute and value** | **Expected result** | **Result** |
| 1 | Ensures ensure chatbot responds to the posed query. | Query:  Hi | Answer:  Hello, How can I help you? | Pass |
| 2 | Ensures ensure chatbot responds to the posed query. | Query:  Show me top educational institutes. | Answer:  According to 2020 HEC ranking:   1. NUST 2. FAST.NU 3. LUMS 4. COMSATS 5. PIEAS 6. IBA 7. UOL | Pass |

**Unit Testing 24:** Academic record form

**Testing Objective:** To ensure academic record form is working correctly.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No.** | **Test case/Test script** | **Attribute and value** | **Expected result** | **Result** |
| 1 | Ensures Matric marks gets added. | SSC marks: 1000 | SSC marks get added. | Pass |
| 2 | Ensures FSC marks gets added. | HSSC marks: 950 | HSSC marks gets added. | Pass |
| 3 | Ensures incorrect entry in Matric marks field pops error message. | SSC marks: abc | Invalid input. | Pass |
| 4 | Ensures Matric marks gets updated. | SSC marks: 985 | Marks gets updated. | Pass |
| 5 | Ensures FSC marks gets updated. | HSSC marks: 1000 | Marks gets updated. | Pass |

**Unit Testing 25:** Calculate aggregate

**Testing Objective:** To ensure aggregate is being calculated correctly.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No.** | **Test case/Test script** | **Attribute and value** | **Expected result** | **Result** |
| 1 | Ensures Matric marks gets added. | SSC marks: 1000 | SSC marks get added. | Pass |
| 2 | Ensures FSC marks gets added. | HSSC marks: 950 | HSSC marks gets added. | Pass |
| 3 | Ensures entry marks gets added. | Entry test marks: 735 | Entry test marks gets added. | Pass |
| 4 | Select university for which aggregate is to be calculated. | University name: Comsats | University name added to database. | Pass |
| 5 | Ensures aggregate is being calculated correctly. | Find aggregate:  true | 81.05% | Pass |

**Unit Testing 26:** Display applicable institutes.

**Testing Objective:** To ensure applicable institutes are being displayed to the user.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No.** | **Test case/Test script** | **Attribute and value** | **Expected result** | **Result** |
| 1 | Ensures applicable institutes are being displayed to the user after he enters his aggregate and desired field. | Aggregate:  80.75%  Field:  Computer science | Applicable university names get displayed i.e.   * Comsats * PUCIT * IBA * FAST | Pass |

**Unit Testing 27:** Manage job listing.

**Testing Objective:** To ensure job listing gets created and deleted.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No.** | **Test case/Test script** | **Attribute and value** | **Expected result** | **Result** |
| 1 | Ensures job listing gets created by employer. | Job title: Software engineer  Location: Lahore  Salary: 50,000  Experience: 1-2 years  Description: abc | Listing gets added. | Pass |
| 2 | Ensures job listing gets created by employer. | Job title: Data scientist  Location: Islamabad  Salary: 60,000  Experience: 1-2 years  Description: abc | Listing gets added. | Pass |
| 3 | Ensures job listing gets updated by employer. | Job title: Software engineer  Location: Lahore  Salary: 70,000  Experience: 2-3 years  Description: xyz | Listing gets updated. | Pass |
| 4 | Ensures job listing gets deleted by employer by entering job title name. | Job title: Software engineer | Listing gets deleted. | Pass |

**Unit Testing 28:** Filter job listing.

**Testing Objective:** To ensure job listing gets filtered by student.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No.** | **Test case/Test script** | **Attribute and value** | **Expected result** | **Result** |
| 1 | Ensures job listing gets filtered by location. | Location: Islamabad | Listing gets filtered and displayed (Data scientist). | Pass |
| 2 | Ensures job listing gets filtered by title. | Job title: Software engineer | Listing gets filtered and displayed (Software engineer). | Pass |

**Unit Testing 29:** Search job listing.

**Testing Objective:** To ensure job listing gets searched ad displayed.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No.** | **Test case/Test script** | **Attribute and value** | **Expected result** | **Result** |
| 1 | Ensures job listing gets searched. | Job title: Software engineer. | Listing gets displayed. | Pass |

**Unit Testing 31:** Upload attachments.

**Testing Objective:** To ensure user attachments gets uploaded.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No.** | **Test case/Test script** | **Attribute and value** | **Expected result** | **Result** |
| 1 | Ensures CV gets uploaded. | File:  Resume.png | CV gets uploaded. | Pass |
| 2 | Ensures matric transcript gets uploaded. | File:  abc.png | Matric transcript gets uploaded. | Pass |
| 3 | Ensures error message pops up if invalid file gets uploaded. | File:  xyz.mp4 | Error! Please try again. | Pass |

**Unit Testing 32:** Upload attachments.

**Testing Objective:** To ensure user attachments gets uploaded.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No.** | **Test case/Test script** | **Attribute and value** | **Expected result** | **Result** |
| 1 | Ensures CV gets uploaded. | File:  Resume.png | CV gets uploaded. | Pass |
| 2 | Ensures matric transcript gets uploaded. | File:  abc.png | Matric transcript gets uploaded. | Pass |
| 3 | Ensures error message pops up if invalid file gets uploaded. | File:  xyz.mp4 | Error! Please try again. | Pass |

## Unit Testing 33: Job applies.

**Testing Objective:** To ensure user applies for the job successfully

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No.** | **Test case/Test script** | **Attribute and value** | **Expected result** | **Result** |
| 1 | Ensures user applies for job successfully after selecting from job listings and clicking on apply button. | Job Listing:  Data scientist  Apply: true | Job application successful. | Pass |

## Functional Testing

The functional testing will take place after the unit testing. In this functional testing, the functionality of each of the module is tested. This is to ensure that the system produced meets the specifications and requirements.

**Functional Testing 1:** Login with correct and incorrect credentials

**Objective**: To ensure that user is logged in with correct credentials.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **No.** | **Test case/Test script** | **Attribute and value** | **Expected result** | **Actual result** | **Result** |
| 1. | Login with correct email and password. | username:  [shehrya636r@gmail.com](mailto:shehrya636r@gmail.com)  Password:  shehryar | Log in successful | Log in successful. | Pass |
| 2. | Login with incorrect email and password. | username:  [shehrya02r@gmail.com](mailto:shehrya02r@gmail.com)  Password:  !awe | Log in unsuccessful | Log in unsuccessful. | Pass |

**Functional Testing 2:** Login as different roles.

**Objective**: To ensure that user is logged in with correct different roles.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **No.** | **Test case/Test script** | **Attribute and value** | **Expected result** | **Actual result** | **Result** |
| 1. | Login as a “Admin” | username:  [shehrya636@gmail.com](mailto:shehrya636@gmail.com)  Password:  shehryar | Logged in Admin. | Logged in as Admin. | Pass |
| 2. | Login as a “Student” | username:  [fahad123@gmail.com](mailto:fahad123@gmail.com)  Password:  fahad123 | Logged in as student | Logged in as student. | Pass |
| 3. | Login as a “Counselor” | username:  [hamza101@gmail.com](mailto:hamza101@gmail.com)  Password:  hamza | Logged in counselor. | Logged in as counselor. | Pass |

**Functional Testing 3:** Sign-up

**Objective**: To ensure that user account is created.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **No.** | **Test case/Test script** | **Attribute and value** | **Expected result** | **Actual result** | **Result** |
| 1. | Ensures user account is created with all correct credentials. | Full name:  Shehryar Afgan  Username:  Shehryar636@gmail.com  Password:  shehryar  Confirm password:  shehryar | user account is created. | user account is created. | Pass |
| 2. | Ensures user account is not created due to incorrect username. | Full name:  Fahad Ali  Username:  fahadgmail  Password:  fahad  Confirm password:  fahad | Incorrect username! | Incorrect username! | Pass |
| 3. | Ensures user account is not created due to incorrect re-entry of password. | Full name:  Fahad Ali  Username:  Fahad636@gmail.com  Password:  fahad  Confirm password:  fah | Error! Please reenter you above password. | Error! Please reenter you above password. | Pass |

**Functional Testing 4:** Video Call

**Objective**: To ensure that video call works properly.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **No.** | **Test case/Test script** | **Attribute and value** | **Expected result** | **Actual result** | **Result** |
| 1. | Ensure user “Shehryar” start video call and call ID is generated. | Caller name:  Shehryar Afgan  Call ID:  246GHY1  Camera access?  allowed | Call started and call ID generated. | Call started and call ID generated. | Pass |
| 2. | Ensure user “Fahad” joins video call by call ID that was shared by user “Shehryar”. | Call ID:  246GHY1  Camera access?  allowed | Call started and call ID generated. | Call started and call ID generated. | Pass |
| 3. | Ensure call logs are displayed after call has ended. | End call?  Yes  View Call Logs:  true | Call logs displayed:  Caller name: Shehryar Afgan  Call ID: 246GHY1  Receiver:  Fahad Khan  Duration:  00:05:15  Date:  12/1/2022  Time:  4:47pm | Call logs displayed:  Caller name: Shehryar Afgan  Call ID: 246GHY1  Receiver:  Fahad Khan  Duration:  00:05:15  Date:  12/1/2022  Time:  4:47pm |  |

**Functional Testing 5:** Aptitude Test

**Objective**: To ensure that user after user takes aptitude test correct score is being displayed along with his previous standings.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **No.** | **Test case/Test script** | **Attribute and value** | **Expected result** | **Actual result** | **Result** |
| 1. | Ensure that test for user gets started when he clicks on begin test. | Begin Test:  True | Test started. | Test started. | Pass |
| 2. | Ensure that test results get displayed after user attempts test. | Begin Test:  True  What is meant by inheritance?  A  Which one of the following is not correct?  C  What is the use of Redux?  D | Total score: 3/10 | Total score: 3/10 | Pass |
| 3. | Ensure that test standings get displayed when user selects view test standings. | View test standings:  true | Attempted by:  Shehryar Afgan  Date:  5/7/2022  Score: 5/10  Attempted by:  Fahad Khan  Date:  5/7/2022  Score: 2/10  Attempted by:  Hamza Ali  Date:  7/7/2022  Score: 7/10 | Attempted by:  Shehryar Afgan  Date:  5/7/2022  Score: 5/10  Attempted by:  Fahad Khan  Date:  5/7/2022  Score: 2/10  Attempted by:  Hamza Ali  Date:  7/7/2022  Score: 7/10 | Pass |

**Functional Testing 6:** University Recommendation

**Objective**: To ensure that suitable institute is recommended to the user.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **No.** | **Test case/Test script** | **Attribute and value** | **Expected result** | **Actual result** | **Result** |
| 1. | Ensure that suitable institute is recommended to the user when he enters his marks obtained and desired tuition fee. | Marks obtained:  270/500  Tuition fee:  45000 | Institute:  Iqra University | Institute:  Iqra University | Pass |

**Functional Testing 7:** Entry test preparation

**Objective**: To ensure that the learning materials along with past papers are being uploaded and shown to the user.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **No.** | **Test case/Test script** | **Attribute and value** | **Expected result** | **Actual result** | **Result** |
| 1. | Ensure that mock test starts when user selects mock test option. | Institute:  NUST  Begin Test:  True | Mock test gets started for the user to attempt. | Mock test gets started for the user to attempt. | Pass |
| 2. | Ensure that learning material gets added when user uploads files and click on add option. | File:  abc.txt  Video url:  https://www.youtube.com/watch?v=LiSSF | File and Video added | File and Video added | Pass |
| 3. | Ensure that learning material gets added when user uploads files and click on add option. | File:  abc.mp4  Video url:  https://www.youtube.com/watch?v=LiSSF | Incorrect file format. Please try again! | Incorrect file format. Please try again! | Pass |
| 4. | Ensure that comsats 2019 past paper gets displayed when user selects view past paper. | University:  Comsats  Year:  2019 | Comsats 2019 past paper displayed. | Comsats 2019 past paper displayed. | Pass |

**Functional Testing 8:** AI chatbot

**Objective**: To ensure chatbot is working properly and answering user queries with appropriate responses.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **No.** | **Test case/Test script** | **Attribute and value** | **Expected result** | **Actual result** | **Result** |
| 1. | Ensure that chatbot answers user queries with appropriate responses. | Question:  Hi  Question:  Show me top careers in Pakistan | Answer:  Hello; Pleasure to meet you  Answer:  Some of the top careers in Pakistan are:   * Charted Accountant * IT * Banking * Doctor * Army | Answer:  Hello; Pleasure to meet you  Answer:  Some of the top careers in Pakistan are:   * Charted Accountant * IT * Banking * Doctor   Army | Pass |

**Functional Testing 9:** One stop admission check

**Objective**: To ensure correct academic record is stored in database and corresponding aggregate is being calculated properly.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **No.** | **Test case/Test script** | **Attribute and value** | **Expected result** | **Actual result** | **Result** |
| 1. | Ensure that appropriate academic record is being added. | SSC marks:  950  HSSC marks:  1000  School name:  Silver Oaks school  Board:  Federal  College name:  Fauji Foundation | Details added. | Details added. | Pass |
| 2. | Ensure that appropriate academic record is being added. | SSC marks:  950  HSSC marks:  empty  School name:  XYZ  Board:  Federal  College name:  Fauji Foundation | No such school in federal board found! Please try again. | No such school in federal board found! Please try again. | Pass |
| 3. | Ensure that correct aggregate is being calculated. | Entry test marks:  600 | Aggregate: 75% | Aggregate: 75% | Pass |

**Functional Testing 10:** Apply for Job

**Objective**: To ensure user as a student can successfully apply for a job listing that an Employer has created.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **No.** | **Test case/Test script** | **Attribute and value** | **Expected result** | **Actual result** | **Result** |
| 1. | Ensure that job is created by the user as a “Employer”. | Actor:  Employer  Job title:  Data Scientist  Company:  ABC corp  Salary:  50000  Location:  Islamabad  Description:  abc | Job Listing successfully created. | Job Listing successfully created. | Pass |
| 2. | Ensure that job is not created by the user as a “Student”. | Actor:  Student  Job title:  Data Scientist  Company:  ABC corp  Salary:  50000  Location:  Islamabad  Description:  abc | Error! Invalid actor (Only employer can create listing). | Error! Invalid actor (Only employer can create listing). | Pass |
| 3. | Ensure that job is applied by the user as a “Student”. | Actor:  Student  Job title:  Data Scientist | Job Successfully applied. | Job Successfully applied. | Pass |
| 4. | Ensure that job is applied by the user as a “Student”. | Actor:  Student  Job title:  IT associate | No such Job listing found. | No such Job listing found. | Pass |

## Integration Testing

Integration tests assess whether a set of classes that must work together do so without error. They

Ensure that the interfaces and linkages between different parts of the system work properly. At this point, the classes have passed their individual unit tests, so the focus now is on the flow of control among the classes and on the data exchanged among them. Integration testing follows the same general procedures as unit testing: The tester develops a test plan that has a series of tests, which, in turn, have a test. Integration testing is often done by a set of programmers and/or systems analysts.

**Integration Testing 1:** Job Portal

**Testing Objective:** To ensure the jobs associated functions is being done correctly by employer and student/applicant` and *the* ***interface*** *between* module ‘User profiling’ and module ‘Job portal’ *is running correctly*.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **No.** | **Test case/Test script** | **Attribute and value** | **Expected result** | **Actual result** | **Result** |
| 1. | Apply for Job (student) | Job Name:  UI designer | Student successfully applies for the respected job. | Student successfully applies for the respected job. | Pass |
| 2. | Add Job Listing (employer) | Job Name:  UI designer  Salary:  60000  Location:  Islamabad | Job Listing added successfully. | Job Listing added successfully. | Pass |

**Integration Testing 2:** Entry Test Preparation

**Testing Objective:** To ensure the entry test materials are being successfully added successfully by the “admin” and *the* ***interface*** *between* module ‘User profiling’ and module ‘Entry Test Preparation’ *is running correctly*.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **No.** | **Test case/Test script** | **Attribute and value** | **Expected result** | **Actual result** | **Result** |
| 1. | Admin successfully adds video | Video link: https://www.youtube.com/watch?v=Q33KBiDriJY | Video gets uploaded. | Video gets uploaded. | Pass |
| 2. | Student successfully asks question on discussion form | Question:  What does ABR mean? | Question displayed on discussion form. | Question displayed on discussion form. | Pass |

**Integration Testing 3:** Audio/ Video call

**Testing Objective:** To ensure the call is successfully being placed between two users and *the* ***interface*** *between* module ‘User profiling’ and module ‘Audio/Video call’ *is running correctly*.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **No.** | **Test case/Test script** | **Attribute and value** | **Expected result** | **Actual result** | **Result** |
| 1. | Start video call by Shehryar (student) logging in and stating call by clicking on start video call button. | Video call button:  pressed | Call gets started with Shehryar as Host. | Call gets started with Shehryar as Host. | Pass |
| 2. | Join video call by Fahad (counselor) logging in and joining call by clicking on join video call button via shared Call ID. | Call ID: Q33KBiDriJY | Fahad joins call by entering the respective link. | Fahad joins call by entering the respective link | Pass |

# Conclusion and Future Work

We conclude our project and highlight future work as described below.



# Conclusion

We hope that the proposed system will be beneficial for the students and fresh graduate in order to guide them for their future path and shape up their career by providing them with proper counseling which is a major issue among youth as they don’t get proper future guidance. Secondly, this project helped us a lot in gaining more knowledge about networking, Machine Learning algorithms and Chabot.

# Future Work

We would like to increase the scope of this proposed system by adding more and better counseling options and making the proposed system recommend Pakistani universities along with international universities and making the application more user friendly by improving its UI design appealing for different age groups.

# References

* [www.eduvision.edu.pk](http://www.eduvision.edu.pk)
* <https://coffeemug.ai/>
* <https://www.boisestate.edu/career/what-is-career-counseling/>
* <https://www.jobs.punjab.gov.pk/new_recruit/jobs>

.