

SOFTWARE REQUIREMENT SPECIFICATION (SRS) FOR CLOUD-BASED CAREER GUIDANCE SYSTEM

1. Introduction

1.1 Purpose:

A *Cloud-based Career Guidance System* is intended to help the user explore career opportunities after 12th (for graduation), or graduation (for post-graduation). It would make available a list of various colleges and the courses they offer. There'd be the possibility to take tests, the results of which would be suggestions for a prospective career. Another functionality to be available is a forum for queries and questions. Finally, the user would have access and be able to read career-guidance e-books.

This document is meant to delineate the features of this *Cloud-based Career Guidance System*, so as to serve as a guide to the developers on one hand and a software validation document for the prospective client on the other.

1.2 Scope:

We describe what features are in the scope of the software and what are not in the scope of the software to be developed.

In Scope:

- a. Managing the profile of a user, which includes maintaining personal details of the user, and his/her activity on the system.
- b. Authentication of the user, using an email-based auto-generated OTP.
- c. The ability of a user to take customized tests on the system and view results.
- d. The ability to search and view available e-books.
- e. The ability to search and filter for disciplines, colleges, and courses.
- f. The ability of an admin user to add questions to a question bank.
- g. The ability of an admin user to create custom tests using the question bank, and add constraints for guidance based on different results.
- h. The ability of a user to create/edit and answer forum questions.
- i. The ability of an admin user to add/edit/remove e-books.

Out of Scope:

- a. Feature for actually applying to colleges.

1.3 Terms:

- a. **Admin:** A master user having the permission of management of the system in terms of adding/editing/removing the content available. This user does not take tests himself/herself, but may contribute to creating/modifying them.

- b. **Non-admin User:** A target user who seeks career guidance. This person can take tests, but not create them.
- c. **User:** Either of admin or non-admin user.
- d. **Question:** A tuple of an interrogative/assertive statement and a set of replies.
- e. **Test:** A set of questions, with the set of accepted sets of replies.
- f. **Forum:** An interactive area of the application with a stream flow, where one can add doubts/queries and/or reply to them.

2. Overall Description:

2.1 Product Perspective:

The *Cloud-based Career Guidance System* aims to automate the job to suggesting a prospective career to a user, based on multiple parameters like interests and candidate evaluation test scores. This is similar to the job of that of a career counsellor, with the exception that this would be through an automated web-based software rather than in person.

2.2 Product Functions:

The Cloud-based Career Guidance System should support the following use cases:

Class of Use Cases	Use Cases	Description of Use Case
Use cases related to <i>User Authorization</i>	Create Account	Create a new account
	Login	Login to existing account
	Reset Password	Change password for existing account
Use cases related to <i>College / courses</i>	Add College	Add college details
	Add Course	Add Course details
	Add Discipline	Add discipline of study
	Filter College / Course / Discipline	Return a list of colleges / courses / disciplines based on some criteria
	View College	View details of a college along with the disciplines and courses it offers
	Remove Course	Remove Course entry
	Remove College	Remove College entry
	Remove Discipline	Remove Discipline entry
Use cases related to <i>Tests</i>	Add Question	Add a question to question bank
	Create Test	Create a test using a bunch of questions

	Undertake test	Record answers for a test
Use cases related to <i>email notifications</i>	Email Verification	Auto-generated link to verify email
	Receive OTP	Receive auto-generated OTP while login
	Receive Test Results	Receive questions, answers, and analysis for an undertaken test
Use cases related to <i>forum</i>	Add Question	Add a forum question
	Reply to Question	Reply to a question asked in the forum
	Remove Question	Remove a question
	Remove Reply	Remove reply to a question
Use cases related to <i>e-books</i>	Add e-book	Add e-book details
	Filter e-book	Return a list of e-books based on filter criteria
	View e-books	View details of an e-book
	Remove e-book	Remove e-book entry

2.3 Principal Actors:

The two principal actors in the *Cloud-based Career Guidance System* are “user” and “admin”.

2.4 General Constraints:

- a. Requires an active Internet connection.

2.5 Assumptions and Dependencies:

- a. The user has a browser up to date with the minimum version requirements of the application.

3. Specific Requirements:

3.1 Functional Requirements:

We describe the functional requirements by giving various use cases.

Use cases related to User Authentication:

Use case 1: Create account

Primary Actor: User

Pre-Condition: Internet connection available

Main Scenario:

1. User visits website domain in the browser
2. User selects “Sign Up”

3. User provided with a form with fields: first name, middle name, last name, email, qualification (dropdown), password
4. Validate details' format
5. Send verification link on email
6. User clicks on verification link
7. New user entry is created in the database
8. User logged in and redirected to home page

Alternate Scenario:

- 4(a) Details format mismatch
 - 4(a) 1. Prompt user to enter correct details in the field/s
 - 4(a) 2. Clear fields
- 4(b) Email exists in database
 - 4(b) 1. Inform the user that account with provided email exists
 - 4(b) 2. Clear fields

Use case 2: Login

Primary Actor: User

Pre-condition: Internet connection available

Main Scenario:

1. User visits website domain in the browser
2. User prompted to provide login credentials
3. User enters email and password
4. Account Authentication
5. User logged in and redirected to homepage

Alternate Scenario:

- 4(a) Authentication fails
 - 4(a) 1. Prompt user to enter correct details
 - 4(b) 2. Clear fields

Use case 3: Reset Password

Primary Actor: User

Pre-condition: Internet connection available

Main Scenario:

1. User visits website domain in the browser
2. User selects "forget password"
3. User sent a link to email
4. User visits link
5. User prompted to enter new password
6. Validate password

7. Update password in database
8. User logged in and redirected to homepage

Alternate Scenario:

- 6(a) Password fails minimum criteria
 - 6(a) 1. Prompt user to enter valid password
 - 6(a) 2. Clear field

Use cases related to Colleges/courses:

Use case 4: Add College

Primary Actor: Admin

Pre-condition: Logged in from admin account

Main Scenario:

1. Add college link visited
2. Admin presented with a form with field: college name, city, governing body, admission policy, and college abstract
3. Admin enters data
4. Validate college data
5. Add college to database
6. Prompt user of successful addition

Alternate Scenario:

- 4(a) College name already exists
 - 4(a) 1. Prompt user that college name already exists
 - 4(a) 2. Clear fields

Use case 5: Add Course

Primary Actor: Admin

Pre-condition: Logged in from admin account

Main Scenario:

1. Add course link visited
2. Admin presented with a form with field: college, course name, major, duration, level, required qualifications, discipline (dropdown: available for selected college)
3. Admin enters data
4. Validate course data
5. Add course to database
6. Prompt user of successful addition

Alternate Scenario:

- 4(a) Course name already exists
 - 4(a) 1. Prompt user that course name already exists
 - 4(a) 2. Clear field

Use case 6: Add Discipline

Primary Actor: Admin

Pre-condition: Logged in from admin account

Main Scenario:

1. Add discipline link visited
2. Admin presented with a form with field: discipline name
3. Admin enters data
4. Validate discipline data
5. Add discipline to database
6. Prompt user of successful addition

Alternate Scenario:

- 4(a) Discipline already exists
 - 4(a) 1. Prompt user that discipline already exists
 - 4(a) 2. Clear field

Use case 7: Filter College/Course/Discipline

Primary Actor: User

Pre-condition: User logged in

Main Scenario:

1. User visits browse page
2. User presented with a form with auto-suggest fields: college name, course, discipline
3. User enters details
4. Filter results displayed as a paginated list having college name, course, discipline matching filter criteria

Use case 8: View College

Primary Actor: User

Pre-condition: User logged in

Main Scenario:

1. User clicks on corresponding filter list entry
2. User displayed a modal having college details

Use case 9: Remove College

Primary Actor: Admin

Pre-condition: Logged in from admin account

Main Scenario:

1. User click delete button corresponding to filter list entry
2. User shown a confirmation modal
3. User confirms

4. College removed from database

Alternate Scenario:

- 3(a) User denies confirmation
 - 3(a) 1. Modal closed

Use case 10: Remove Course

Primary Actor: Admin

Pre-condition: Logged in from admin account

Main Scenario:

1. User clicks delete course button in a college view modal
2. User shown a confirmation modal
3. User confirms
4. Course removed from database

Alternate Scenario:

- 3(a) User denies confirmation
 - 3(a) 1. Modal closed

Use case 11: Remove Discipline

Primary Actor: Admin

Pre-condition: Logged in from admin account

Main Scenario:

1. User clicks delete discipline button in a college view modal
2. User shown a confirmation modal
3. User confirms
4. Discipline removed from database

Alternate Scenario:

- 3(a) User denies confirmation
 - 3(a) 1. Modal closed

Use cases related to Tests:

Use case 12: Add Question

Primary Actor: Admin

Pre-condition: Logged in from admin account

Main Scenario:

1. User visits add question link
2. User displayed a modal with fields: Question statement, answers
3. User enters details and submits
4. Validate fields
5. Question added to question bank

6. Modal closed

Alternate scenario:

4(a) Any field empty

4(a) 1. Prompt user to fill field details

Use case 13: Create Test

Primary Actor: Admin

Pre-condition: Logged in from admin account

Main Scenario:

1. User visits create test link
2. User displayed a modal with a field for test name, and a list of all questions along with a check box for each
3. User enters details and submits
4. Validate fields
5. Display a list of selected questions along with a text box for each for weightage
6. User enters weights and submits
7. Validate fields
8. Display fields to add test result criterion corresponding to multiple scores
9. User fills details and submits
10. Validate data
11. Add test in database
12. Prompt user of successful entry

Alternate Scenario:

- 4(a) Test name empty or test name exists in database or less than minimum number of questions selected
 - 4(a) 1. Prompt user for action
- 7 (a) Weights missing
 - 7(a) 1. Prompt user for action
- 10(a) Details invalid
 - 10(a) 1. Prompt user for action

Use case 14: Undertake Test

Primary Actor: User

Pre-condition: Logged in from non-admin account

Main Scenario:

1. User visits undertake test link
2. User shown a list of tests available
3. User selects test
4. User shown questions one at a time with answer fields, and a sidebar of question number links corresponding to each question

5. User selects end test
6. Test submitted and details written to database
7. Test results emailed
8. User displayed test analysis

Use cases related to email notifications

Use case 15: Email Verification

Primary Actor: System

Pre-condition: Create account requested

Main Scenario:

1. Generate a verification token using the user details and generate a link carrying it as payload
2. Send email it to email ID
3. User clicks link
4. User's account is created

Alternate Scenario:

- 3(a) Link not clicked within 24 hours of generation
- 3(a) 1. Link expires and error message is shown to user.

Use case 16: Send OTP

Primary Actor: System

Pre-condition: Log in requested

Main Scenario:

1. Generate a random 6-digit number and send it to user's email

Use case 17: Send Test Results

Primary Actor: System

Pre-condition: Test submitted

Main Scenario:

1. Send test report to user's email

Use cases related to forum:

Use case 18: Add Question

Primary Actor: Non-admin User

Pre-condition: Logged in from non-admin account

Main Scenario:

1. User visits add forum question link
2. User displayed a modal with field: Question
3. User enters questions and submits
4. Validate question

5. Question added to the database
6. Modal closed

Alternate Scenario:

- 7(a) Question length less than minimum length
- 7(a) 1. Prompt user to correct data

Use case 19: Remove Question

Primary Actor: Non-admin User

Pre-condition: Logged in from non-admin account

Main Scenario:

1. User visits remove forum question link
2. User selects question
3. User displayed a confirmation modal
4. User confirms deletion
5. Question removed from database
6. Modal closed

Alternate Scenario:

- 4(a) User denies confirmation
- 4(a) 1. Modal closed

Use case 20: Add Reply

Primary Actor: User

Pre-condition: Logged in

Main Scenario:

1. User selects add reply to a forum question
2. User displayed a field: reply after the question
3. User adds reply
4. Validate reply
5. Add reply to database
6. Field closed

Alternate Scenario:

- 4(a) Reply field is empty
- 4(a) 1. Prompt user to enter something

Use case 21: Remove Reply

Primary Actor: User

Pre-condition: Logged in

Main Scenario:

1. User visits forum question link
2. User selects delete reply

3. User displayed a confirmation modal
4. User confirms deletion
5. Reply removed from database
6. Modal closed

Alternate Scenario:

- 4(a) User denies confirmation
- 4(a) 1. Modal closed

Use cases related to e-books

Use case 22: Add e-book

Primary Actor: Admin

Pre-condition: Logged in from admin account

Main Scenario:

1. User visits add e-book link
2. User displayed a modal with fields: e-book id, name, authors, category, and e-book link (file upload)
3. User adds data
4. Validate data
5. E-book added to database
6. User displayed success message

Alternate Scenario:

- 4(a) E-book name or file missing
- 4(a) 1. Prompt user to enter data

Use case 23: Filter e-book

Primary Actor: User

Pre-condition: Logged in

Main Scenario:

1. User visits e-book browse page
2. User is displayed fields: e-book search, category (dropdown)
3. User adds search filters
4. User displayed a list of e-books based on filter criteria

Use case 24: View e-book

Primary Actor: User

Pre-condition: User clicks on e-book filter list item

Main Scenario:

1. e-book launched as a modal

Use case 25: Remove e-book

Primary Actor: Admin

Pre-condition: User visiting e-book

Main scenario:

1. User clicks on remove e-book
2. User displayed a confirmation modal
3. User confirms
4. E-book removed from database
5. Success message displayed to user

Alternate scenario:

- 3(a) User denies confirmation
 - 3(a) 1. Modal closed

3.2 Software Quality Attributes:

- a. The database management system should enforce “ACID” (*Atomicity, Consistency, Isolation, Durability*) properties so that there is no inconsistency between reads (view) and writes (addition/updating).
- b. The database schema should enforce integrity constraints, so that there is no disallowed duplication, reference error, etc.