

Table of Contents

User Registration and Authentication	1
1. SRS	1
2. UMLS	4
3. Testing.....	6
Resume-Based Job Recommendation System	10
1. SRS	10
2. UMLS	11
3. Testing.....	13
Job Management System	15
1. SRS	15
2. UMLS	17
3. Testing.....	18
Review Management System.....	22
1. SRS	22
2. UMLS	23
3. Testing.....	24
Chat System	27
1. SRS	27
2. UMLS	28
3. Testing.....	28
Schema Design Diagram.....	31
Wireframes and Designs	32

User Registration and Authentication

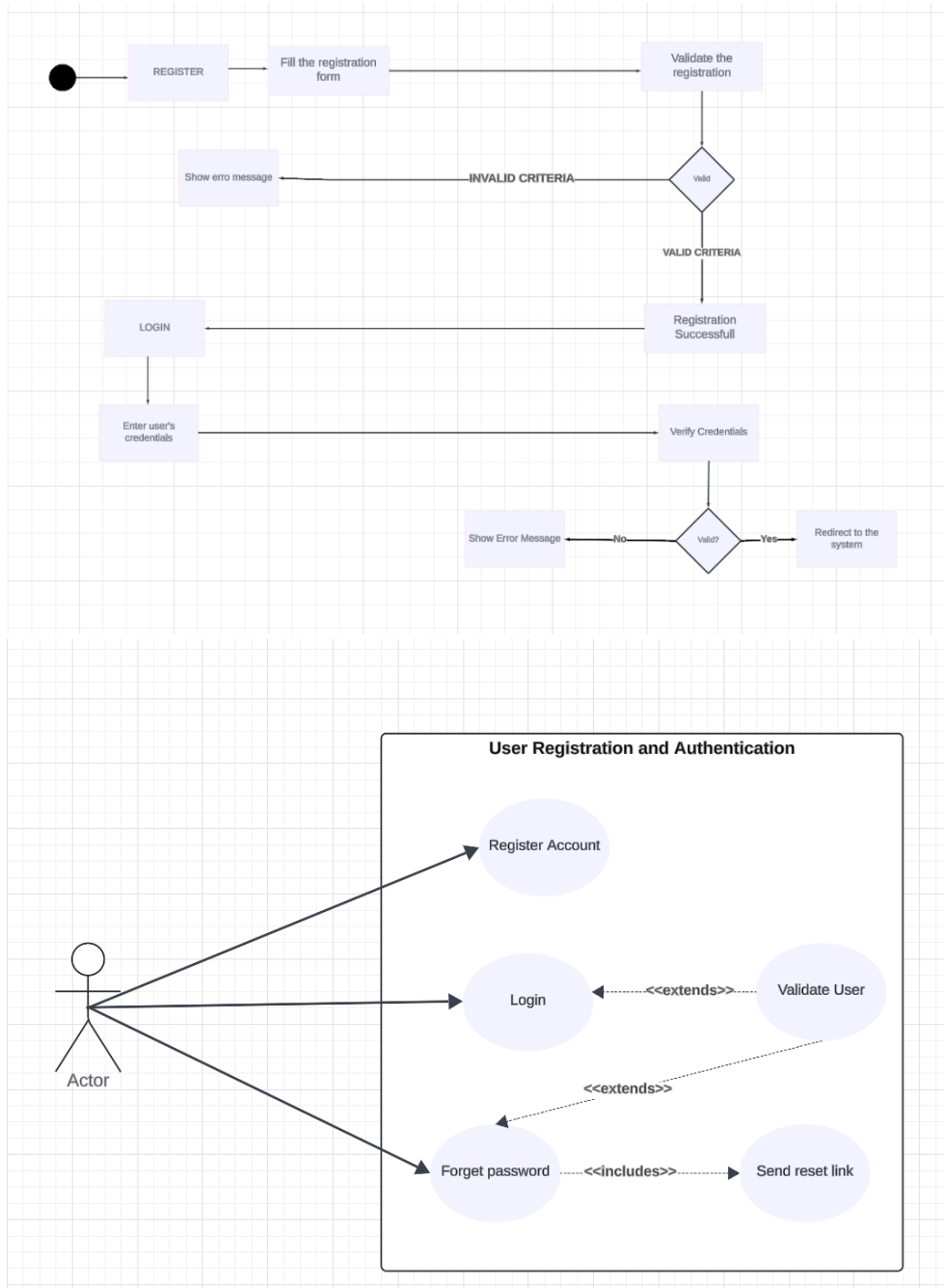
1. SRS

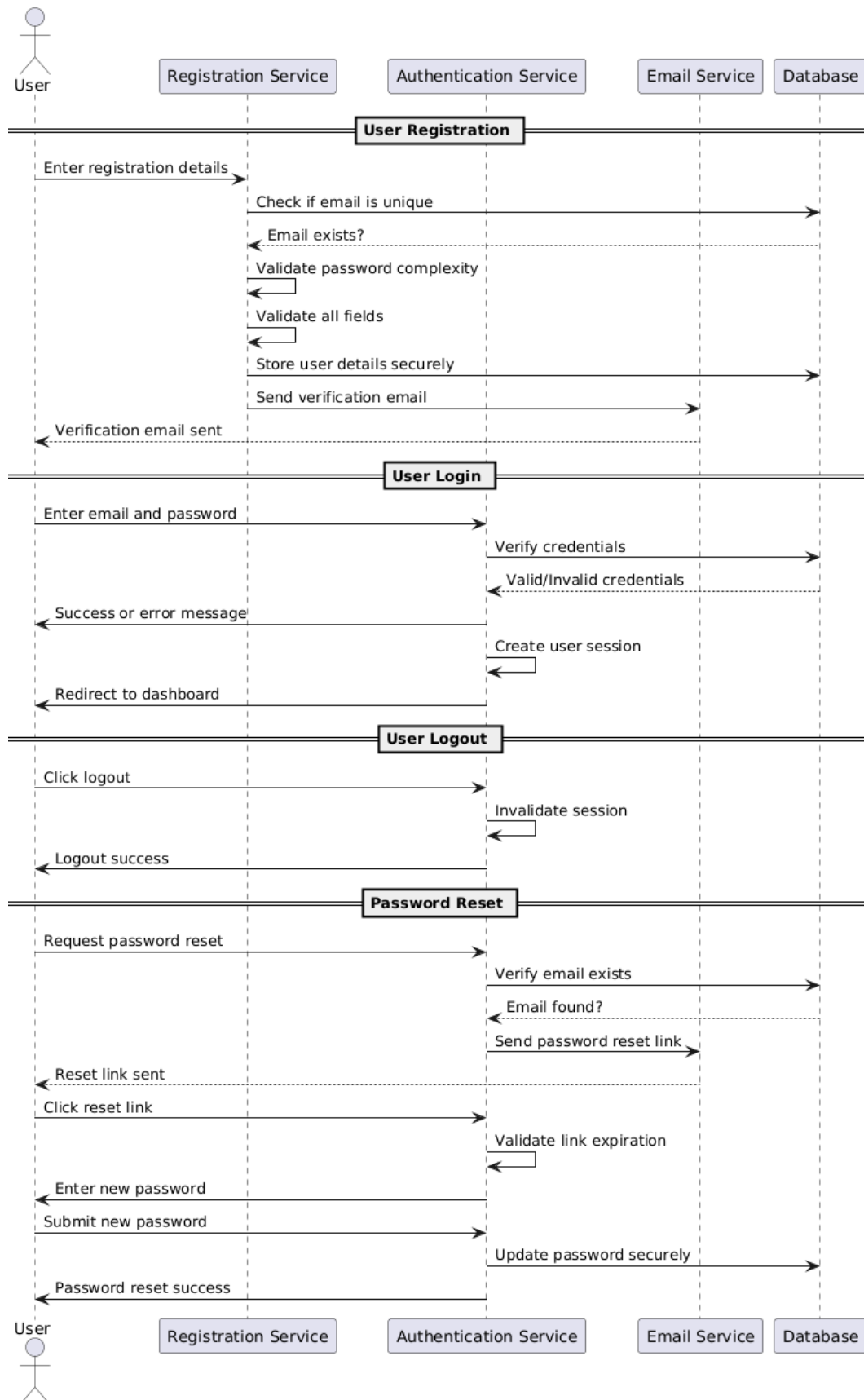
Req. Code	Req. Description	Use Case
URA-F-1.0	The system shall allow new users to register an account.	User Registration
URA-F-1.1	The system shall require a unique email for each user account.	User Registration
URA-F-1.2	The system shall require a password that meets defined complexity criteria (e.g., minimum length, uppercase/lowercase, numbers, special characters).	User Registration
URA-F-1.3	The system shall require password confirmation during registration.	User Registration
URA-F-1.4	The system shall validate all registration fields to ensure required information is provided and is in the correct format.	User Registration
URA-F-1.5	The system shall store user registration information securely in a database.	User Registration
URA-F-1.6	The system shall send an email to the registered email address.	User Registration
URA-UR-1.7	The system shall provide appropriate error messages to the user during registration if validation fails.	User Registration
URA-F-1.8	The system shall allow registered users to log in with their email and password.	User Login

URA-F-1.9	The system shall verify user credentials against stored data.	User Login
URA-F-1.10	The system shall allow users to login using password-based authentication.	User Login
URA-NF-1.11	The system shall implement secure password handling by not storing passwords in plaintext.	User Login
URA-F-1.12	Upon successful login, the system shall create a session for the logged-in user.	User Login
URA-F-1.13	Upon successful login, the system shall redirect the user to the appropriate default page.	User Login
URA-F-1.14	The system shall allow logged-in users to log out securely.	User Logout
URA-F-1.15	The system shall invalidate user session after logging out.	User Logout
URA-UR-1.16	The system shall display an error message to the user if login fails (invalid username/password).	User Login
URA-F-1.17	The system shall provide a "Forgot Password" feature.	Password Reset
URA-F-1.18	The system shall send a password reset link (or code) to the user's registered email address.	Password Reset
URA-F-1.19	The system shall allow users to reset their password using the received link (or code).	Password Reset
URA-F-1.20	The system shall ensure the password reset link is time limited.	Password Reset

URA-NF-1.21	The system shall encrypt sensitive data, like passwords, at rest and in transit.	Security
URA-NF-1.22	The system shall protect against common security vulnerabilities (e.g., SQL injection, cross-site scripting).	Security
URA-NF-1.23	The system shall not store passwords or email reset links in plain text	Security
URA-NF-1.24	The system shall protect access to user accounts through strong authentication and authorization	Security
URA-NF-1.25	The system shall respond to user requests (e.g., login, registration)	General
URA-UR-1.26	The system shall provide clear visual feedback to the user during any processing that takes longer than 1 second .	General
URA-UR-1.27	The user interface shall adapt to various screen sizes (desktop, tablet, mobile) without loss of functionality or usability.	General

2. UMLS





3. Testing

Req. Code	Req. Description	Test ID	Test Case Description	Expected Result	Status
URA-F-1.0	The system shall allow new users to register an account.	TC-URA-F-1.0_001	Verify successful user registration with valid input data.	New user account is successfully created. Confirmation email is sent.	Pass
URA-F-1.1	The system shall require a unique email for each user account.	TC-URA-F-1.1_002	Attempt to register a new user with an email already associated with another user.	System should display an error message stating email is already registered. Registration should fail.	Pass
URA-F-1.2	The system shall require a password that meets defined complexity criteria.	TC-URA-F-1.2_003	Attempt to register with a password that is too short or does not contain required characters (uppercase, lowercase, numbers, special characters).	System should display an error message stating password does not meet complexity requirements. Registration should fail.	Pass
URA-F-1.2	The system shall require a password that meets defined complexity criteria.	TC-URA-F-1.2_004	Attempt to register with a valid password meeting all complexity requirements.	System should allow user creation with successful password.	Pass
URA-F-1.3	The system shall require password confirmation during registration.	TC-URA-F-1.3_005	Attempt to register with password and confirmation fields not matching.	System should display an error message stating passwords do not match. Registration should fail.	Pass
URA-F-1.3	The system shall require password confirmation during registration.	TC-URA-F-1.3_006	Attempt to register with password and confirmation fields matching.	System should allow password confirmation.	Pass
URA-F-1.4	The system shall validate all registration fields to ensure required information is provided and is in the correct format.	TC-URA-F-1.4_007	Attempt to register with missing required fields (email, password etc.) or incorrect format data.	System should display appropriate error message stating the invalid field. Registration should fail.	Pass
URA-F-1.4	The system shall validate all registration fields to ensure required	TC-URA-F-1.4_008	Attempt to register with all required fields with correct format and proper input data.	User registration should be successful.	Pass

	information is provided and is in the correct format.				
URA-F-1.5	The system shall store user registration information securely in a database.	TC-URA-F-1.5_009	Verify data stored in the database after successful registration, including hashing of password.	User data is correctly stored in the database with hashed password.	Pass
URA-F-1.6	The system shall send an email to the registered email address.	TC-URA-F-1.6_010	Register a new user and verify the confirmation email is received.	Email should be delivered to the specified email address upon successful registration.	Pass
URA-UR-1.7	The system shall provide appropriate error messages to the user during registration if validation fails.	TC-URA-UR-1.7_011	Verify all error messages during registration and provide meaningful feedback on the cause of failure.	Relevant and understandable error messages displayed for each registration validation failure.	Pass
URA-F-1.8	The system shall allow registered users to log in with their email and password.	TC-URA-F-1.8_001	Attempt to log in with a registered user's valid email and password.	User should be successfully logged in and directed to the default page.	Pass
URA-F-1.9	The system shall verify user credentials against stored data.	TC-URA-F-1.9_002	Attempt to log in with an invalid email or password.	System should display an error message stating invalid credentials and login should fail.	Pass
URA-F-1.10	The system shall allow users to login using password-based authentication.	TC-URA-F-1.10_003	Attempt to login using valid credentials and password authentication method.	User should be successfully logged in.	Pass
URA-NF-1.11	The system shall implement secure password handling by not storing passwords in plaintext.	TC-URA-NF-1.11_004	Verify password storage method in the database after successful registration.	Passwords should be stored using a strong hashing method.	Pass
URA-F-1.12	Upon successful login, the system shall create a session for the logged-in user.	TC-URA-F-1.12_005	Log in with a valid user and verify the presence of an active session.	Session for user is created and maintained.	Pass
URA-F-1.13	Upon successful login, the system shall redirect the user to the appropriate default page.	TC-URA-F-1.13_006	Log in with a valid user and verify user is redirected to the default homepage after successful login.	User is redirected to the correct default page after successful login.	Pass

URA-F-1.14	The system shall allow logged-in users to log out securely.	TC-URA-F-1.14_001	Log in and then log out and ensure log out process is successful.	User is logged out successfully and is no longer an active session.	Pass
URA-F-1.15	The system shall invalidate user session after logging out.	TC-URA-F-1.15_002	Log in and then log out, then try accessing authorized pages.	User should not be able to access authorized pages after session invalidation.	Pass
URA-UR-1.16	The system shall display an error message to the user if login fails (invalid username/password).	TC-URA-UR-1.16_007	Attempt login with invalid credentials and verify an error message is displayed.	Error message should be displayed indicating login failed and should indicate if it is due to incorrect credentials.	Pass
URA-F-1.17	The system shall provide a "Forgot Password" feature.	TC-URA-F-1.17_001	Request a password reset and verify the reset password link email is received.	User should receive an email with the password reset link.	Pass
URA-F-1.18	The system shall send a password reset link (or code) to the user's registered email address.	TC-URA-F-1.18_002	Request a password reset and verify the reset link received to users mail.	Password reset email with valid reset link is delivered to the specified email address.	Pass
URA-F-1.19	The system shall allow users to reset their password using the received link (or code).	TC-URA-F-1.19_003	Reset password using the link received to email and ensure successful password reset and that they are logged in with the updated password.	User can successfully reset their password and log in with the new password.	Pass
URA-F-1.20	The system shall ensure the password reset link is time limited.	TC-URA-F-1.20_004	Request a password reset and wait beyond the time limit of the reset link and then attempt to reset the password using the same link.	Password reset using an expired link should fail and display error message to the user.	Pass
URA-NF-1.21	The system shall encrypt sensitive data, like passwords, at rest and in transit.	TC-URA-NF-1.21_001	Verify all sensitive data (passwords, tokens) are encrypted both in the database and during network communication.	Sensitive data is encrypted in both database at rest and during transit.	Pass
URA-NF-1.22	The system shall protect against common security vulnerabilities (e.g., SQL injection, cross-site scripting).	TC-URA-NF-1.22_002	Conduct vulnerability scans and penetration testing to ensure system is resistant to common security vulnerabilities.	System should be resistant to common security vulnerabilities and should not be easily attacked.	Pass

URA-NF-1.23	The system shall not store passwords or email reset links in plain text.	TC-URA-NF-1.23_003	Verify password and email reset link storage in the database, ensure plain text is not being used.	Passwords and reset links should be stored in a secure, encrypted format, never in plain text.	Pass
URA-NF-1.24	The system shall protect access to user accounts through strong authentication and authorization.	TC-URA-NF-1.24_004	Test access to user accounts with valid user credentials, ensure unauthorized users cannot access others accounts or functionalities.	Only authorized users should have access to their own accounts and functionalities.	Pass
URA-NF-1.25	The system shall respond to user requests (e.g., login, registration).	TC-URA-NF-1.25_001	Perform multiple login, registration, password reset requests to ensure system does respond to user requests.	System responds to all user requests with appropriate time.	Pass
URA-UR-1.26	The system shall provide clear visual feedback to the user during any processing that takes longer than 1 second.	TC-URA-UR-1.26_002	Perform actions like registration and password reset that take more than 1 second and verify the loading indicator is provided.	Loading animation/indicator should be present if any user request is taking longer than 1 second.	Pass
URA-UR-1.27	The user interface shall adapt to various screen sizes	TC-URA-UR-1.27_003	Test on different screen sizes to ensure the responsive	Application UI should adapt to different screen size without any loss of functionality and usability.	Pass

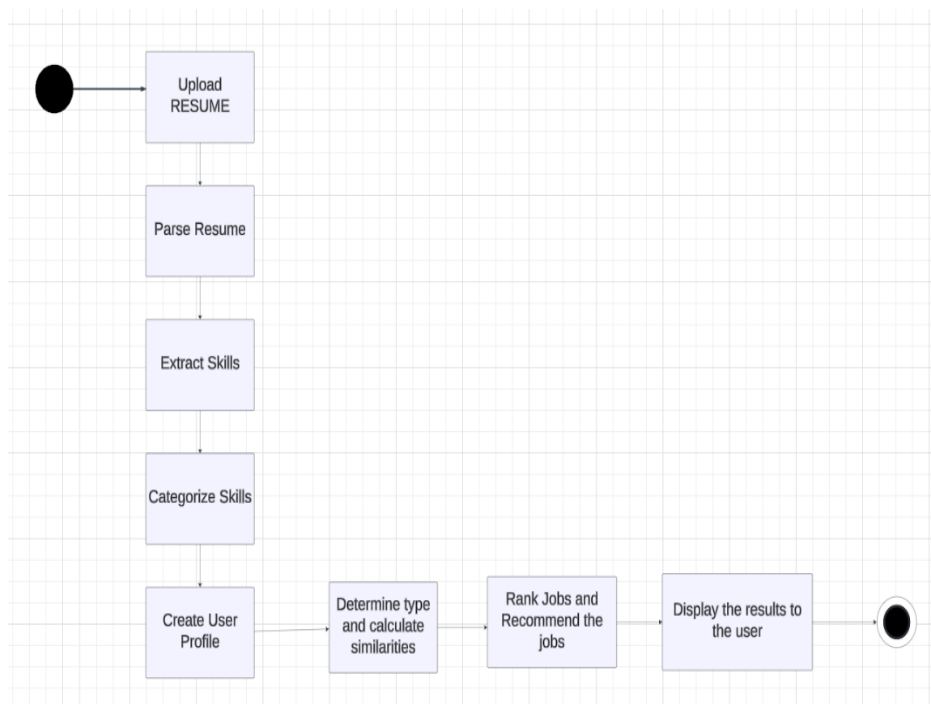
Resume-Based Job Recommendation System

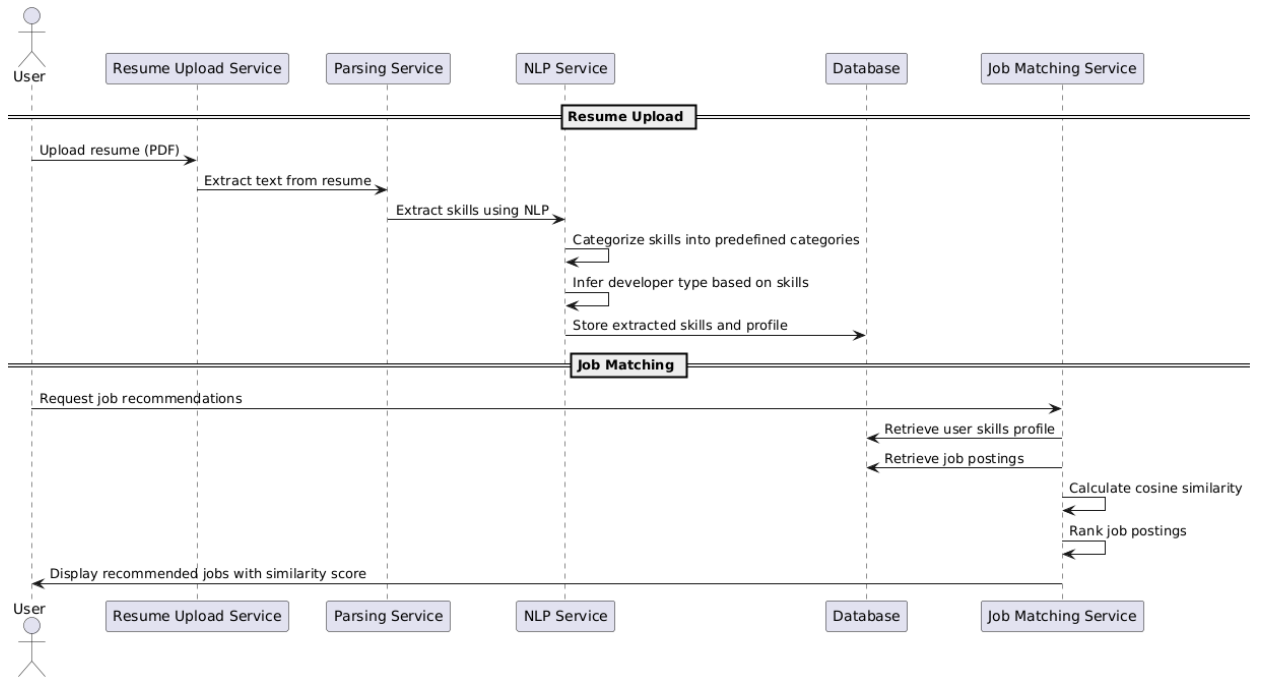
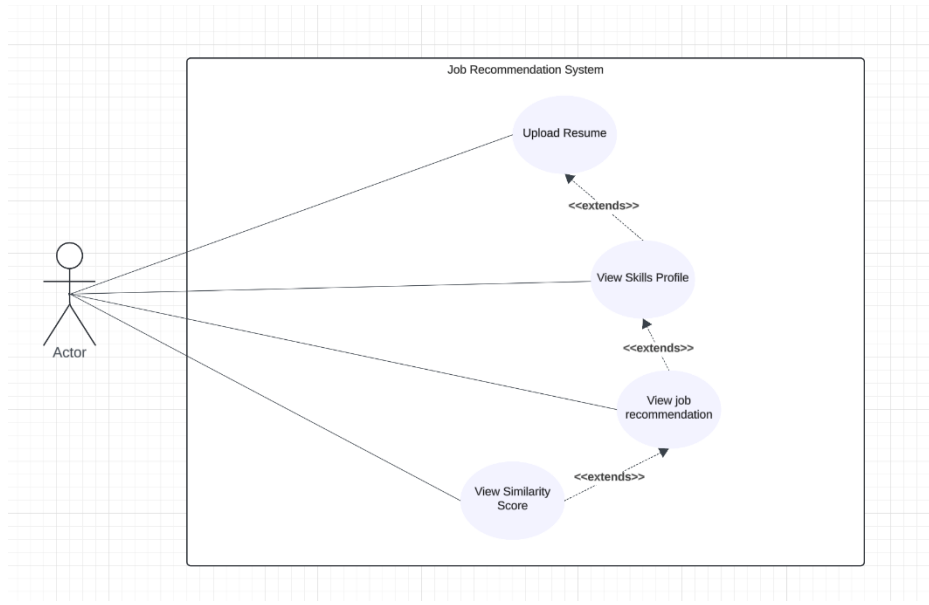
1. SRS

Req. Code	Req. Description	Use Case
JRS-F-1.0	The system shall allow users to upload their resume in a common format (e.g., PDF).	Resume Upload
JRS-F-1.1	The system shall parse the uploaded resume to extract text content.	Resume Parsing
JRS-F-1.2	The system shall identify and extract relevant skills mentioned in the resume using Natural Language Processing (NLP) techniques (e.g., Named Entity Recognition, keyword extraction).	Skills Extraction
JRS-F-1.3	The system shall categorize skills into predefined categories (e.g., Programming Languages, Databases, Frameworks, Cloud Technologies, etc.).	Skills Categorization
JRS-F-1.4	The system shall analyze the extracted skills and job titles/descriptions in the resume to determine the developer type (e.g., Frontend, Backend, Full-Stack, Mobile, DevOps, etc.).	Developer Type Inference
JRS-F-1.5	The system shall create a skills profile for each user based on the extracted and categorized skills.	User Profile Generation
JRS-F-1.6	The system shall calculate the cosine similarity between the user's skills profile and job profile for all job postings in the system.	Similarity Calculation
JRS-F-1.7	The system shall rank job postings based on their cosine similarity score to a user's skill profile.	Job Ranking

JRS-F-1.8	The system shall recommend the top N job postings with highest cosine similarity score to the user. (where N is configurable)	Job Recommendation
JRS-F-1.9	The system shall display recommended job postings to the user in an understandable format, including job title, company, short description, and a link to the full posting.	Job Display
JRS-F-1.10	The system shall allow user to see the similarity score between his profile and the job recommended.	Job Display

2. UMLS





3. Testing

Req. Code	Req. Description	Test ID	Test Case Description	Expected Result	Status
JRS-F-1.0	The system shall allow users to upload their resume in a common format (e.g., PDF).	TC-JRS-F-1.0_001	Attempt to upload a resume in PDF format.	Resume is uploaded successfully.	Pass
JRS-F-1.0	The system shall allow users to upload their resume in a common format (e.g., PDF).	TC-JRS-F-1.0_002	Attempt to upload a resume in a non-supported format (e.g., DOCX).	System displays an error message indicating the file format is not supported.	Pass
JRS-F-1.1	The system shall parse the uploaded resume to extract text content.	TC-JRS-F-1.1_003	Upload a resume and verify that the text content is extracted correctly.	Text content of the resume is extracted without any errors.	Pass
JRS-F-1.1	The system shall parse the uploaded resume to extract text content.	TC-JRS-F-1.1_004	Upload a resume with image-based text and verify the parsing failure.	Text content is not parsed or parsed with errors.	Pass
JRS-F-1.2	The system shall identify and extract relevant skills mentioned in the resume using Natural Language Processing (NLP) techniques (e.g., Named Entity Recognition, keyword extraction).	TC-JRS-F-1.2_005	Upload a resume with clearly listed technical skills.	All listed technical skills are identified and extracted correctly.	Pass
JRS-F-1.2	The system shall identify and extract relevant skills mentioned in the resume using Natural Language Processing (NLP) techniques (e.g., Named Entity Recognition, keyword extraction).	TC-JRS-F-1.2_006	Upload a resume with skills mentioned in various contexts (e.g., project descriptions, job roles).	Skills mentioned in different contexts are identified and extracted accurately.	Fail
JRS-F-1.3	The system shall categorize skills into predefined categories (e.g., Programming Languages, Databases, Frameworks, Cloud Technologies, etc.).	TC-JRS-F-1.3_007	Upload a resume with varied skills and validate categorization.	Extracted skills are categorized into predefined categories correctly.	Pass
JRS-F-1.3	The system shall categorize skills into predefined categories (e.g., Programming Languages, Databases, Frameworks, Cloud Technologies, etc.).	TC-JRS-F-1.3_008	Upload a resume with skills that don't fit predefined categories.	Skills that don't fit predefined categories are either categorized into a default category or are flagged.	Fail
JRS-F-1.4	The system shall analyze the extracted skills and job titles/descriptions in the resume to determine the developer type (e.g., Frontend, Backend, Full-Stack, Mobile, DevOps, etc.).	TC-JRS-F-1.4_009	Upload a resume with frontend-related skills and job titles.	Developer type should be identified as Frontend Developer.	Pass
JRS-F-1.4	The system shall analyze the extracted skills and job titles/descriptions in the resume to determine the developer type	TC-JRS-F-1.4_010	Upload a resume with skills and titles related to both	Developer type should be identified	Pass

	(e.g., Frontend, Backend, Full-Stack, Mobile, DevOps, etc.).		frontend and backend.	as Full-Stack Developer.	
JRS-F-1.5	The system shall create a skills profile for each user based on the extracted and categorized skills.	TC-JRS-F-1.5_011	Upload a resume and verify the user profile generated.	A user skills profile is generated based on the extracted and categorized skills.	Pass
JRS-F-1.5	The system shall create a skills profile for each user based on the extracted and categorized skills.	TC-JRS-F-1.5_012	Upload multiple resumes and verify the latest user profile.	Latest profile is updated based on latest uploaded resume.	Pass
JRS-F-1.6	The system shall calculate the cosine similarity between the user's skills profile and job profile for all job postings in the system.	TC-JRS-F-1.6_013	Upload a resume, then compare the profile with various job postings.	Cosine similarity scores are calculated correctly for all job postings.	Pass
JRS-F-1.6	The system shall calculate the cosine similarity between the user's skills profile and job profile for all job postings in the system.	TC-JRS-F-1.6_014	Upload a resume and verify cosine similarity with no skills match.	Cosine similarity score of 0 should be given to unmatched job postings.	Pass
JRS-F-1.7	The system shall rank job postings based on their cosine similarity score to a user's skill profile.	TC-JRS-F-1.7_015	Verify job posting ranking based on cosine similarity scores.	Job postings are ranked based on the cosine similarity in descending order.	Pass
JRS-F-1.8	The system shall recommend the top N job postings with highest cosine similarity score to the user. (where N is configurable)	TC-JRS-F-1.8_016	Upload a resume and verify recommended job postings (N=3).	Top 3 jobs with highest cosine similarity scores are recommended.	Pass
JRS-F-1.8	The system shall recommend the top N job postings with highest cosine similarity score to the user. (where N is configurable)	TC-JRS-F-1.8_017	Upload a resume and verify recommended job postings (N=5).	Top 5 jobs with highest cosine similarity scores are recommended.	Pass
JRS-F-1.9	The system shall display recommended job postings to the user in an understandable format, including job title, company, short description, and a link to the full posting.	TC-JRS-F-1.9_018	Upload a resume and verify job posting display format.	Job postings are displayed with job title, company, short description, and link.	Pass
JRS-F-1.9	The system shall display recommended job postings to the user in an understandable format, including job title, company, short description, and a link to the full posting.	TC-JRS-F-1.9_019	Upload a resume and verify the UI in mobile layout.	Job postings are displayed well on mobile layout	Pass
JRS-F-1.10	The system shall allow user to see the similarity score between his profile and the job recommended.	TC-JRS-F-1.10_020	Upload resume, verify the similarity score displayed in UI with recommendation.	Cosine similarity score is displayed alongside the recommended jobs.	Pass

Job Management System

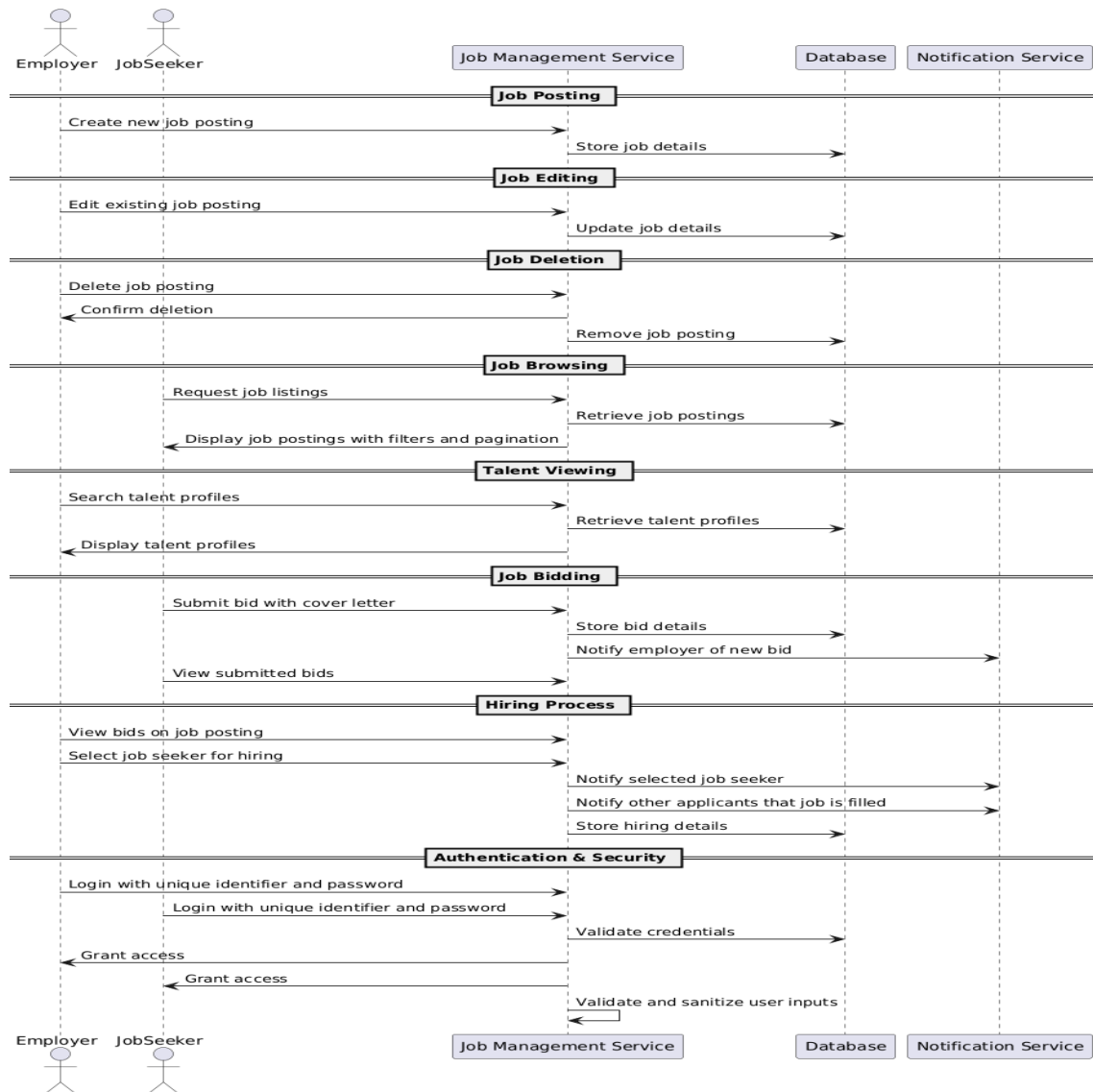
1. SRS

Req. Code	Req. Description	Use Case
JMS-F-1.0	The system shall allow registered clients to post new job openings.	Job Posting
JMS-F-1.1	Job postings shall include a job title, description, required skills, salary range.	Job Posting
JMS-F-1.2	The system shall allow employers to edit existing job postings.	Job Editing
JMS-F-1.3	Clients shall be able to modify all fields of a job posting.	Job Editing
JMS-F-1.4	The system shall allow clients to delete job postings.	Job Deletion
JMS-F-1.5	The system shall provide confirmation before permanently deleting a job posting.	Job Deletion
JMS-F-1.6	The system shall store job postings in a database.	Job Management
JMS-UR-1.7	The system shall display job postings in a user-friendly format to job seekers.	Job Browsing
JMS-F-1.8	Job seekers should be able to filter job listings based on various criteria.	Job Browsing
JMS-F-1.9	The system shall support pagination for long lists of job postings.	Job Browsing

JMS-F-1.10	The system shall allow employers to browse talent profiles.	Talent Viewing
JMS-F-1.11	The system shall allow employers to search talent profiles based on criteria.	Talent Viewing
JMS-F-1.12	Employers shall be able to see talent profiles which includes all information	Talent Viewing
JMS-F-1.13	The system shall allow job seekers to bid on job postings.	Job Bidding
JMS-F-1.14	Job seekers shall be able to submit a bid with a personalized message (cover letter).	Job Bidding
JMS-F-1.15	The system shall notify employers of new bids on their job postings.	Job Bidding
JMS-F-1.16	Job seekers shall be able to view all the jobs on which they bid.	Job Bidding
JMS-F-1.17	The system shall allow employers to view bids on their job postings.	Hiring Process
JMS-F-1.18	Employers shall be able to view job seeker profiles associated with bids.	Hiring Process
JMS-F-1.18	The system shall allow employers to select a job seeker for hiring.	Hiring Process
JMS-F-1.19	The system shall notify job seekers if they have been selected for a job.	Hiring Process
JMS-F-1.20	The system shall notify other job seekers of the job being filled.	Hiring Process

JMS-F-1.21	The system shall store information about the hiring process.	Hiring Process
JMS-F-1.22	The system shall implement authentication for employers and job seekers using a unique identifier and password.	Authentication
JMS-NF-1.23	The system shall implement data validation and sanitization for all user inputs.	Security

2. UMLS



3. Testing

Req. Code	Req. Description	Test ID	Test Case Description	Expected Result	Status
JMS-F-1.0	The system shall allow registered clients to post new job openings.	TC-JMS-F-1.0_001	Registered client attempts to post a new job.	Job opening is successfully posted.	Pass
JMS-F-1.0	The system shall allow registered clients to post new job openings.	TC-JMS-F-1.0_002	Unregistered user attempts to post a new job.	System should display an error indicating user is not authorized.	Pass
JMS-F-1.1	Job postings shall include a job title, description, required skills, salary range.	TC-JMS-F-1.1_003	Post a job with job title, description, required skills and salary range.	Job posting is created successfully and include all specified details.	Pass
JMS-F-1.1	Job postings shall include a job title, description, required skills, salary range.	TC-JMS-F-1.1_004	Post a job without some required fields.	System should display error indicating required fields are missing.	Pass
JMS-F-1.2	The system shall allow employers to edit existing job postings.	TC-JMS-F-1.2_005	Registered client edits an existing job posting.	Job posting is updated successfully.	Pass
JMS-F-1.2	The system shall allow employers to edit existing job postings.	TC-JMS-F-1.2_006	Registered client edits a job posting not owned by them.	System should display error indicating user is not authorized.	Pass
JMS-F-1.3	Clients shall be able to modify all fields of a job posting.	TC-JMS-F-1.3_007	Client modifies all fields of a job posting (title, description, skills, salary).	All fields are updated successfully.	Pass
JMS-F-1.4	The system shall allow clients to delete job postings.	TC-JMS-F-1.4_008	Client attempts to delete their own job posting.	Job posting is deleted successfully.	Pass
JMS-F-1.4	The system shall allow clients to delete job postings.	TC-JMS-F-1.4_009	Client attempts to delete another client's job posting.	System displays an error indicating user is not authorized.	Pass
JMS-F-1.5	The system shall provide confirmation before permanently deleting a job posting.	TC-JMS-F-1.5_010	Client tries to delete job posting and checks the confirmation message.	System displays a confirmation before the job is deleted permanently.	Pass

JMS-F-1.6	The system shall store job postings in a database.	TC-JMS-F-1.6_011	Post a job, verify it is stored in database.	Job posting is stored in the database successfully.	Pass
JMS-UR-1.7	The system shall display job postings in a user-friendly format to job seekers.	TC-JMS-UR-1.7_012	Browse job postings as a job seeker.	Job postings are displayed with relevant details in a clear format.	Pass
JMS-F-1.8	Job seekers should be able to filter job listings based on various criteria.	TC-JMS-F-1.8_013	Job seeker filters job postings by job title.	Job postings are filtered based on provided title successfully.	Pass
JMS-F-1.8	Job seekers should be able to filter job listings based on various criteria.	TC-JMS-F-1.8_014	Job seeker filters job postings by salary range.	Job postings are filtered based on provided salary range successfully.	Pass
JMS-F-1.8	Job seekers should be able to filter job listings based on various criteria.	TC-JMS-F-1.8_015	Job seeker filters job postings by skills.	Job postings are filtered based on provided skills successfully.	Pass
JMS-F-1.9	The system shall support pagination for long lists of job postings.	TC-JMS-F-1.9_016	Browse a long list of job postings and check pagination.	Job postings are displayed with pagination controls.	Pass
JMS-F-1.10	The system shall allow employers to browse talent profiles.	TC-JMS-F-1.10_017	Employer attempts to browse talent profiles.	Employer is able to browse the list of talent profiles.	Pass
JMS-F-1.10	The system shall allow employers to browse talent profiles.	TC-JMS-F-1.10_018	Job Seeker attempts to browse talent profiles.	System should display error indicating user is not authorized.	Pass
JMS-F-1.11	The system shall allow employers to search talent profiles based on criteria.	TC-JMS-F-1.11_019	Employer searches talent profiles based on skills.	Talent profiles are filtered based on the skill input.	Pass
JMS-F-1.11	The system shall allow employers to search talent profiles based on criteria.	TC-JMS-F-1.11_020	Employer searches talent profiles based on name.	Talent profiles are filtered based on the name input.	Pass
JMS-F-1.12	Employers shall be able to see talent profiles which includes all information	TC-JMS-F-1.12_021	Employer views a talent profile and all information is present.	Talent profile is displayed with complete set of information.	Pass

JMS-F-1.13	The system shall allow job seekers to bid on job postings.	TC-JMS-F-1.13_022	Job seeker bids on a job posting.	Job seeker is able to bid on job postings successfully.	Pass
JMS-F-1.13	The system shall allow job seekers to bid on job postings.	TC-JMS-F-1.13_023	Job seeker tries to bid on a job posting twice.	System should display error indicating job seeker already applied for the position.	Pass
JMS-F-1.14	Job seekers shall be able to submit a bid with a personalized message (cover letter).	TC-JMS-F-1.14_024	Job seeker bids with a cover letter message.	Cover letter is successfully saved with the bid.	Pass
JMS-F-1.15	The system shall notify employers of new bids on their job postings.	TC-JMS-F-1.15_025	Job seeker bids on job, employer gets the notification.	Employer gets notification on a new bid successfully.	Pass
JMS-F-1.16	Job seekers shall be able to view all the jobs on which they bid.	TC-JMS-F-1.16_026	Job seeker browses jobs which they have bid on.	Job seeker can view all jobs where they have bids.	Pass
JMS-F-1.17	The system shall allow employers to view bids on their job postings.	TC-JMS-F-1.17_027	Employer views bids on their job postings.	Employer is able to see bids on their job posting.	Pass
JMS-F-1.18	Employers shall be able to view job seeker profiles associated with bids.	TC-JMS-F-1.18_028	Employer views a job seeker profile associated with a bid.	Employer is able to see job seeker profile associated with the bid.	Pass
JMS-F-1.18	The system shall allow employers to select a job seeker for hiring.	TC-JMS-F-1.18_029	Employer selects a job seeker for hiring for a specific job.	Employer is able to hire a job seeker.	Pass
JMS-F-1.19	The system shall notify job seekers if they have been selected for a job.	TC-JMS-F-1.19_030	A selected job seeker gets the notification.	Selected job seeker receives the notification.	Pass
JMS-F-1.20	The system shall notify other job seekers of the job being filled.	TC-JMS-F-1.20_031	Other job seekers receives the notification that job is filled.	Other job seeker receives notification job position is filled.	Pass
JMS-F-1.21	The system shall store information about the hiring process.	TC-JMS-F-1.21_032	Hiring process is carried out and information stored.	Hiring process is successfully tracked and stored in database.	Pass
JMS-F-1.22	The system shall implement authentication for	TC-JMS-F-1.22_033	Login with valid user credentials	Users are able to login with valid username and	Pass

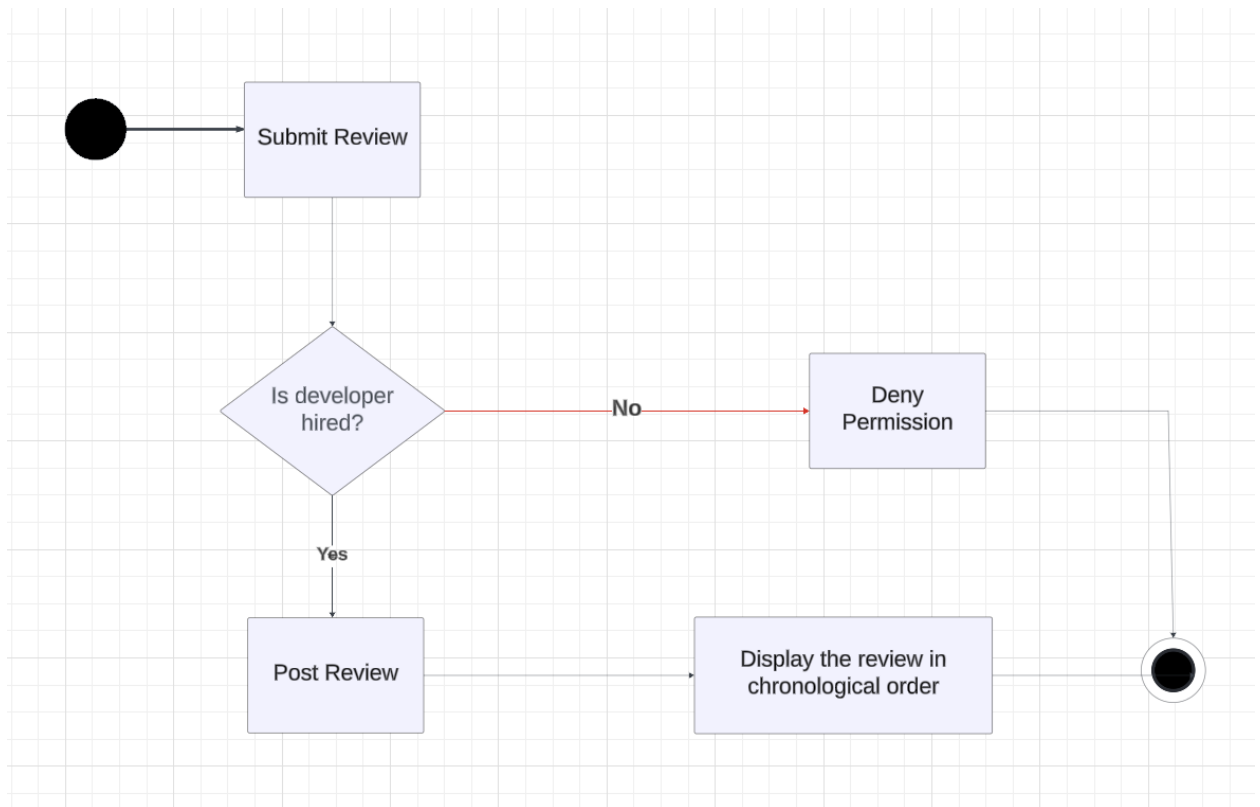
	employers and job seekers using a unique identifier and password.			password successfully.	
JMS-F-1.22	The system shall implement authentication for employers and job seekers using a unique identifier and password.	TC-JMS-F-1.22_034	Login with invalid user credentials	User is not able to login and gets error message.	Pass

Review Management System

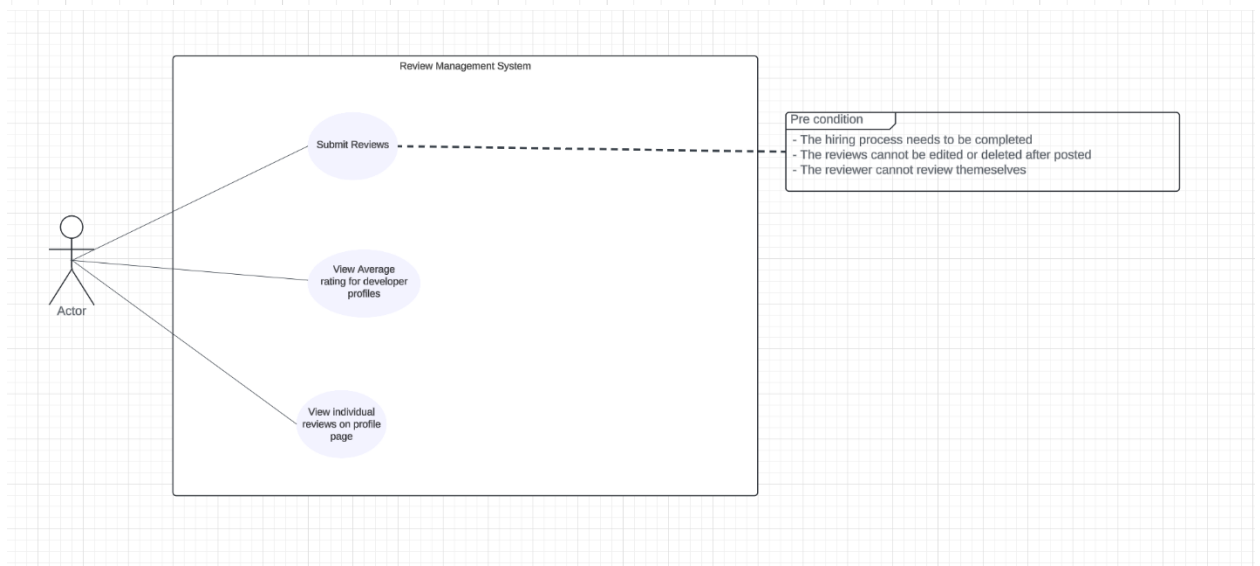
1. SRS

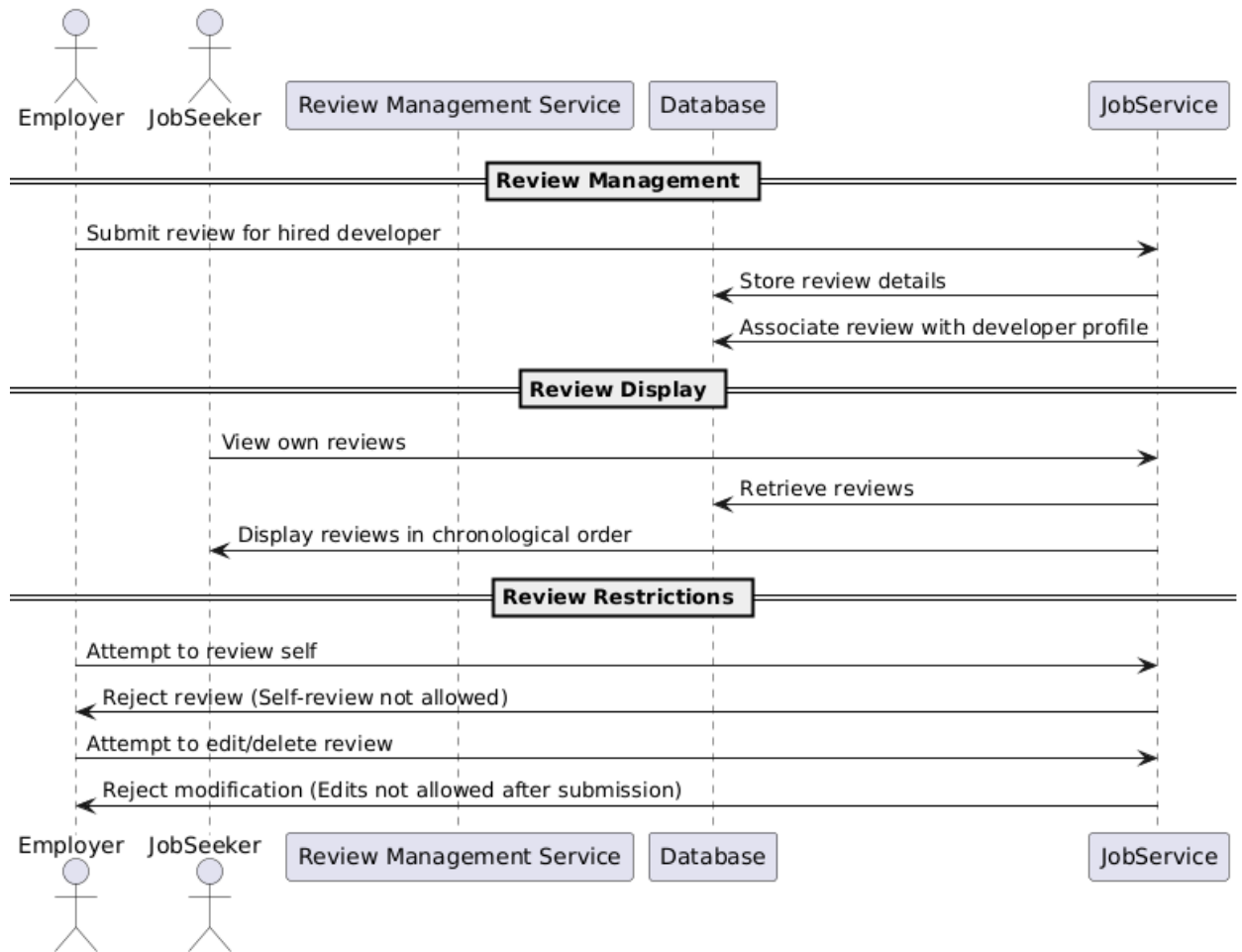
Req. Code	Req. Description	Use Case
RMS-F-1.0	The system shall allow employers to submit reviews for developers they have hired.	Review Creation
RMS-F-1.1	Reviews shall include a rating (e.g., on a scale of 1-5 stars) and an optional text-based comment.	Review Creation
RMS-F-1.2	Reviews shall be associated with the specific developer profile they were given for.	Review Association
RMS-F-1.3	The system shall not allow users to review themselves.	Review Restriction
RMS-F-1.4	Employers shall only be able to review a developer after completing a hiring process with them.	Review Restriction
RMS-F-1.5	The system shall display the average rating for each developer profile.	Review Display
RMS-F-1.6	The system shall display all individual reviews on the developer profile page.	Review Display
RMS-F-1.7	The system shall display the reviewer's name or alias with each review.	Review Display
RMS-F-1.8	Reviews shall be displayed in chronological order (most recent first).	Review Display
RMS-F-1.9	The system shall not allow employers to edit or delete reviews they have submitted after they have been posted.	Review Restriction

2. UMLS



3.





3. Testing

Test ID	Test Case Description	Expected Result	Status
TC-RMS-F-1.0_001	Employer submits a review for a hired developer.	Review is successfully submitted.	Pending
TC-RMS-F-1.0_002	Job Seeker tries to submit a review.	System should display an error that only employers can submit review.	Pending
TC-RMS-F-1.1_003	Employer submits a review with a 3-star rating and text comment.	Review is submitted with provided star rating and comment.	Pending
TC-RMS-	Employer submits a review with 5-star rating and no comment.	Review is submitted with provided star rating and no comment.	Pending

F-1.1_004			
TC-RMS-F-1.1_005	Employer submits a review with invalid rating value.	System should display an error indicating that the rating value is invalid.	Pending
TC-RMS-F-1.2_006	Employer submits a review for a developer and check the profile of developer.	Review is correctly associated with the correct developer profile.	Pending
TC-RMS-F-1.3_007	Developer attempts to review their own profile.	System displays an error message that user can't review themselves.	Pending
TC-RMS-F-1.4_008	Employer tries to review a developer they have not hired.	System displays error indicating they can only review a developer after the hiring process.	Pending
TC-RMS-F-1.4_009	Employer tries to review a developer after hiring them.	System allows the employer to submit review for the developer after hiring them.	Pending
TC-RMS-F-1.5_010	Check the developer profile with one review, it should display average rating as same as the provided rating.	Average rating is displayed for developer profile correctly after one review.	Pending
TC-RMS-F-1.5_011	Check the developer profile with multiple reviews, it should display average rating as a computed average.	Average rating is displayed for developer profile correctly after multiple reviews.	Pending
TC-RMS-F-1.6_012	Check the developer profile and verify all the individual reviews are present.	All reviews given to the developer are displayed in developer profile.	Pending
TC-RMS-F-1.7_013	Check the developer profile with reviews and reviewer information.	Reviewer's name or alias is shown with every review.	Pending
TC-RMS-F-1.8_014	Check the developer profile with reviews and their order.	Reviews are displayed in chronological order with most recent first.	Pending
TC-RMS-F-1.9_015	Employer tries to edit a review after posting it.	System does not allow employer to edit their own reviews after submission.	Pending

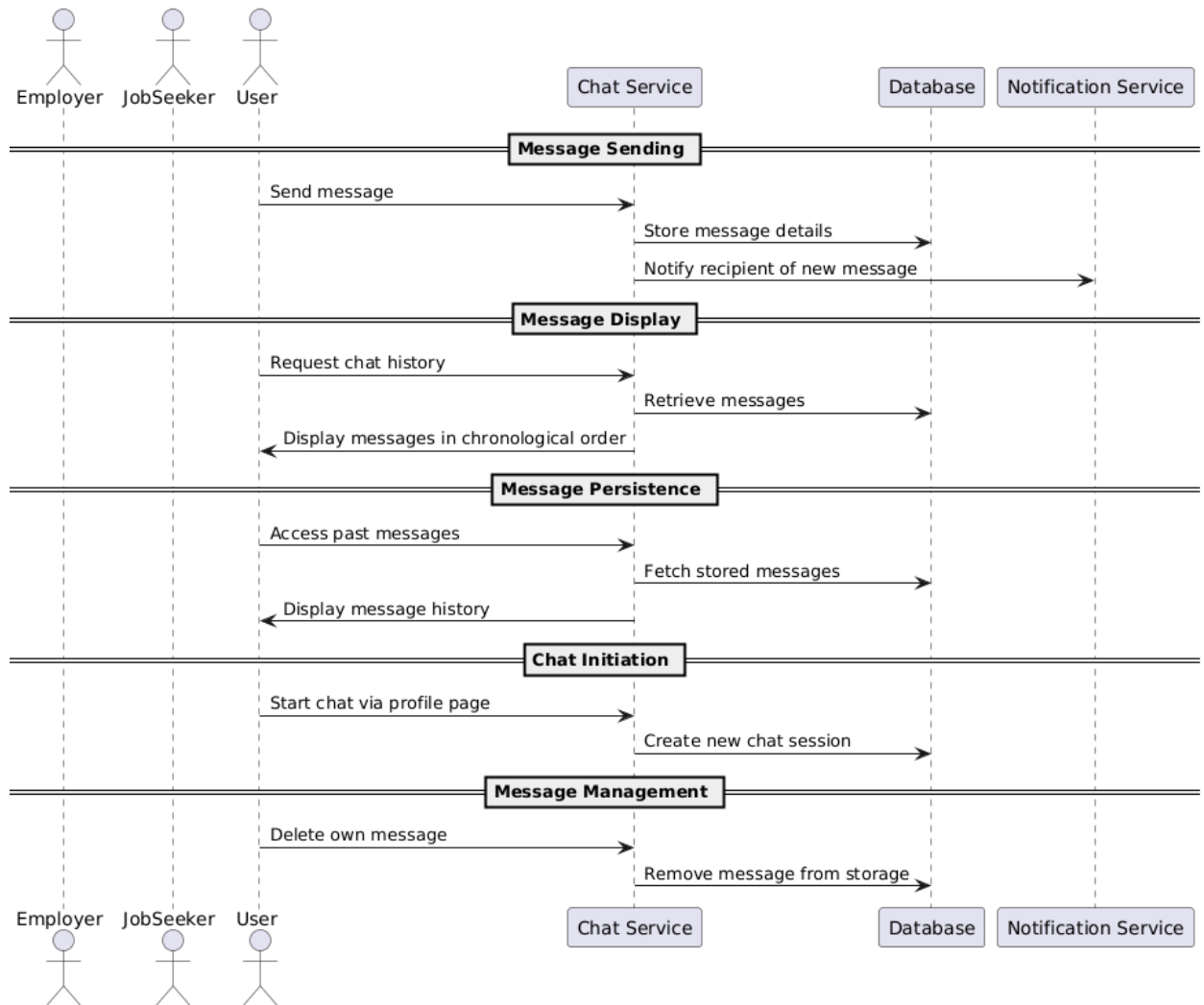
TC- RMS- F- 1.9_016	Employer tries to delete a review after posting it.	System does not allow employer to delete their own reviews after submission.	Pending
------------------------------	---	--	---------

Chat System

1. SRS

Req. Code	Req. Description	Use Case
CHAT-F-1.0	The system shall allow users to send messages to other users.	Message Sending
CHAT-F-1.1	Users shall be able to send text-based messages.	Message Sending
CHAT-F-1.2	The system shall display the sender's name or alias with each message.	Message Display
CHAT-F-1.3	The system shall display messages in chronological order (earliest first).	Message Display
CHAT-F-1.4	The system shall provide visual notifications when new messages are received.	Message Notification
CHAT-F-1.5	The system shall persist messages for future access.	Message Persistence
CHAT-F-1.6	Users shall be able to search through their message history.	Message Retrieval
CHAT-F-1.7	Users shall be able to initiate a chat with another user through their profile page.	Chat Initiation
CHAT-F-1.8	Users shall be able to delete their own messages after sending.	Message Management

2. UMLS



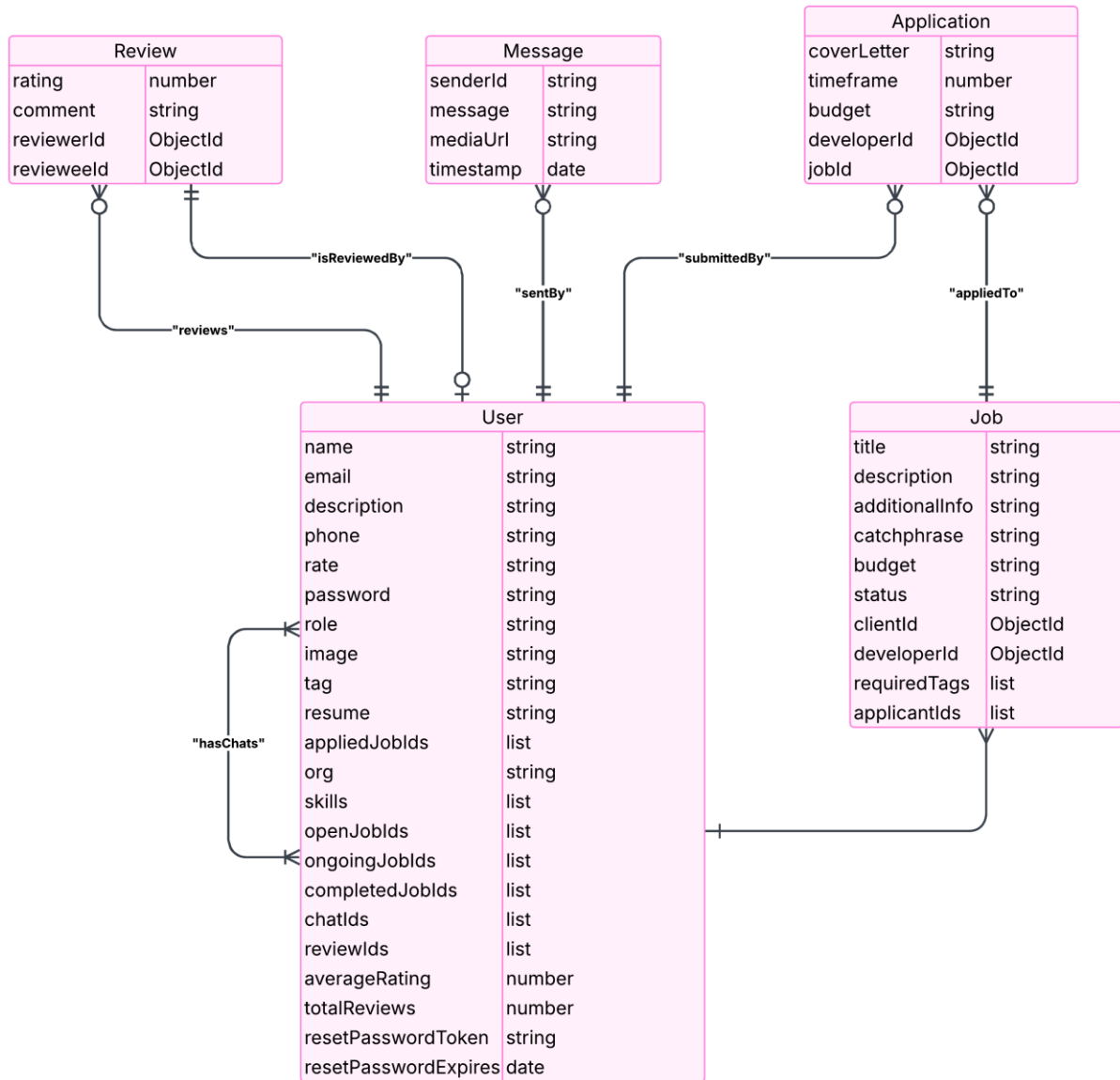
3. Testing

Req. Code	Req. Description	Test ID	Test Case Description	Expected Result	Status
CHAT-F-1.0	The system shall allow users to send messages to other users.	TC-CHAT-F-1.0_001	Send a message to another user with valid credentials and content.	Message is successfully sent to the intended recipient.	Pending
CHAT-F-1.0	The system shall allow users to send messages to other users.	TC-CHAT-F-1.0_002	Attempt to send a message to a user who doesn't exist.	System displays an appropriate error message (if applicable) or the message is not sent.	Pending

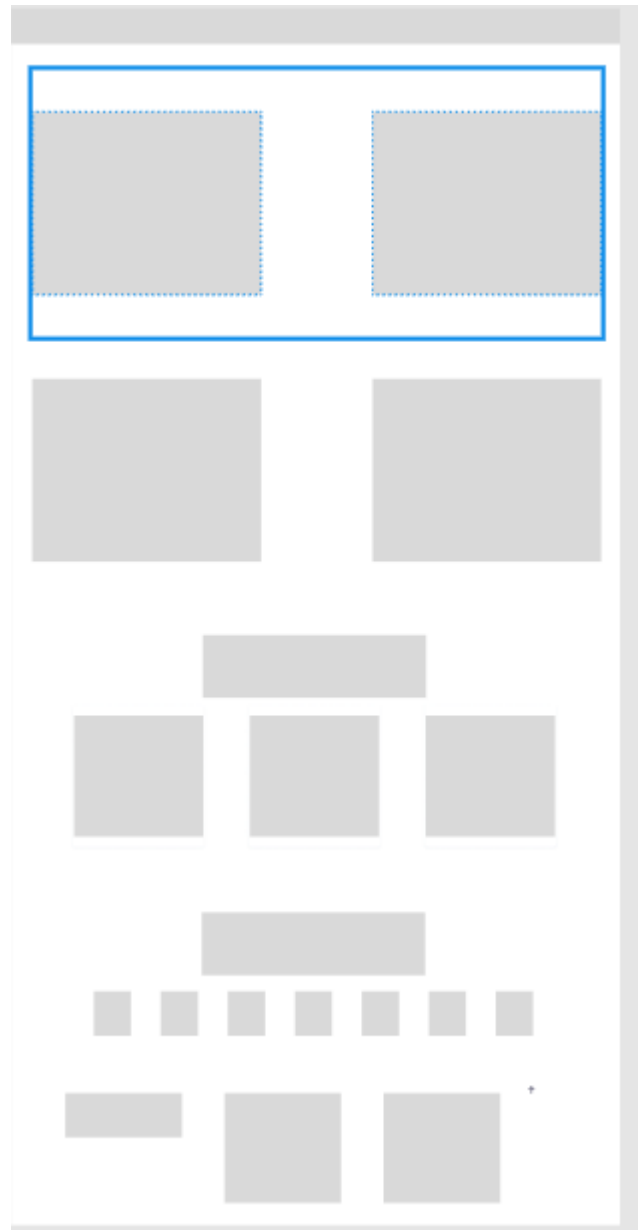
CHAT-F-1.1	Users shall be able to send text-based messages.	TC-CHAT-F-1.1_003	Send a message containing standard text characters.	Message is sent correctly.	Pending
CHAT-F-1.1	Users shall be able to send text-based messages.	TC-CHAT-F-1.1_004	Send a message containing special characters, emojis, or symbols.	Message is sent correctly.	Pending
CHAT-F-1.1	Users shall be able to send text-based messages.	TC-CHAT-F-1.1_005	Send a long message that exceeds the specified length limit.	Message is not sent or an error is displayed if the message length limit is breached.	Pending
CHAT-F-1.2	The system shall display the sender's name or alias with each message.	TC-CHAT-F-1.2_006	Verify the sender's name/alias is displayed correctly along with the message.	Sender's name/alias appears accurately next to the sent message.	Pending
CHAT-F-1.2	The system shall display the sender's name or alias with each message.	TC-CHAT-F-1.2_007	Verify the sender's name is displayed correctly even with special characters or formatting.	Sender's name/alias appears accurately next to the sent message.	Pending
CHAT-F-1.3	The system shall display messages in chronological order (earliest first).	TC-CHAT-F-1.3_008	Send multiple messages and verify they appear in the correct order.	Messages are displayed in chronological order, with the earliest at the top.	Pending
CHAT-F-1.4	The system shall provide visual notifications when new messages are received.	TC-CHAT-F-1.4_009	Receive a new message and verify visual notification is displayed (e.g., badge, sound).	Visual notification is correctly displayed/delivered for new message reception.	Pending
CHAT-F-1.4	The system shall provide visual notifications when new messages are received.	TC-CHAT-F-1.4_010	Receive multiple new messages and verify visual notification reflects the new count.	Visual notification updates with new messages.	Pending

CHAT-F-1.5	The system shall persist messages for future access.	TC-CHAT-F-1.5_011	Send messages and verify they are still available after logging out and logging back in.	Messages are persisted and accessible after logging out and back in.	Pending
CHAT-F-1.5	The system shall persist messages for future access.	TC-CHAT-F-1.5_012	Verify messages are persisted correctly even when application is restarted or closed.	Messages are persisted and accessible after application restart or close.	Pending
CHAT-F-1.6	Users shall be able to search through their message history.	TC-CHAT-F-1.6_013	Search for a specific word or phrase within message history using search function.	System correctly locates and displays messages containing the search criteria.	Pending
CHAT-F-1.6	Users shall be able to search through their message history.	TC-CHAT-F-1.6_014	Attempt to search for a term that doesn't exist in the message history.	System correctly indicates that there are no matching messages.	Pending
CHAT-F-1.7	Users shall be able to initiate a chat with another user through their profile page.	TC-CHAT-F-1.7_015	Navigate to a user profile and click on the chat option.	Chat window opens with user profile.	Pending
CHAT-F-1.8	Users shall be able to delete their own messages after sending.	TC-CHAT-F-1.8_016	Send a message and then delete it, verify the message is deleted from all recipients view.	Message is removed from the chat thread for both users after deletion.	Pending
CHAT-F-1.8	Users shall be able to delete their own messages after sending.	TC-CHAT-F-1.8_017	Send a message and verify that you can't delete the message after time specified.	System throws an error that the message cannot be deleted as the time to delete the message has lapsed.	Pending

Schema Design Diagram



Wireframes and Designs





Find Work

[Filter](#)

Showing 1–16 of 32 results

Show

16

Sort By

Default

React Portfolio Website

Verified | Kathmandu | 24 Nov

We are looking for a creative person who can redesign & build beautiful portfolio website

[React](#) [CSS](#) [JS](#) [NodeJs](#) [MongoDB](#)

\$ 25.00/hour | 10 proposals [Apply Now](#)

React Portfolio Website

Verified | Kathmandu | 24 Nov

We are looking for a creative person who can redesign & build beautiful portfolio website

[React](#) [CSS](#) [JS](#) [NodeJs](#) [MongoDB](#)

\$ 25.00/hour | 10 proposals [Apply Now](#)

React Portfolio Website

Verified | Kathmandu | 24 Nov

We are looking for a creative person who can redesign & build beautiful portfolio website

[React](#) [CSS](#) [JS](#) [NodeJs](#) [MongoDB](#)

\$ 25.00/hour | 10 proposals [Apply Now](#)

React Portfolio Website

Verified | Kathmandu | 24 Nov

We are looking for a creative person who can redesign & build beautiful portfolio website

[React](#) [CSS](#) [JS](#) [NodeJs](#) [MongoDB](#)

\$ 25.00/hour | 10 proposals [Apply Now](#)

React Portfolio Website

Verified | Kathmandu | 24 Nov

We are looking for a creative person who can redesign & build beautiful portfolio website

[React](#) [CSS](#) [JS](#) [NodeJs](#) [MongoDB](#)

\$ 25.00/hour | 10 proposals [Apply Now](#)

React Portfolio Website

Verified | Kathmandu | 24 Nov

We are looking for a creative person who can redesign & build beautiful portfolio website

[React](#) [CSS](#) [JS](#) [NodeJs](#) [MongoDB](#)

\$ 25.00/hour | 10 proposals [Apply Now](#)



123 Market St. #22B
Charlottesville, California
44635

(434) 546-4356

contact@lift.agency

[About](#)[Growers](#)[Merchants](#)[Partners](#)[Contact](#)[Facebook](#)[Twitter](#)[LinkedIn](#)[Instagram](#)

© 2020 Lift media. All rights reserved.



DEVX

[Home](#) [Find Talent](#) [Find Work](#) [Why DevX](#)



[Login](#)

[Sign Up](#)

Find Talent

Filter

Showing 1–16 of 32 results

Show

16

Sort By

Default

Aayush Tamang

Verified | Kathmandu | 24 Nov

We are looking for a creative person who can redesign & build beautiful portfolio website

[React](#) [CSS](#) [JS](#) [NodeJs](#) [MongoDB](#)

\$ 25.00/hour | 10 projects completed

[Hire Now](#)

Aayush Tamang

Verified | Kathmandu | 24 Nov

We are looking for a creative person who can redesign & build beautiful portfolio website

[React](#) [CSS](#) [JS](#) [NodeJs](#) [MongoDB](#)

\$ 25.00/hour | 10 projects completed

[Hire Now](#)

Aayush Tamang

Verified | Kathmandu | 24 Nov

We are looking for a creative person who can redesign & build beautiful portfolio website

[React](#) [CSS](#) [JS](#) [NodeJs](#) [MongoDB](#)

\$ 25.00/hour | 10 projects completed

[Hire Now](#)

Aayush Tamang

Verified | Kathmandu | 24 Nov

We are looking for a creative person who can redesign & build beautiful portfolio website

[React](#) [CSS](#) [JS](#) [NodeJs](#) [MongoDB](#)

\$ 25.00/hour | 10 projects completed

[Hire Now](#)

Aayush Tamang

Verified | Kathmandu | 24 Nov

We are looking for a creative person who can redesign & build beautiful portfolio website

[React](#) [CSS](#) [JS](#) [NodeJs](#) [MongoDB](#)

\$ 25.00/hour | 10 projects completed

[Hire Now](#)

Aayush Tamang

Verified | Kathmandu | 24 Nov

We are looking for a creative person who can redesign & build beautiful portfolio website

[React](#) [CSS](#) [JS](#) [NodeJs](#) [MongoDB](#)

\$ 25.00/hour | 10 projects completed

[Hire Now](#)



DEVX

123 Market St. #22B
Charlottesville, California
44635

(434) 546-4356

contact@lift.agency

[About](#)

[Growers](#)

[Merchants](#)

[Partners](#)

[Contact](#)

[Facebook](#)

[Twitter](#)

[LinkedIn](#)

[Instagram](#)

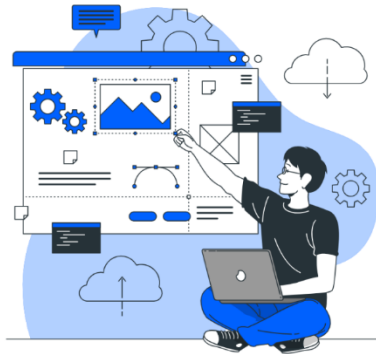


© 2020 Lift media. All rights reserved.



Work **Remotely** from your home or any other places

We curate the best digital jobs for those who are looking to start their career in designing.

[Get Started](#)

Where **Idea** meets **Talent**

We curate the best digital jobs for those who are looking to start their career in designing.

Manage your entire community in a single system

Who is Nextcent suitable for?



Membership Organisations

Our membership management software provides full automation of membership renewals and payments



National Associations

Our membership management software provides full automation of membership renewals and payments



Clubs And Groups

Our membership management software provides full automation of membership renewals and payments

Our Clients

We have been working with some Fortune 500+ clients



123 Market St. #22B
Charlottesville, California
44635

(434) 546-4356

contact@lift.agency

[About](#)
[Growers](#)
[Merchants](#)
[Partners](#)
[Contact](#)

[Facebook](#)
[Twitter](#)
[LinkedIn](#)
[Instagram](#)



© 2020 Lift media. All rights reserved.



DEVX

[Home](#) [Find Talent](#) [Find Work](#) [Why DevX](#)



[Login](#)

[Sign Up](#)

Welcome Back!

☐ Remember me

[Forgot password?](#)

[Log In](#)

 Or sign in with Google

Dont have an account? [Sign up now](#)



DEVX

123 Market St. #22B
Charlottesville, California
44635

[\(434\) 546-4356](#)

[contact@lifecore.com](#)

[About](#)
[Growers](#)
[Merchants](#)
[Partners](#)
[Contact](#)
© 2020 lift media. All rights reserved.

[Facebook](#)
[Twitter](#)
[LinkedIn](#)
[Instagram](#)



DEVX

[Home](#) [Find Talent](#) [Find Work](#) [Why DevX](#)



[Login](#)

[Sign Up](#)

Join as a client or a talent



I'm a client, hiring for a project



I'm a freelancer, looking for work

Already have an account? [Login](#)



DEVX

123 Market St. #22B
Charlottesville, California
44635

[About](#)
[Growers](#)
[Merchants](#)
[Partners](#)
[Contact](#)
© 2020 lift media. All rights reserved.

[Facebook](#)
[Twitter](#)
[LinkedIn](#)
[Instagram](#)





Create Account

First Name

Last Name

Email

Password

Create Account

Already have an account? [Login](#)

or



Sign up with Google



Sign up with Facebook



DEVX

123 Market St. #22B
Charlottesville, California
44635

[About](#)

[Growers](#)

[Merchants](#)

[Partners](#)

© 2020 Litt media. All rights reserved.

[Contact](#)

[Facebook](#)

[Twitter](#)

[LinkedIn](#)

[Instagram](#)

Work

Showing 1–16 of 32 results

**Where
Idea
meets Talent**

We curate the best digital jobs for those who are looking to start their career in design.

Freelance Platform



Verified



Kathmandu



24 Nov

We are looking for a creative person who can redesign & build beautiful portfolio website

React

CSS

JS

Node.js

MongoDB

\$ 25.00/hour | 10 proposals

[Apply Now](#)

Description

Additional Information

Reviews [5]

Embodying the raw, wayward spirit of rock 'n' roll, the Kilburn portable active stereo speaker takes the unmistakable look and sound of Marshall, unplugs the chords, and takes the show on the road.

Weighing in under 7 pounds, the Kilburn is a lightweight piece of vintage styled engineering. Setting the bar as one of the loudest speakers in its class, the Kilburn is a compact, stout-hearted hero with a well-balanced audio which boasts a clear midrange and extended highs for a sound that is both articulate and pronounced. The analogue knobs allow you to fine tune the controls to your personal preferences while the guitar-influenced leather strap enables easy and stylish travel.

**Where Idea
meets Talent**

We curate the best digital jobs for those who are looking to start their career in design.

**Where Idea
meets Talent**

We curate the best digital jobs for those who are looking to start their career in design.



123 Market St. #22B
Charlottesville, California
44635

(434) 546-4356

contact@lift.agency

[About](#)
[Growers](#)
[Merchants](#)
[Partners](#)
[Contact](#)

[Facebook](#)
[Twitter](#)
[LinkedIn](#)
[Instagram](#)



© 2020 Lift media. All rights reserved.