**SOFTWARE REQUIREMENT SPECIFICATION**

**3. SOFTWARE REQUIREMENT SPECIFICATION**

**3.1 INTRODUCTION**

Requirements specification is the starting step for the development activities.It is currently one of the weak areas of software engineering.During requirement specification, the goal is to produce a document of the client’s requirements. This document forms the basis of development and software validation.The basic reason for the difficulty in software requirements specification comes from the fact that there are three interested parties- the client, the end users and the software developer.

**3.2 PURPOSE**

The origin of most software systems is in the need of a client, who either wants to automate an existing manual system or desires a new software system. The software system itself is created by the developer. Finally, the completed system will be used by the end users. Thus, there are three major parties interested in a new system: the client, the users and the developer. A basic purpose of software requirements specification is to bridge the communication gap. SRS is the medium through which the client and user needs are accurately specified. Indeed, SRS forms the basis of software development. A good SRS should satisfy all the parties, something very hard to achieve, and involves trade-offs and persuasion. Another important purpose of developing an SRS is helping the clients understand their own needs. Advantages are:

* An SRS establishes the basis for agreement between the client and the supplier on what the software product will do
* An SRS provides a reference for validation of the final product
* A high-quality SRS is a prerequisite to high-quality software.
* A high-quality SRS reduces the development cost.

**3.3 SCOPE**

HiveMind is a website that is developed to meet the needs of small niche of people who are interested in teaching small skills and for the people who are interested to learn more on small skills and broaden their skill sets. The system will further provide to add skills to your dashboard, your skill lists, schedule classes, create skill as a Expert, and chat one-on-one Expert-learner conversation. Website admins can log in and view activity logs. Chats are supposedly going to be encrypted, which means no one other than the sender and the receiver would know about the chat, not even the administrator can access private chats.

**3.4 TECHNICAL OVERVIEW**

**3.4.1 USER CHARACTERISTICS**

The system can be accessed by three set of users, the Admin, The Experts and the Learners. The admin will have access to all the functionality of the system. The User will have access to the site and all its services. The admin dashboard is invisible to user.

**3.5 STATED REQUIREMENTS**

**3.5.1 GENERAL REQUIREMENTS**

The HiveMind platform includes 9 core functional modules used by Admins, Experts, and Learners. These modules are responsible for managing user activities, skill offerings, messaging, and session scheduling.

**1. Login**

- Only registered users and admins can log into the system to access services.  
- Login requires a valid email and password.  
- The email field must always follow standard email formatting.  
- Passwords must contain a combination of uppercase and lowercase letters, numbers, and special characters.  
- If an admin logs in using predefined credentials, they will be redirected to the admin dashboard.  
- Learners and Experts will be redirected to their respective dashboards after successful login.

**2. Sign Up**

- New users must register to use HiveMind.  
- Registration fields include:  
 • Full Name (letters only)  
 • Email Address  
 • Phone Number  
 • Password  
 • User Role (Learner or Expert)  
- Admin accounts are manually predefined and cannot be created through the signup form.

**3. Admin Panel**

The admin panel enables complete system oversight. Key admin actions include:  
- Approving or rejecting skill proposals from Experts.  
- Viewing, editing, or deleting skill listings.  
- Monitoring reviews and blocking inappropriate content.  
- Tracking activity logs and registration trends.

**4. Post Skill**

Experts can post new skills they wish to offer through this module. The form includes:  
- Skill Name  
- Skill Category  
- Description  
- Weekly Availability Schedule  
- Status (pending approval / approved)

**5. Messaging**

HiveMind includes a 1-on-1 messaging feature:  
- Learners and Experts can exchange direct messages.  
- One chat room exists per learner-Expert pair.  
- Conversations remain accessible from dashboards.  
- New message notifications are shown in real time.

**6. Profile Management**

- All users can edit their profile info (name, bio, contact).  
- Experts can add their area of expertise and upload profile pictures.  
- Admins cannot change their email as it serves as their login identifier.

**7. Dashboard**

- Learners view their active sessions, upcoming schedule, and received messages.  
- Experts see teaching schedules, performance stats (e.g., rating overview), and received messages.  
- Admins view platform activity, review moderation queue, and key system stats.

**8. Review & Feedback**

- After a session, learners can rate and review the Expert or skill.  
- Experts can view reviews received for feedback.  
- Admins can monitor and remove reviews if flagged or inappropriate.

**9. Homepage & Skill Discovery**

- Only logged-in users can access the skill listing page.  
- Learners can browse all approved skills, filter by category, and view detailed descriptions.  
- Registration is only enabled after logging in.

**3.5.2 INPUTS**

HiveMind supports dynamic and interactive skill-based learning. Key inputs include:  
- User credentials (email, password)  
- Skill proposal details (title, category, availability)  
- Reviews and ratings  
- Messages exchanged between users

**3.5.3 PROCESSING**

* All form inputs are validated on both frontend and backend.
* Admin actions include creating, editing, deleting skills, reviewing content, and managing users
* Learners can search skills, register for sessions, send messages, and leave feedback.
* Experts can create/edit skills, manage schedule, respond to learners, and track performance.
* User activity and interactions are logged for admin review.

**3.5.4 OUTPUTS**

This system produces the following outputs:

* Skill information (title, category, description, availability)
* User registrations and session details
* Reviews and rating summaries
* Real-time messaging threads
* Notifications and alerts

**3.6. EXTERNAL INTERFACE REQUIREMENTS**

**3.6.1. USER INTERFACE**

All user interfaces will be GUI interfaces. Interfaces are designed to use with ease and without any confusion. The user interface shall have a pleasing appearance and high functionality.

* Suitable design and pleasing colours are selected to design the window page to make the users comfortable to operate the software
* Component like textboxes, combo boxes and buttons are selected to make it easy to fill with appropriate data.

**3.6.2 HARDWARE INTERFACES**

The System needs a computer or any laptops or devices with network availability to browse into the web application. No other external hardware is required.

**HARDWARE SPECIFICATION**

Processor: AMD RYZEN 7000 series

RAM: 256 MB or higher

Hard disk drive: 100MB is required on disk

Keyboard: Standard QWERTY keyboard

**IMPLEMENTATION SPECIFICATION**

Operating system: Windows OS

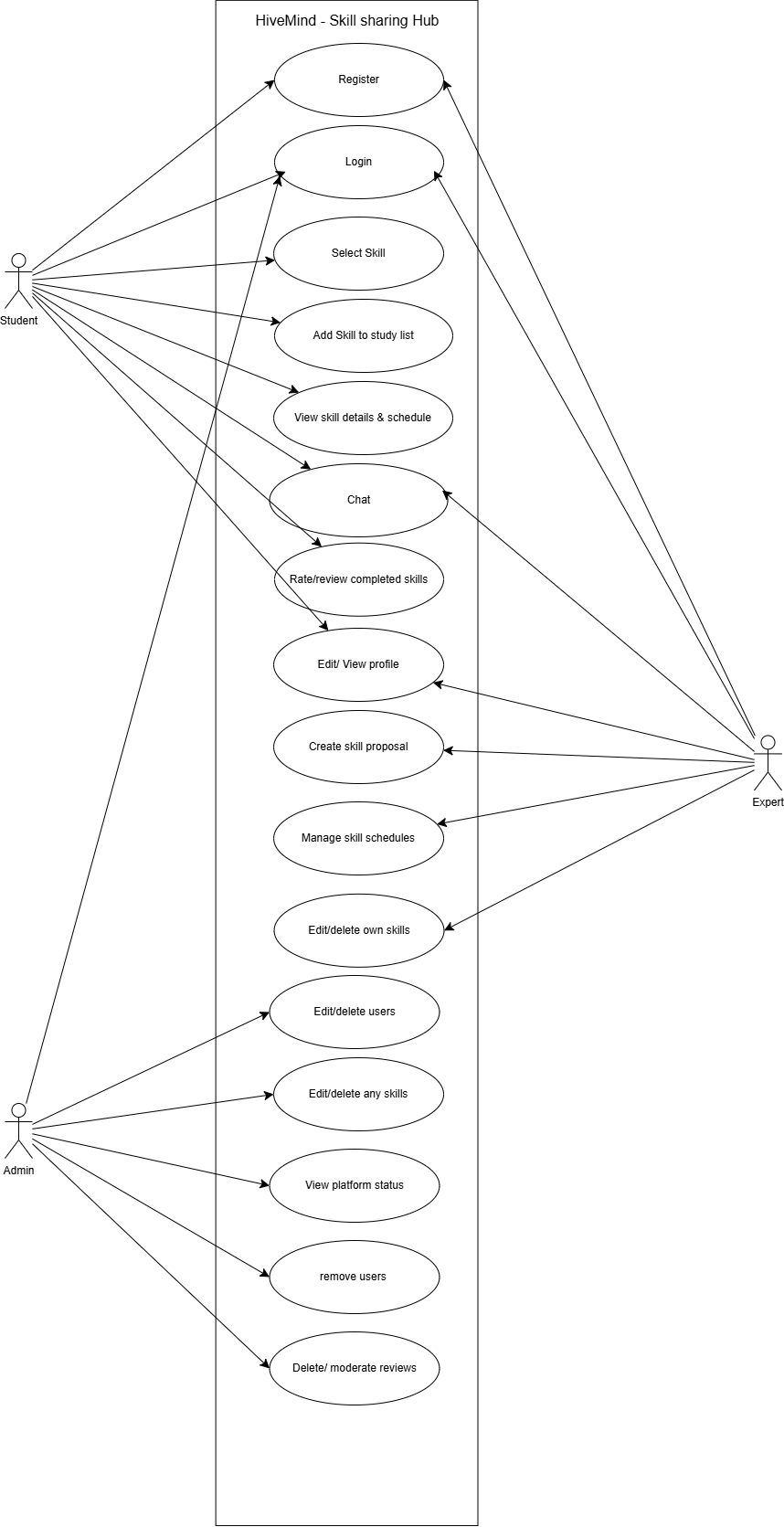
**SOFTWARE SPECIFICATION**

Operating System: Windows 11

Frontend: HTML, CSS, JavaScript

Backend: PHP, SQL

**UML DIAGRAMS**

****

**A diagram of a company

AI-generated content may be incorrect.**

**EXPERT SIDE:**

**A diagram of a flowchart

AI-generated content may be incorrect.**

**LEARNER SIDE:**

**A diagram of a diagram

AI-generated content may be incorrect.**

**4.3 SYSTEM ARCHITECTURE AND PROCESS FLOW USECASES**

**1. REGISTRATION**

|  |  |
| --- | --- |
| Use Case ID | HM\_UC\_01 |
| Use Case Name | Registration |
| Created By | Aleena Elza Mathew |
| Date Created | 02-07-2024 |
| Description | Allows new users (learners or Experts) to create an account. |
| Primary Actor | User |
| Secondary Actor | None |
| Precondition | User is on the registration page. |
| Postcondition | New account is created and stored in the system. |
| Main Flow | 1. User navigates to registration page. 2. User enters full name, email, phone, password, role. 3. System validates input. 4. Account is created with default status. 5. User redirected to login page. 6. Use case ends. |

**2. LOGIN**

|  |  |
| --- | --- |
| Use Case ID | HM\_UC\_02 |
| Use Case Name | Login |
| Created By | Aleena Elza Mathew |
| Date Created | 02-07-2024 |
| Description | This use case allows the user to login to the system using relevant data. The various user roles are admin and the customer. To login to the system the admin must enter the admin email and password and the customer must enter his/her email and password. |
| Primary Actor | User (Admin, Expert, or Learner) |
| Secondary Actor | None |
| Precondition | User already has a registered account. |
| Postcondition | User is redirected to the appropriate dashboard. |
| Main Flow | 1. User enters email and password. 2. System verifies credentials. 3. If valid, redirect to learner/Expert/admin dashboard. 4. If invalid, error message shown. 5. Use case ends. |

**3. BROWSE SKILLS**

|  |  |
| --- | --- |
| Use Case ID | HM\_UC\_03 |
| Use Case Name | Browse Skills |
| Created By | Aleena Elza |
| Date Created | 02-07-2024 |
| Description | Learners can browse available skills to learn. |
| Primary Actor | Learner |
| Secondary Actor | None |
| Precondition | User is logged in as learner. |
| Postcondition | Skill details are viewed, possibly registered. |
| Main Flow | 1. Learner logs in. 2. Navigates to Skill Page. 3. Filters by category or searches. 4. Clicks a skill to view more info. 5. Reads description, availability, Expert info. 6. Use case ends. |

**4. REGISTER FOR SKILL SESSION**

|  |  |
| --- | --- |
| Use Case ID | HM\_UC\_04 |
| Use Case Name | Register for Skill Session |
| Created By | Aleena Elza |
| Date Created | 02-07-2024 |
| Description | Learner selects a schedule and sends a registration request. |
| Primary Actor | Learner |
| Secondary Actor | Expert |
| Precondition | Learner is logged in and has browsed a skill. |
| Postcondition | Registration is stored and awaits approval. |
| Main Flow | 1. Learner views skill details. 2. Clicks "Register" button. 3. Chooses preferred schedule. 4. Confirms registration. 5. System stores request with status "pending". 6. Notification sent to Expert. 7. Use case ends. |

**5. SUBMIT SKILL PROPOSAL**

|  |  |
| --- | --- |
| Use Case ID | HM\_UC\_05 |
| Use Case Name | Submit Skill Proposal |
| Created By | Aleena Elza |
| Date Created | 02-07-2024 |
| Description | Expert proposes a new skill to be reviewed by admin. |
| Primary Actor | Expert |
| Secondary Actor | Admin |
| Precondition | Expert is logged in. |
| Postcondition | Proposal is stored and marked as "pending". |
| Main Flow | 1. Expert navigates to "Add Skill" section. 2. Enters title, category, description, availability. 3. Submits proposal. 4. System stores skill with status "pending". 5. Admin notified. 6. Use case ends. |

**6. APPROVE/REJECT SKILL**

|  |  |
| --- | --- |
| Use Case ID | HM\_UC\_06 |
| Use Case Name | Approve/Reject Skill |
| Created By | Aleena Elza |
| Date Created | 02-07-2024 |
| Description | Admin moderates skill proposals from Experts. |
| Primary Actor | Admin |
| Secondary Actor | Expert |
| Precondition | Skill is submitted and status is "pending". |
| Postcondition | Skill is marked as approved or rejected. |
| Main Flow | 1. Admin logs in. 2. Views all pending skills. 3. Reviews details. 4. Clicks Approve or Reject. 5. System updates status. 6. Expert notified. 7. Use case ends. |

**7. MESSAGING**

|  |  |
| --- | --- |
| Use Case ID | HM\_UC\_07 |
| Use Case Name | Messaging |
| Created By | Aleena Elza |
| Date Created | 02-07-2024 |
| Description | Enables direct messaging between learner and Expert. |
| Primary Actor | Learner, Expert |
| Secondary Actor | None |
| Precondition | Both users are logged in. |
| Postcondition | Message is stored and displayed in chat room. |
| Main Flow | 1. Learner opens Expert profile. 2. Clicks on "Message" button. 3. Chat room opens or is created. 4. Message is typed and sent. 5. System stores message. 6. Expert receives it in dashboard. 7. Use case ends. |

**8. LEAVE A REVIEW**

|  |  |
| --- | --- |
| Use Case ID | HM\_UC\_08 |
| Use Case Name | Leave a Review |
| Created By | Aleena Elza |
| Date Created | 02-07-2024 |
| Description | Learner submits a review and rating after session. |
| Primary Actor | Learner |
| Secondary Actor | Expert |
| Precondition | A session has been attended. |
| Postcondition | Review is stored and visible to other users. |
| Main Flow | 1. Learner opens skill or Expert page. 2. Clicks on "Leave a Review". 3. Enters rating and optional comment. 4. Submits the review. 5. System stores it with timestamp. 6. Use case ends. |

**TABLE DESIGN**

1. Table name: users

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Sl.no** | **Field Name** | **Data Type** | **Constraint** | **Description** |
| **1** | User\_id | int | Primary Key, Auto\_increment,  Not null | User ID |
| **2** | email | Varchar(100) | Not null | Email of the user |
| **3** | password | Varchar(255) | Not null | Password of the user (encrypted) |
| **4** | role | enum | (‘learner’, ‘expert’, ‘admin’) default ‘learner’ | Role of the user: Learner, Expert, Admin |
| **5** | status | enum | (‘active’, ‘inactive’) | Status of the account |
| **6** | Created\_at | timestamp | Default current timestamp | Timestamp of when the account was created |
| **7** | Updated\_at | timestamp | null | Timestamp of the last update on the account |

1. Table name: user\_profiles

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **sl.no** | **Field Name** | **Data Type** | **Constraints** | **Description** |
| 1 | profile\_id | INT | Primary Key, Auto\_increment, Not null | Unique profile ID |
| 2 | user\_id | INT | Foreign Key → users(user\_id), Not null | Links profile to user |
| 3 | full\_name | VARCHAR(100) | Not null | Full name of user |
| 4 | Bio | TEXT | Optional | Short description or bio |
| 5 | DOB | date | Not Null | Date of birth |
| 6 | phone\_number | VARCHAR(15) | Unique, Optional | Contact number |
| 7 | profile\_picture | VARCHAR(255) | Optional | Path to profile picture |
| 8 | gender | Char(1) | Not Null | Gender of the user |

1. Table Name: skills

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Sl.no | Field Name | Data Type | Constraints | Description |
| 1 | skill\_id | INT | Primary Key, Auto\_increment, Not null | Unique ID for each skill |
| 2 | Expert\_id | INT | Foreign Key → users(user\_id) | Owner of the skill |
| 3 | title | VARCHAR(100) | Not null | Name of the skill |
| 4 | category | VARCHAR(100) | Not null | Skill category |
| 5 | description | TEXT | Not null | Detailed skill description |
| 6 | is\_approved | BOOLEAN | Default FALSE | Admin approval status |
| 7 | created\_at | TIMESTAMP | Default CURRENT\_TIMESTAMP | Time of skill creation |

1. Table name: Schedule

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Sl.no | Field Name | Data Type | Constraints | Description |
| 1 | schedule\_id | INT | Primary Key, Auto\_increment, Not null | Unique schedule ID |
| 2 | skill\_id | INT | Foreign Key → skills(skill\_id) | Skill being scheduled |
| 3 | day\_of\_week | ENUM | Not null | Day of the session (Mon–Sun) |
| 4 | start\_time | TIME | Not null | Starting time |
| 5 | end\_time | TIME | Not null | Ending time |

1. Table Name: registration

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Sl.no | Field Name | Data Type | Constraints | Description |
| 1 | registration\_id | INT | Primary Key, Auto\_increment, Not null | Unique ID for each registration |
| 2 | learner\_id | INT | Foreign Key → users(user\_id) | Learner registering for a skill |
| 3 | skill\_id | INT | Foreign Key → skills(skill\_id) | Skill being registered |
| 4 | schedule\_id | INT | Foreign Key → skill\_schedules(schedule\_id) | Chosen time slot |
| 5 | status | ENUM | Default 'pending' | Registration approval status |
| 6 | registered\_at | TIMESTAMP | Default CURRENT\_TIMESTAMP | Time of registration |

1. Table Name: reviews

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Sl.no** | **Field Name** | **Data Type** | **Constraints** | **Description** |
| 1 | review\_id | INT | Primary Key, Auto\_increment, Not null | Unique review ID |
| 2 | learner\_id | INT | Foreign Key → users(user\_id) | Reviewer ID |
| 3 | skill\_id | INT | Foreign Key → skills(skill\_id) | Reviewed skill |
| 4 | rating | INT | Not null | Numeric rating (1 to 5) |
| 5 | comment | TEXT | Optional | Text feedback |
| 6 | created\_at | TIMESTAMP | Default CURRENT\_TIMESTAMP | Time of review |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Sl.no | Field Name | Data Type | Constraints | Description |
| 1 | request\_id | INT | Primary Key, Auto\_increment, Not null | Unique ID for request |
| 2 | Expert\_id | INT | Foreign Key → users(user\_id) | Expert making the request |
| 3 | proposed\_title | VARCHAR(100) | Not null | Title of proposed skill |
| 4 | proposed\_description | TEXT | Not null | Skill details |
| 5 | status | ENUM | Default 'pending' | Request approval status |
| 6 | submitted\_at | TIMESTAMP | Default CURRENT\_TIMESTAMP | Submission time |

1. Table Name: expert\_skill\_requests

1. Table Name: activity\_log

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Sl.no** | **Field Name** | **Data Type** | **Constraints** | **Description** |
| 1 | Log\_id | INT | Primary Key, Auto\_increment, Not null | Unique ID for activity log |
| 2 | User\_id | INT | Foreign Key → users(user\_id) | User doing the activity |
| 3 | Action | VARCHAR(255) | Not null | Title of the activity |
| 4 | Details | TEXT | Not null | Description of the activity |
| 5 | Performed\_at | TIMESTAMP | Default CURRENT\_TIMESTAMP | Submission time |

1. Table Name: chatrooms

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Sl.no | Field Name | Data Type | Constraints | Description |
| 1 | chat\_room\_id | INT | Primary Key, Auto\_increment, Not null | Unique ID for each chat room |
| 2 | participant\_one\_id | INT | Foreign Key → users(user\_id) | One user in the chat |
| 3 | participant\_two\_id | INT | Foreign Key → users(user\_id) | The other user |
| 4 | created\_at | TIMESTAMP | Default CURRENT\_TIMESTAMP | Chat room creation time |

1. Table Name: messages

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Sl.no | Field Name | Data Type | Constraints | Description |
| 1 | message\_id | INT | Primary Key, Auto\_increment, Not null | Unique ID for message |
| 2 | chat\_room\_id | INT | Foreign Key → chat\_room(chat\_room\_id) | Chat room ID |
| 3 | sender\_id | INT | Foreign Key → users(user\_id) | Sender of the message |
| 4 | message\_text | TEXT | Not null | Message content |
| 5 | sent\_at | TIMESTAMP | Default CURRENT\_TIMESTAMP | When the message was sent |
| 6 | is\_read | BOOLEAN | Default false | Whether message has been read or not |

1. Table Name: notifications

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Sl.no** | **Field Name** | **Data Type** | **Constraints** | **Description** |
| 1 | notification\_id | INT | Primary Key, Auto\_increment, Not null | Notification ID |
| 2 | user\_id | INT | Foreign Key → users(user\_id) | User receiving the notification |
| 3 | type | ENUM | Not null | Notification type |
| 4 | message | TEXT | Not null | Notification text |
| 5 | is\_read | BOOLEAN | Default false | Read/unread status |
| 6 | created\_at | TIMESTAMP | Default CURRENT\_TIMESTAMP | Notification time |