Project2 - Skill Xchange

Problem Description (*Noun, Verb*):

Learning new skills can be expensive, time-consuming, and difficult without proper guidance. Many individuals have skills they are willing to share but lack an efficient platform to connect with others for mutual learning. Traditional learning methods often require financial investment, which may not be feasible for everyone. The Skill Swap Marketplace aims to address this issue by providing a platform where users can exchange skills without monetary transactions, fostering a collaborative and accessible learning environment.

Issues & Challenges:

- **Lack of Accessible Learning Opportunities**: Not everyone can afford paid courses or private lessons.
- **Difficulty in Finding Skill Exchange Partners**: There is no structured system to match people with complementary skill needs.
- **Quality Control**: Users may be concerned about the reliability and proficiency of those offering skills.
- **Communication Barriers**: Users need an efficient way to connect and arrange skill swaps.
- **User Engagement & Retention**: Encouraging consistent participation and preventing drop-off after initial usage.
- **Security & Privacy**: Ensuring safe interactions between users while preventing misuse of the platform.

Rules

- Users should be able to provide reviews after sharing the skill.
- Badges earned should be allowed to be transferred across accounts
- Users must register with valid email and password(Firebase Authentication).
- Profile setup must include at least one skill before accessing the marketplace.
- Users can offer a skill or request a skill.
- Users cannot post duplicate skills.
- After a swap, both users must rate each other on a 1-5 star scale.
- Completed swaps are stored but cannot be edited.
- Skill posts must be automatically archived after 30 days if no swap occurs.

Nouns:

- New Skills
- Individuals
- Efficient Platform
- Others
- Mutual Learning
- Financial Investment
- Platform
- Monetary Transactions
- Learning environment
- Paid Courses
- Private Lessons
- Skill Exchange Partners
- System
- People
- Users
- Reliability
- Proficiency
- Communication barrier
- Efficient Way
- Skill Swaps
- User Engagement
- Retention
- Participation
- Drop off
- Initial Usage
- Security
- Privacy
- Safe Interactions
- Misuse
- Issue

Verbs:

- learn
- Be (Time consuming, Expensive, Difficult)
- Have (skills)
- Guidance
- Share
- Connect
- Exchange
- Collaborative
- Accessible

- Afford
- Finding
- Match
- Concerned
- Offering
- Arrange
- Preventing
- Ensuring
- Address

Summary of Functions, Attributes and Associations

1. User Module

Attributes:

- user_id (Unique identifier)
- name
- age
- Email
- Location
- SkillsOffered
- SkillsRequested
- Rating

Methods:

- registerUser(userData)
- getUserProfile(userID)
- updateUserProfile(userID, updatedData)
- getUserSkills(userID)

Associations:

- One User can create multiple Skill Listings.
- One User can send and receive Swap Requests.
- One User can receive multiple Ratings and Reviews.
- One User can be reported multiple times in Security and Moderation.
- One User can send and receive multiple Messages.

2. Skill Listing Module

Attributes:

- skill_id
- skill_name
- age
- email_proficiency_level
- description
- mode

Methods:

- addSkillListing(userID, skillData)
- getAllSkills()
- getSkillsByUser(userID)
- searchSkills(query)
- deleteSkillListing(skillID)

Associations:

- One **Skill Listing** belongs to one **User**.
- One **Skill Listing** can be included in multiple **Swap Requests**.

3. Swap Request Module

Attributes:

- request_id
- requestor_id
- provider_id
- skill_id
- status

Methods:

- createSwapRequest(requestData)
- getSwapRequestsForUser(userID)
- updateSwapStatus(requestID, newStatus)
- cancelSwapRequest(requestID)

Associations:

- One **Swap Request** is made by one **User** (requestor).
- One **Swap Request** is sent to one **User** (provider).
- One **Swap Request** is linked to one **Skill Listing**.

4. Ratings and Reviews Module

Attributes:

- review_id
- reviewer_id
- reviewee_id
- rating
- comment

Methods:

- addReview(reviewData)
- getReviewsForUser(userID)
- getAverageRating(userID)

Associations:

- One **User** can receive multiple **Ratings and Reviews**.
- One **User** can write multiple **Ratings and Reviews**.

5. Security and Moderation Module

Attributes:

- report_id
- reported_user_id
- reason
- status

Methods:

- reportUser(reportData)
- getReports()
- resolveReport(reportID)

Associations:

- One **User** can be reported multiple times.
- One **Report** is linked to one **User** (reported).

6. Messaging Module (If time permits)

Attributes:

- message_id
- sender_id

- receiver_id
- timestamp
- content

Methods:

- sendMessage(senderID, receiverID, content)
- getMessagesForUser(userID)
- deleteMessage(messageID)

Associations:

- One **User** can send multiple **Messages**.
- One **User** can receive multiple **Messages**.

Challenge Questions

- How will you prevent users from spamming skill requests to multiple users?
- How will you handle disputes if a skill swap is not fulfilled by one party?
- How will you encourage users to leave honest ratings and prevent review manipulation?
- How will you ensure that inactive users do not clutter the skill listings?
- What incentive mechanisms (e.g., badges, leaderboards) would be most effective in keeping users engaged?

Target Audience

1. **Students & Lifelong Learners** – Those looking to gain knowledge in subjects such as coding, languages, music, or academic tutoring without financial barriers.
2. **Freelancers & Professionals** – Individuals who want to enhance their skills or trade industry knowledge to stay competitive.
3. **Hobbyists & Enthusiasts** – People interested in learning or teaching creative skills like painting, photography, or fitness training.
4. **Career Changers** – Individuals seeking to acquire new skills for a career shift without investing in expensive courses.
5. **Language Learners** – Users looking for language exchange partners to practice speaking and cultural understanding.
6. **Community-Oriented Individuals** – People who believe in knowledge-sharing and want to build a network based on skill exchange.

User Personas and User Stories

User Persona 1: Melissa

Age: 29
Location: New York, USA
Occupation: Freelance Graphic Designer
Background: Melissa is an experienced graphic designer who works independently and wants to expand her skills in digital marketing to promote her services. She enjoys learning online but prefers personalized learning experiences through hands-on practice.
Motivation: Melissa wants to exchange her design knowledge with someone who can teach her digital marketing techniques without spending money on expensive courses.
Pain Points:

- Difficulty finding trustworthy partners for mutual learning.
- Uncertainty about the skill proficiency of others.

User Stories for Melissa:
1. As Melissa, I want to register and set up my profile so that I can list my graphic design skills and request digital marketing skills.
2. As Melissa, I want to browse available skill listings so that I can identify potential matches for a skill swap.
3. As Melissa, I want to send a swap request to someone with digital marketing expertise so that I can propose a mutual learning exchange.

User Persona 2: Arjun

Age: 34
Location: Bangalore, India
Occupation: Software Engineer
Background: Arjun is a backend developer with expertise in Node.js and Python. He is interested in learning Spanish to improve his language skills for future travel and work opportunities. He works full-time and prefers learning in short, focused sessions.
Motivation: Arjun wants to learn Spanish conversationally from a native speaker while teaching coding basics to others who are interested.

Pain Points:

- Lack of time to enroll in traditional language classes.
- Difficulty in finding reliable partners to practice consistently.

User Stories for Arjun:
1. As Arjun, I want to set up my profile with my skills in coding and my interest in learning Spanish so that I can start receiving swap requests.

2. As Arjun, I want to send a swap request to a native Spanish speaker so that I can practice speaking with a fluent partner.
3. As Arjun, I want to add my skills and be able to edit it any time.

Usecase:

1. **User Registration & Authentication**
   - Actors: New User, Firebase Authentication
   - Description: A new user registers using email/password or social login. Firebase Authentication handles user verification.
   - Steps:
     1. User provides registration details.
     2. System validates input and stores user data in Firestore.
     3. User receives a confirmation email (if email-based authentication is used).
     4. Upon confirmation, the user can log in.

2. Profile Management

   - Actors: Registered User, Firestore
   - Description: Users can create and update their profile, including skills they offer and skills they want to learn.
   - Steps:
     1. User navigates to the profile page.
     2. User adds or updates skills, experience level, and preferred mode (virtual/in-person).
     3. System updates Firestore with changes.

3. Skill Listing & Browsing

   - Actors: Registered User, Firestore
   - Description: Users can browse skills offered by others and filter based on category, skill level, and mode.
   - Steps:
     1. User searches or filters available skills.
     2. System fetches relevant results from Firestore.
     3. User clicks on a skill listing to view more details.

4. Skill Swap Request

   - Actors: Skill Seeker, Skill Provider, Firestore
   - Description: Users can request skill swaps by contacting other users.
   - Steps:
     1. Skill seeker selects a skill and initiates a request.
     2. System notifies the skill provider.
     3. Skill provider accepts or rejects the request.
     4. If accepted, a skill swap session is arranged.

5. Messaging System

- Actors: Skill Seeker, Skill Provider, Firestore
- Description: Users can communicate within the platform to finalize skill swap details.
- Steps:
    1. User sends a message to another user.
    2. System stores messages in Firestore.
    3. Recipient receives a notification and replies.

6. Ratings & Reviews

- Actors: Skill Swappers, Firestore
- Description: Users can leave ratings and reviews after a skill swap.
- Steps:
    1. After completing a skill swap, the system prompts users to rate each other.
    2. User submits a rating and review.
    3. System updates the skill provider's profile with new feedback.

UML Diagram

SkillListing

-skill_id
-skill_name
-age
-email_proficiency_level
-description
-mode

+ addSkillListing(userID, skillData) :
Promise<Skill>
+ getAllSkills() : Promise<[Skills]>
+ getSkillsByUser(userID): Promise<[Skills]>
+ searchSkills(query) : Promise<[Skills]>
+ deleteSkillListing(skillID) : Promise<void>

Assumption that the user has to list atleast 1 skill while they create an account

Review
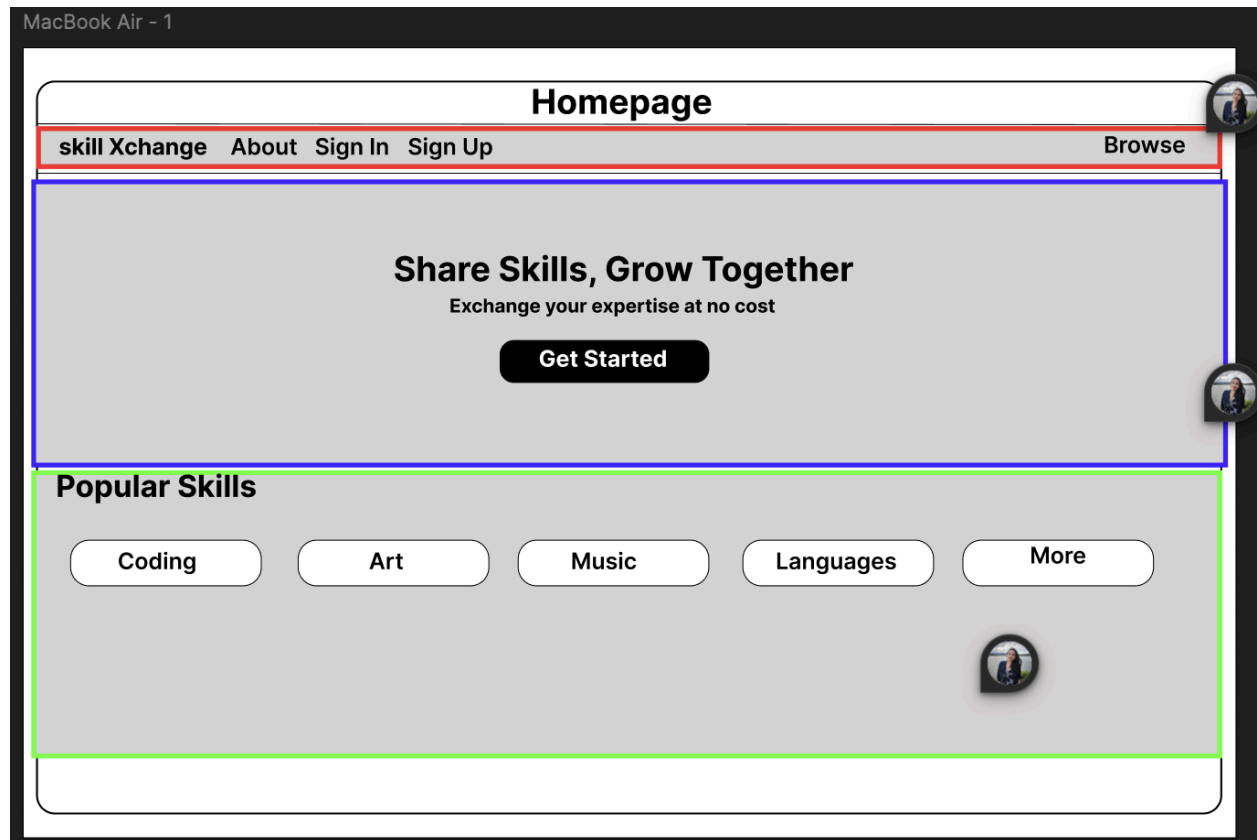
-review_id
-reviewer_id
-reviewee_id
-rating
-comment

+ addReview(reviewData) : Promise<Review>
+ getReviewsForUser(userID) :
Promise<[Review]>
+ getAverageRating(userID): Promise<number>

Own←

Reviews↑

Affects↑

1..*

0..*

0..*

User

-user_id : String
-name : String
-age : String
-email : Int
-location: String
-skills_offered : String
-skills_requested: List<Pets>
-rating

+ registerUser(userData) : Promise<User>
+ getUserProfile(userID) : Promise<User>
+ updateUserProfile(userID, updatedData)
: Promise<void>
+ getUserSkills(userID) : Promise<[Skills]>

1

1

1

1

Report

-report_id
-reported_user_id
-reason
-status

+ reportUser(reportData) : Promise<void>
+ getReports(): Promise<Report>
+ resolveReport(reportID): Promise<void>

Can report

0..*

Requests↓

Multiple users can raise swap requests but given swap request will be unique to 1 user

0..*

SwapRequest

-request_id
-requestor_id
-provider_id
-skill_id
-status

+ createSwapRequest(requestData) :
Promise<[Request]>
+ getSwapRequestsForUser(userID) :
Promise<[Request]>
+ updateSwapStatus(requestID, newStatus):
Promise<void>
+ cancelSwapRequest(requestID) :
Promise<void>

Interface Mockup

Homepage

skill Xchange   About   Sign In   Sign Up   **Browse**

Share Skills, Grow Together
Exchange your expertise at no cost

Get Started

Popular Skills

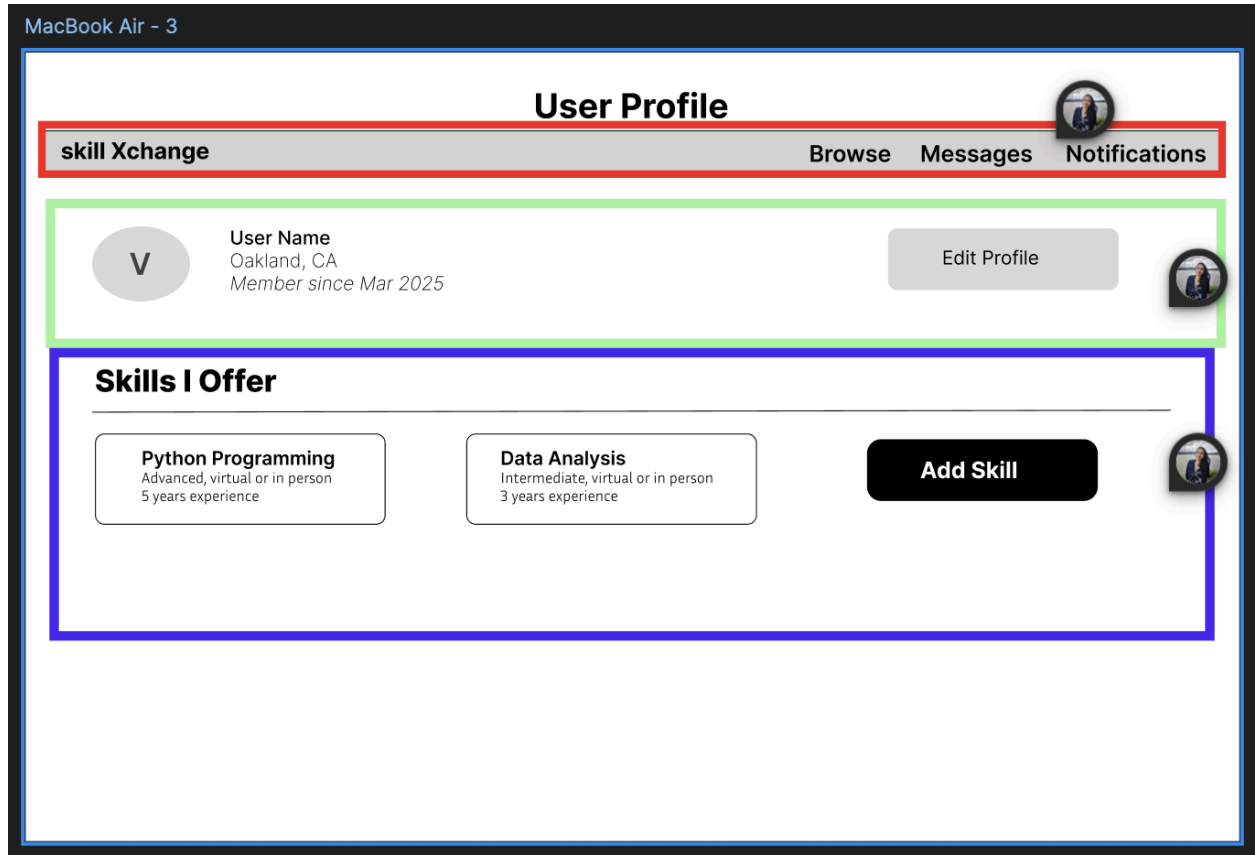| Coding | Art | Music | Languages | More |

Registration

Email

Value

Password

Value

Register

Already have an account? Log in

Melissa UserStory 1: As Melissa, I want to register and set up my profile so that I can list my graphic design skills and request digital marketing skills.
Arjun UserStory 3 : As Arjun, I want to add my skills and be able to edit it any time.
Arjun UserStory 2: As Arjun, I want to set up my profile with my skills in coding.

MacBook Air - 3

User Profile

skill Xchange   Browse   Messages   Notifications

V   **User Name**
     Oakland, CA
     Member since Mar 2025   Edit Profile

Skills I Offer

Python Programming   **Data Analysis**
Advanced, virtual or in person   Intermediate, virtual or in person   **Add Skill**
5 years experience   3 years experience

Melissa UserStory 2: As Melissa, I want to browse available skill listings so that I can identify potential matches for a skill swap.
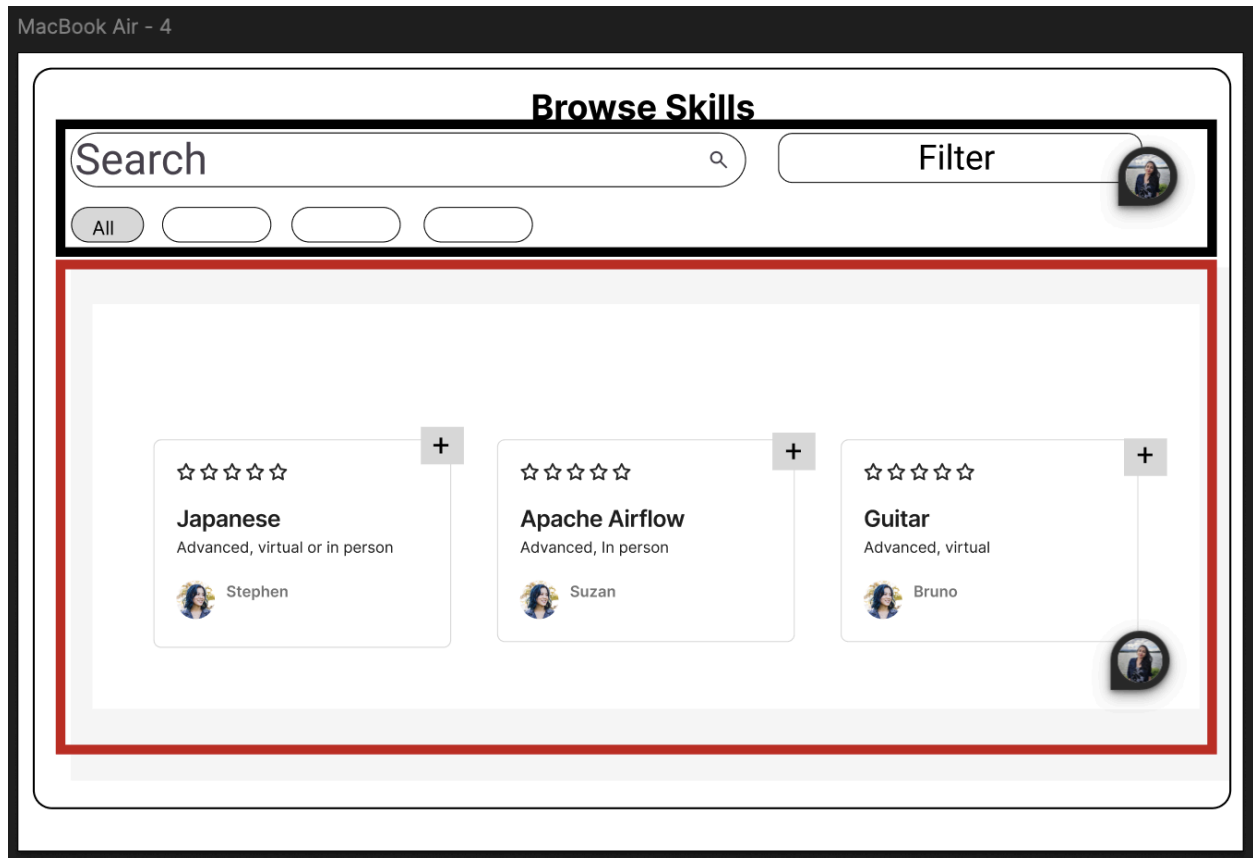
Melissa UserStory 3: I want to send a swap request to someone with digital marketing expertise so that I can propose a mutual learning exchange.

Arjun UserStory 3: As Arjun, I want to send a swap request to a native Spanish speaker so that I can practice speaking with a fluent partner.
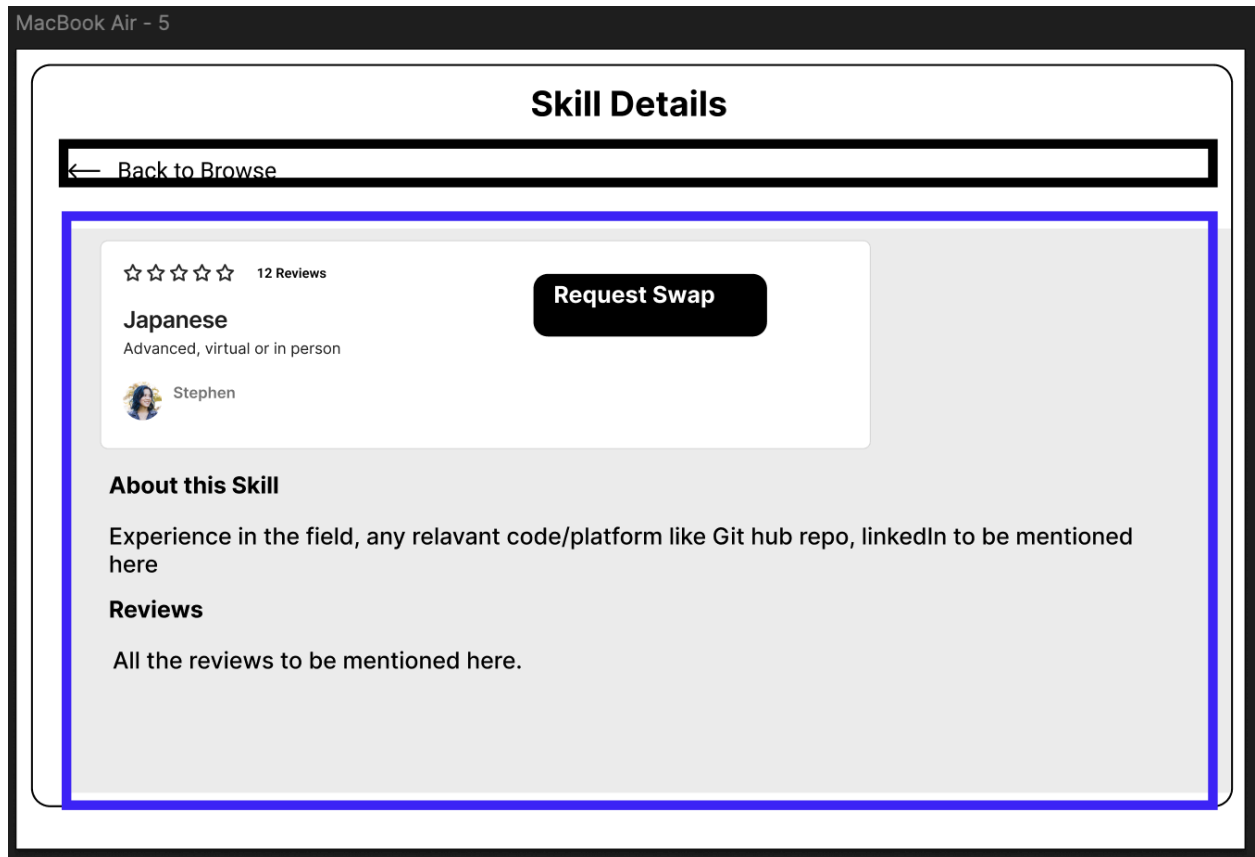
Agent: Chat gpt

Version : 4o

Prompt 1 : What is the difference between Module diagram and class diagram.

Response :

The **key difference** between a **module diagram** and a **class diagram** is how they represent the system:

Feature	Module Diagram	Class Diagram
Focus	High-level architecture and component interactions	Detailed structure of classes, attributes, and methods
Representation	Functional units (e.g., User Module, Skill Listing Module)	Object-oriented classes (e.g., User, Skill, Request)
Purpose	Shows how different modules interact	Defines relationships and behaviors of individual objects
Granularity	Broad overview, showing connections between components	Detailed view of class attributes, methods, and associations
Used In	System design, architecture planning	Object-oriented programming, database schema design

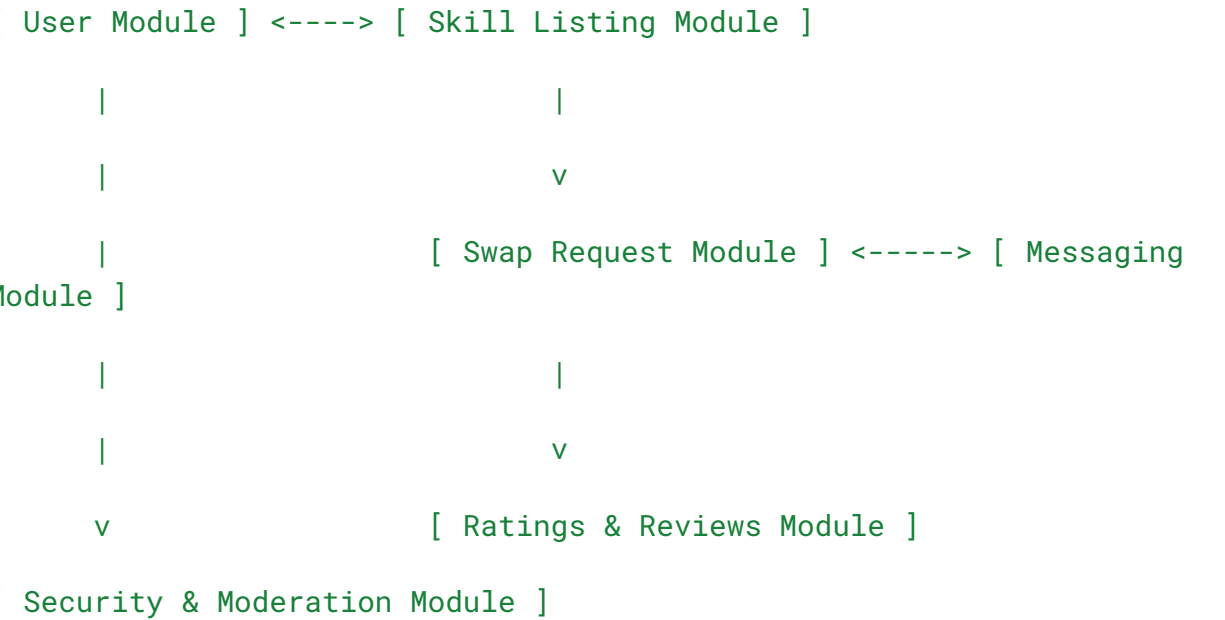| **Example Elements**	Modules, components, dependencies, API connections	Classes, attributes, methods, inheritance, associations

---

Example Comparison

Module Diagram (High-Level)

Shows how the main modules interact:

sql

CopyEdit

```
[ User Module ] <----> [ Skill Listing Module ]

        |                         |

        |                         v

        |              [ Swap Request Module ] <-----> [ Messaging
Module ]

        |                         |

        |                         v

        v              [ Ratings & Reviews Module ]

[ Security & Moderation Module ]
```

Class Diagram (Detailed)

Defines object structures and relationships:

diff

CopyEdit

```
Class: User
```

```
Attributes:

- user_id

- name

- email

- skills_offered

- skills_requested


Methods:

- registerUser()

- getUserProfile()

- updateUserProfile()


Associations:

- One user can create multiple skill listings.

- One user can send multiple swap requests.
```

Which One to Use?

- **Use a Module Diagram** when designing the overall architecture of your React + Firebase app.
- **Use a Class Diagram** when defining data models and implementing business logic.

Would you like a **visual representation** of a module diagram for your project? I can generate one for you! 🚀

Prompt 2: This is my project rubric. Can you compare it against my document to see if I have covered everything ? Rubric: Business Requirements Nouns-verbs Target audience Use Cases Target Audience Interface mockups Modules Diagram: a relaxed class like diagram that shows the components and their interactions. You won't have inheritance, but will have aggregations and associations.

Response :

Comparison of Your Project Document Against the Rubric

Here's an evaluation of whether your Project 2 - Skill Xchange document covers all rubric requirements:

Rubric Criteria	Covered in Document?	Comments
Business Requirements	✅ Yes	Clearly defined in the Problem Description and Issues & Challenges sections.
Nouns-Verbs Identification	✅ Yes	Listed comprehensively under "Nouns" and "Verbs."
Target Audience	✅ Yes	Well-defined, covering students, professionals, hobbyists, career changers, etc.
Use Cases	✅ Yes	Includes user registration, profile management, skill listing, swap requests, messaging, and reviews.
Interface Mockups	❌ No	The document does not seem to include interface mockups. Consider adding wireframes or UI sketches.

Modules Diagram   ⚠️ Partially   The document details module associations, but an explicit visual diagram is missing. You should include a UML or module diagram.

Recommendations for Missing Parts

1. Interface Mockups
   - Add a basic wireframe/UI sketch showing how users will interact with the platform (e.g., landing page, skill swap request form, chat interface).
   - Use tools like Figma, Balsamiq, or draw.io, or even a simple hand-drawn UI design.
2. Modules Diagram
   - You have the module descriptions and associations, but it would be beneficial to add a visual module diagram (UML-style with relationships).
   - If you'd like, I can generate a module diagram for you.

Would you like help creating interface mockups or a modules diagram? 🚀