Acknowledgement

We, as a team, express our sincere gratitude to Cardiff Metropolitan University in providing a very detailed syllabus that prepared us for our assessments. Furthermore, we are also grateful to our lecturer Mr. Gajindu Bandara for his support and guidance throughout this entire module. His attentive feedback regrading our project was crucial in making it work to meet the standards set by the University.

Contributions made by the entire team was crucial to have completed the project on time. We would also like to acknowledge the open-source communities whose heartful contributions have been a key factor in us developing this web application.

Chapter 01

Introduction

Sri Lankan University students face difficulties in attaining non-academic and soft skills due to a traditional exam driven curricula being practiced in Sri Lanka. Skills that are crucial to increase chances of employability and academic success like communication, collaboration and critical thinking are not competent enough according to reports (WBG16). A theoretical and exam-driven curricula proves to be a disadvantage in professional settings in Sri Lanka based on a report from World Bank Group (WBG14). Moreover, Sri Lankans also face difficulties in language competency and digital literacy (Silva25).

SkillShare resolves the limitations in traditional education by enabling students to share their skills without any financial barriers while increasing efficiency by adding synchronous features like video conferencing and planners etc. Furthermore, this web app is designed to promote lifelong learning and skill building among an attentive community (UNESCO22).

The development of SkillShare began after the proposal was approved and this report serves as the final thesis for SkillShare in addition to the interim report. The intended users and stakeholders for this web app are students and system administrators.

Problem Statement

The common problems faced by Sri Lankan students in acquiring non-academic skills are financial barriers and peer-based learning opportunities. Due to the lack of extensive support from universities, students rely on MOOCs, which are certificate based and costly. Finance is a heavily discouraging factor for students in pursuing external skill building opportunities which also heavily impacts practical experience. Tuition exists as an alternative but can be costly and scarce (Dewasiri25).

YouTube and peer study are viable alternatives to traditional curricula, but they can be both unstructured and lack two-way participation. Students tend to go to colleges and institutions to learn skills or lese they opt for informal learning due to minimal acceptance of Sri Lankan students by universities.

SkillShare addresses these issues by allowing students to share and learn skills by removing all financial barriers and providing structured learning pathways with peer-to-peer collaboration. An ecosystem where students will be able to learn and share skills while learning soft skills is the goal of SkillShare. Synchronous features like video conferencing, schedule planners and a skill matching algorithm are implemented within the platform to help students connect with each other and encourage reciprocal learning.

Literature Review

SkillShare has substantial uses in the field of higher education on par with traditional lecture-based outlets. This literature review will highlight the core factors crucial enough to identify gaps and issues in the current system.

Peer-to-peer Learning and Engagement

Peer-to-peer learning motivates students to teach and learn simultaneously as individuals or groups. Research shows that identification and bridging of syllabus gaps was possible through the student-led informal "Kuppi classes" in Sri Lanka (Kommalage). Peer Assisted Learning (PAL) has also been helpful for students in improving theoretical knowledge and practical skills due to the reinforcement of repeated learning through tutoring (Theneli) (Li25).

PAL has also been resourceful in enhancing confidence among students, as reported by 60% of students who had resolved academic doubts within peer groups instead of instructors. When the formality between students and tutors are lifted, better results can be obtained (Zhang2022). It also reduces anxiety among students, promotes inclusion and retention and is helpful in building communication skills (Donald2023) (Parmer25).

All studies referenced above acknowledge the importance of PAL for both learners and tutors by yielding satisfaction and proper training in knowledge and skills (Feng2024).

Student Engagement and Collaborative Learning

Engagement is crucial for students to gain academic success. Collaborative learning helps in reinforcing engagement by increasing participation, problem solving and accountability (Donald23).

Collaboration is also significant to learn information quicker based on the learning context. Informal and peer-driven groups are proven to increase motivation among students while simultaneously reducing tutor dependency (Parmar25).

SkillShare offers contact-driven learning capabilities to its users by integrating several synchronous features within itself.

Skill-Sharing Platforms and Knowledge Exchange

Similar apps like SkillShare are in dire need to enable real-time collaboration and feedback-based engagement to not deter users away from the platform. People will only use a platform like this if safety was guaranteed and not exploited on any bias and unstructured learning. A study has validated a skill-sharing platform that allows its users to share and retain information through blogs, webpages and collaboration (Obionwu).

Theoretical Framework Supporting SkillShare

A theoretical framework formulated via research and academia supports SkillShare because it has been validated through various studies. Communities of Practice (Lave and Wenger) is a theoretical framework that refers to a group passionate towards a common goal that they practice and learn through regular interactions. Domain, community and practice are the three main components under this framework. SkillShare is heavily influenced by this framework in its overall design (Leanringtheories).

Limitations of Formal and MOOC Teaching Models

Formal education and MOOCs are often considered by all students for learning, but they are highly disadvantaged by their rigidity and affordability. Multiple studies have provided evidence that MOOCs have low completion rates and challenges users to engage with them on a regular basis with no extra factors like feedback and structured resources (Feng).

Formal curricula also do not align directly with employers' demands or the market volatility which hinders professional growth to an extent (WBG2016).

Proposed Solution

SkillShare is an open-source, web-based skill exchange platform that terminates both financial and accessibility barriers to students across all demographics in Sri Lanka. Regardless of the location, students will be able to share their skills in exchange for expertise in another skill they want to learn. This helps SkillShare to create a structured, community driven ecosystem to foster collaboration and accessibility among students.

The core features of this system are,

1. Skill-Trading Engine
    - An algorithm to match users based on skills being offered and requested, availability and additional structured resources available.
    - Minimal bias in matching users is present to fairly promote collaboration among students.
2. Video Conferencing
    - Jitsi Meet, an open-source application is being used to integrate video conferencing within SkillShare.
    - This is a web app which can support both one-on-one and group sessions for learning.
3. Scheduling Planner
    - A calendar is integrated within SkillShare to schedule sessions easily.
    - Helps optimize time management and coordination issues among peers.
4. Learning Pathways
    - Structured learning pathways is present to help both tutors and students learn.
    - Proper resources and roadmaps will be linked to the platform.
5. Location Assistance
    - Maps are integrated within profile dashboard to provide simplified navigation to all users.
6. Peer Reputation System
    - Ratings and reviews are integrated within profile dashboard to provide context on reliability and other qualities of study etc.
    - Integration of this feature was crucial to build trust among users.
7. Secure Backend Infrastructure
    - A secure backend was developed to manage and secure sensitive data obtained from users.
    - MongoDB Atlas is used to guarantee robust performance.

The optional features of this platform are,

1. Skill Verification System

- This feature was developed as an optimal feature to help users verify tutors' proficiency in skills and increase credibility.
- This is integrated within the profile dashboard.

Two optional features provided in the Interim report, gamification module and community forum module couldn't be developed due to time constraints and development issues.

Project Aim

The aim of SkillShare is to provide a reliable, user-friendly platform for students to exchange skills for one another. This system was created to nurture mutual support among peers while attaining informal education. Furthermore, this platform is intended for use by all informal learners, regardless of their socioeconomic background.

Project Objectives

The objectives of SkillShare are given below,

1) Build a Skill-Trading Engine
    - Develop a module to match users based on pre-defined criteria while also supporting scaling across multiple simultaneous users without any performance disruptions.
2) Implement Collaboration tools
    - Integrate Jitsi Meet for a browser-based video conference experience by ensuring end-to-end encryption.
3) Design a scheduling suite
    - Design a scheduling suite with calendar and booking management dashboard. Time slots will be used to help users book multiple time slots and check user availability.
4) Enable Location assistance
    - Enable location assistance to help users navigate meetups and location-based skill-matching safely.
5) Enable a trust framework
    - Implement reviews based on personal experiences to moderate and flag uses, both appropriate and inappropriate, of the platform.
6) Support various learning models
    - Facilitate learning between individuals and groups while providing features like multilingual support in specific areas.
7) Ensure Security Compliance and Data Privacy
    - Adhere to GDPR standards while handling sensitive user data and use of proper encryption techniques to increase security.
8) Deliver Comprehensive Documentation
    - Produce a user guide and tutorial for users.

9) Integrate resources and learning pathways for multiple categories.
    - Integrate learning pathways like roadmaps and resources from the web by providing links.

The objectives related to gamification module and open-source contribution for SkillShare have also been scrapped due to time constraints and a lack of proper resources for development.

Functionalities and Non-Functionalities

Functionalities

1) Identification and Accessibility
    - Users can register with email and password.
    - Email and password are verified and secured.
    - Login as User.
2) Profiles
    - Profiles can be created by users to display their information
    - Users can add skills they offer and would want to learn.
3) Matching Users
    - Users are matched using pre-defined criteria like skills, location and availability etc.
4) Scheduling
    - Learners can create time slots for their learning sessions.
    - The system creates the bookings and shows it to users.
5) Conferencing
    - Video conferencing features are available.
6) Resources
    - Brief and elaborate resources.
7) Moderation and Trust
    - Users can report misconduct and fake skill proficiency through reviews.
8) Ratings and Feedback
    - Post session feedback that goes both ways.
9) Optional Enhancements
    - Skill Verification: Provide skill proficiency evidence through endorsements and evidence.

Non-Functionalities

1) Security
    - Limited Login rate.
    - Server-side checks during access control
    - Privacy control for consent related activities in the platform.

- TLS 1.2+ for transport and storage of data.

2) Privacy and Data Protection
    - Store only what is necessary through minimization.
    - Retain necessary information.
3) Reliability
    - Target more than 95% monthly uptime for core APIs and authentication at minimum.
    - If video conferencing fails, provide proper suggestions as placeholder.
4) Scalability
    - Scale horizontally during peak seasons using auto scalable hard-coded rules.
5) Maintainability
    - Modular architecture within the system to resolve issues individually after identification.
6) Ethical Constraints
    - Financial transactions are not applicable without a proper code of conduct.
7) Compliance
    - Terms of policy and Privacy policy being available on the platform.
8) Usability
    - Desktop-first responsive UI and stable browser version to be used on chromium browsers.
9) Performance
    - All matches should be done within 5-10 seconds at worst and 2-5 seconds at best.
    - Video conferencing should have adaptive bitrate.
    - API latency should be under 50% under 300ms (milliseconds) and 95% under 1500ms (milliseconds).

Software Development Methodology

Models Considered for SkillShare

Waterfall model and V-model were considered for the SkillShare development but were then rejected due to the following reasons,

I. Waterfall model doesn't allow iterative approach during development as it has a sequential structure. A phase must be completed before moving on to the next which is a burden when experimenting with changes and functionalities based on constant demands. Hence, this model is not suitable to build SkillShare which requires constant iterative approach (GeeksforGeeks25).
II. V-model requires that testing should be done once the project is entirely or almost over and doesn't allow to constantly iterate and reiterate based on feedback which impacts the development of features. Furthermore, it requires a set of instructions to be completed before development can begin or conclude (Geeksforgeeks2025).

SDLC Model Selection and Justification

Agile methodology is the most suitable SDLC model for SkillShare development due to the iterative nature it offers. It also uses an incremental approach to prototype and test features rapidly (GeeksforGeeks2025).

The team can develop and deliver features within mere sprints in comparison to other models through constant feedback from stakeholders. This is very useful when developing a community-driven application in a real-world scenario to ensure the product doesn't fail (GeeksforgeekS2025).

Advantages of Agile Methodology

The pros of agile methodology are as follows (gEeksforGeeks25),

- Flexible and iterative development allowing feature refinement based on feedback.
- Rapid prototyping allows users to test and fix bugs, promoting user-centered practices.
- Development time is quicker in comparison to other models through frequent sprints.
- Very collaborative friendly for stakeholders in suggesting changes.

Chapter 02

Feasibility Study Report

This report analyses the feasibility of SkillShare as a platform to exchange skills.

Technical Feasibility

- Team and Skills: Defined roles for teammates, agile sprints enable faster development.
- Tech Stack: Tech Stack is apt for the features.
- Infrastructure: Frontend deployed in Vercel, Linux VPS for APIs and Docker.
- Security and Privacy: Encryption and GDPR compliant data access and handling
- Scalability: Managed authentication, horizontal scaling and adaptive video quality.

Operational Feasibility

- Operational risks: Low adoption, resolved through marketing and referrals etc.
- Users and workflows: Clear UX with workflow, supported through moderation and trust.
- Resources Availability: Documented team structure and Hardware requirements being fulfilled.
- Delivery Plan: 90-day schedule from September to November with multiple development phases.
- Compliance: Privacy and data protection practices being aligned with security measures.

Financial Feasibility

- Market Need: SkillShare targets the demand to provide education through an interactive learning platform.
- Value: Has multiple features to help users interact and learn with peers.
- Cost Estimate: LKR 200,000, development requires LKR 145,000; domain requires LKR 3000 and marketing requires LKR 52,000.
- Cost Control: Using open-source components throughout development.

Requirement Gathering Process

The requirement gathering process was targeted towards students. All information collected through google forms were anonymized.

Identification of Stakeholders

Students are the direct stakeholders of SkillShare.

Users (Students): Users facing difficulty in finding cost-free informal learning opportunities were surveyed through google forms.

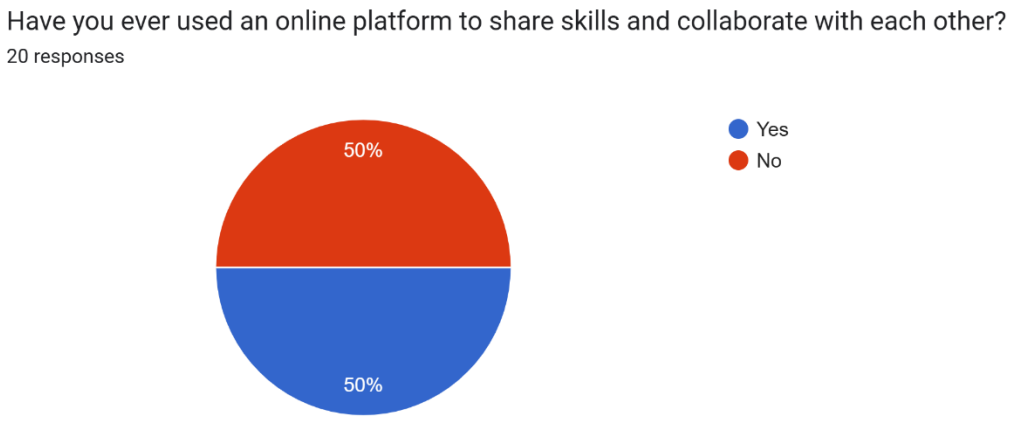
Data Collection Methods

Survey was used to collect data for this project. 20 questions were given, in which 16 of them were MCQs and the rest were open-ended questions. The questions targeted the information required to start development and planning phases accordingly.

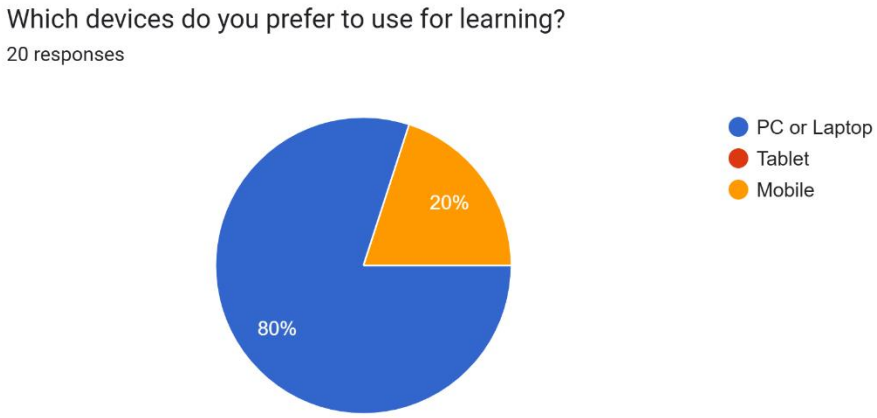
Interviews were also conducted within team members but were not formally recorded as the data was collected verbally.

Survey Questionnaire with Answers: Findings and Outcome

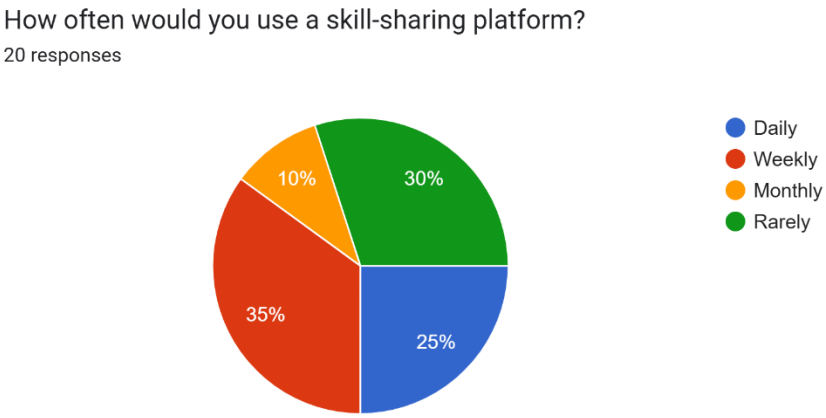
Multiple Choice Questions



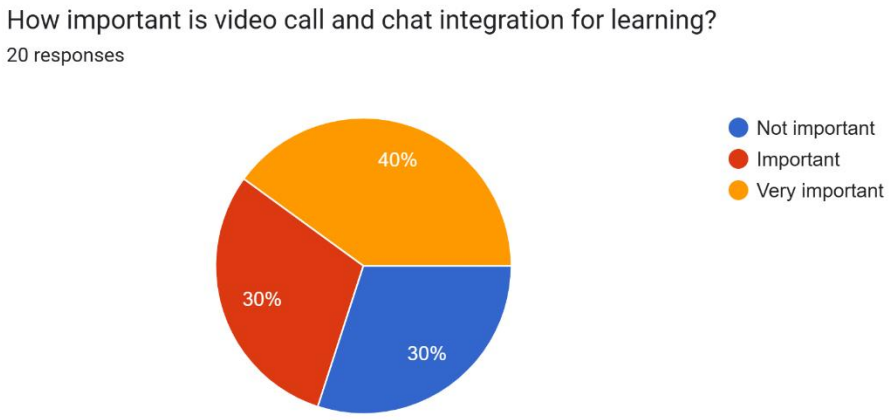
50% of the answers have answered “Yes” while the rest chose “No” which validated that SkillShare is being used by some already.



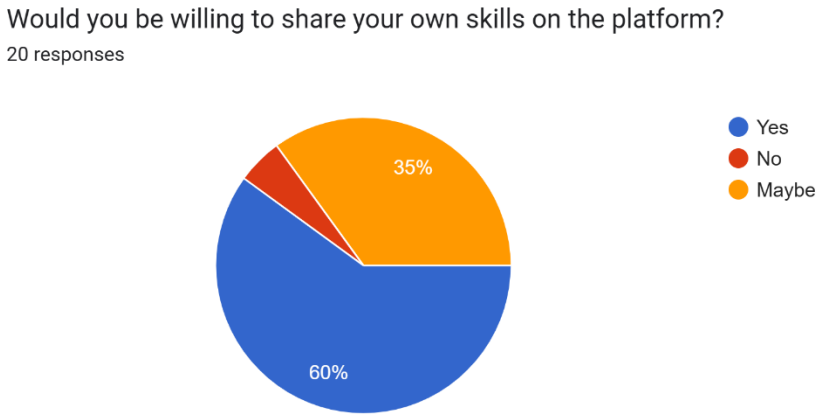
80% of the answers opted for a PC or a laptop as their primary learning device. Due to this, SkillShare is developed as a desktop-first application.



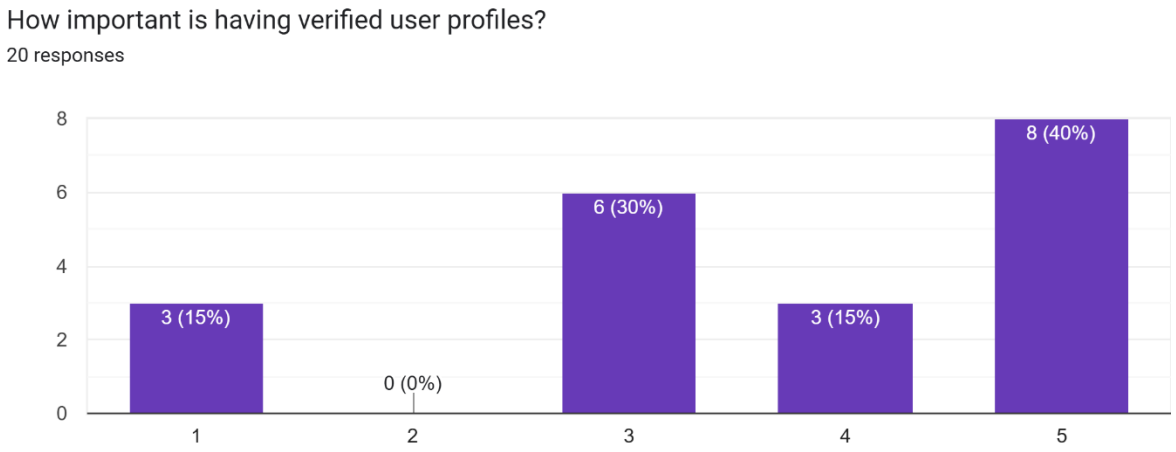
Based on this, SkillShare is rarely used by the majority as an option but 70% collectively said they might use the application either daily, weekly or monthly.



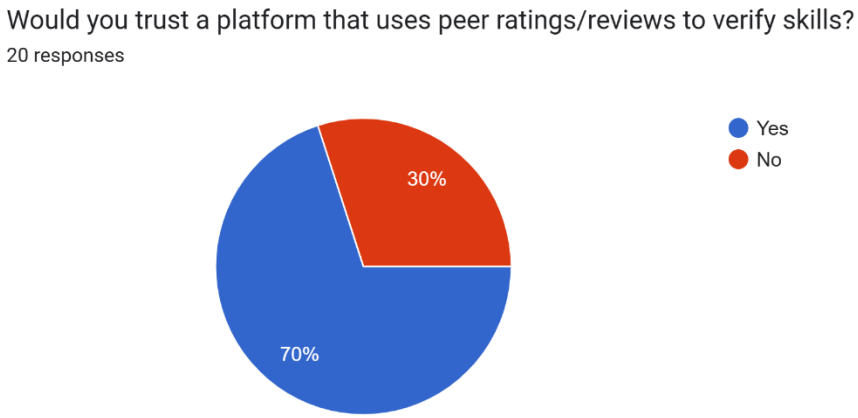
30% have dictated that this feature is not required while the rest have said this is either important or very important.



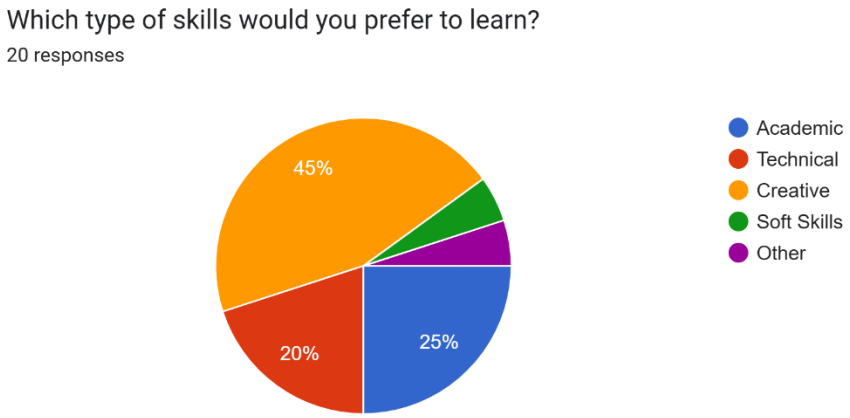
Majority have chosen “Yes” while 35% had chosen “maybe” and the rest choosing “No”.



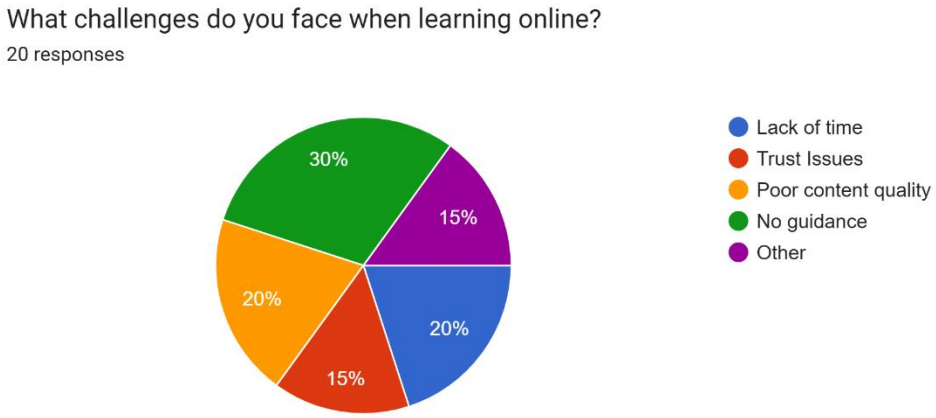
The response was varied to this question with the majority choosing a moderate option of it being required while only 15% had chosen the adverse.



Majority have chosen “Yes” indicting that ratings and reviews option will increadse trustin the platform.

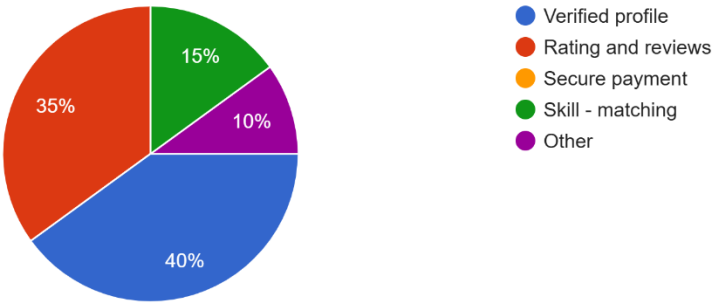


Most people have chosen to learn creative skills while less people have chosen academic and other skills.



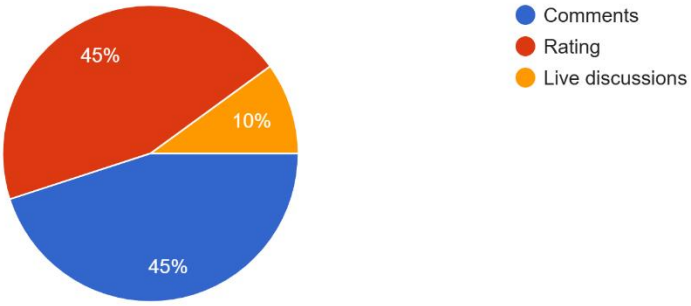
“No guidance” and “Lack of guidance” are the most chosen options which validated the need for a scheduling planner and learning pathways.

Which features would make you trust and regularly use our platform?
20 responses

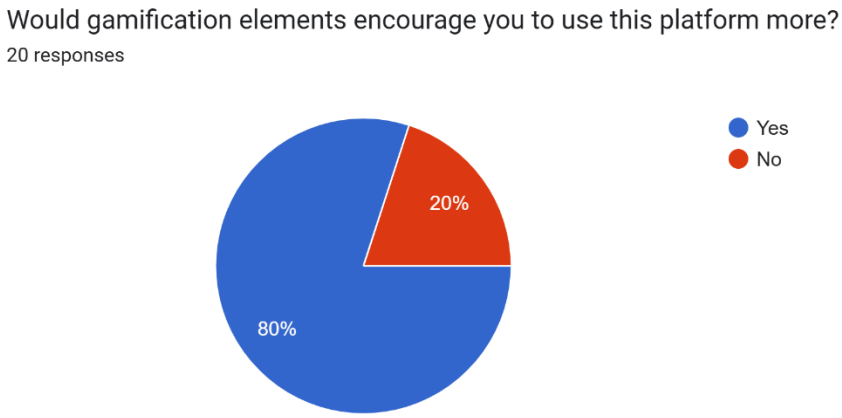


This MCQ furthermore provides assurance for us, the development team, to input verified profiles within SkillShare to guarantee the safety our users deserve.

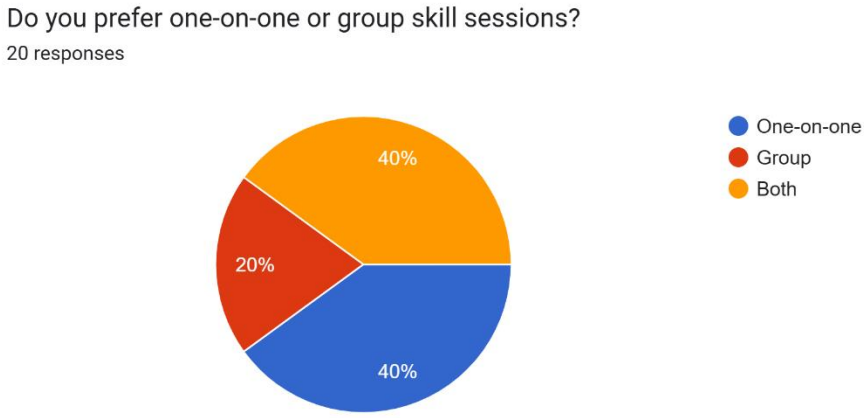
How do you prefer to receive feedback on your skills?
20 responses



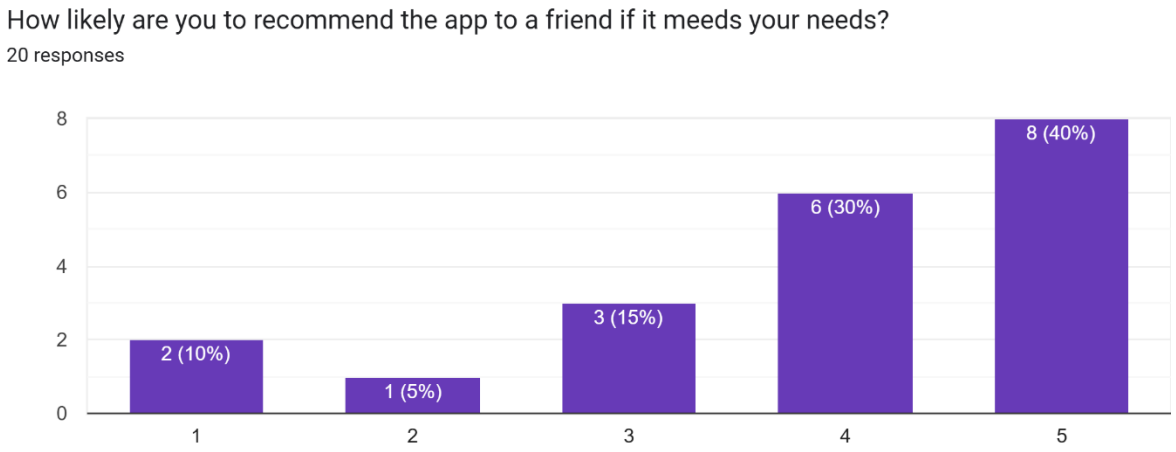
Comments and ratings have been chosen by the majority leaving the live discussions option as non-viable during the initial phases.



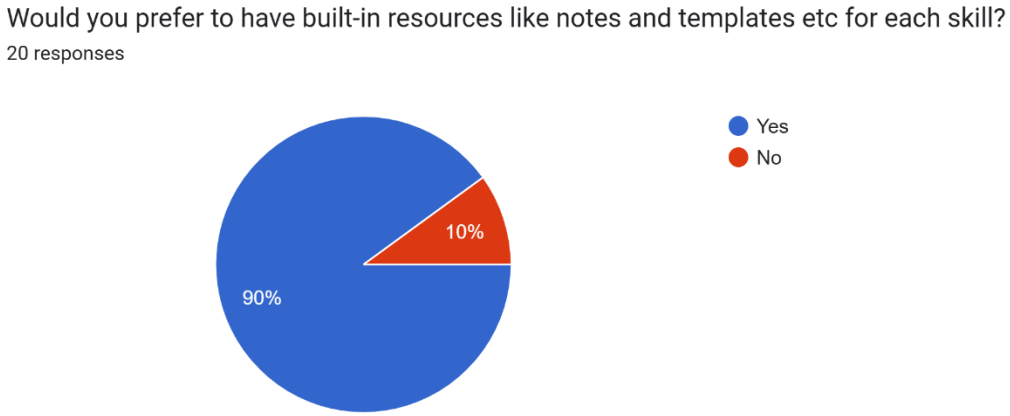
Gamification elements were given priority by the respondents.



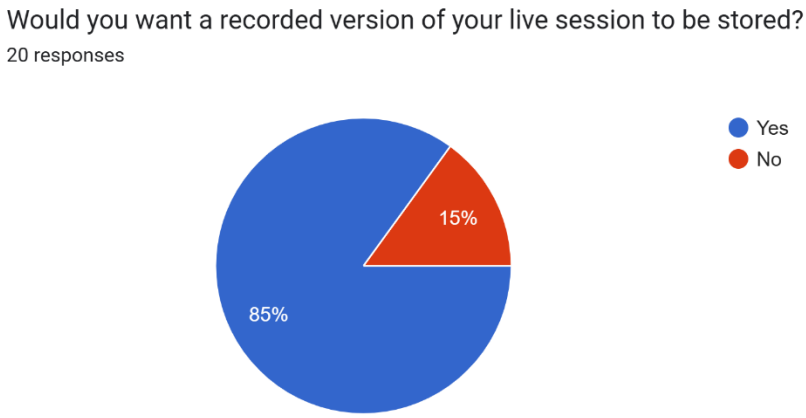
People are quite okay with any type of sessions to meet online hence this will have absoutely no impact on our plans from earlier on.



Majority of the respondents have chosen the option to recommend this to their friends if it met their needs and expectations



Proper resources' links will be provided through the platform if possible.



Video recording feature is optional but the possibility of it being bundled with Jitsi Meet is being considered.

Open-Ended Questions

Totally 04 questions were asked in the paragraph style open questions to which there have been 80 responses so far collectively. But three answers cannot be considered valid due to it being ultra short with no substantial meaning or irrelevant phrases or words to the context of the question. Hence only 77 out of 80 responses can be considered valid for this survey which marks the data quality rate at approximately 96.25% while the rest 3.75% is invalid. Let's analyze the questions individually.

1. What Concerns do you have about used online skill-sharing platforms?
   The answers provided were as follows,
   - Nothing
   - The untrust people of this generation, they make me want to puke on, one peak at their mind and all the disgusting thoughts turns into puke
   - Quality content
   - Worthy enough
   - trust, quality, and consistency are usually the biggest concerns
   - Hi
   - Features being locked behind a pay wall
   - Trust issues
   - Difficulty finding the right skills and limited support
   - I don't know
   - Proper guidance not being available
   - Privacy and data security
   - Privacy & security
   - Let's see
   - I think its security and privacy
   - Not brief
   - Shyness
   - About the content quality
   - My data being accessed by third parties

When this question is analysed, we can understand that these responses validate the core requirement priorities, which are privacy and trust, and then matching and finally usability

and onboarding. We have successfully iterated and made changes to the initial development plan by including these changes in it too.

2. If you were to teach a skill, what would it be and why?
    - Stop racism
    - Web designing because I love to
    - Trading
    - I’d probably teach it through practical scenarios (case studies, puzzles, real-life challenges) rather than just theory, so learners see immediate benefits.
    - Bye
    - Academic writing
    - Problem solving
    - Communication skill
    - Web Development to help learners gain practical and in demand skills
    - Poetry as I know that
    - Communication skills because they are useful
    - Writing and because I'm proficient in it
    - If I were to teach a skill, I would teach effective communication.
    - Time management
    - Because time is the one thing we can’t earn back. If people learn how to manage it wisely, they can balance work, rest, and dreams turning chaos into clarity and stress into success.
    - Art as I know
    - Coding
    - Technical
    - Kungfu
    - Web designing
    - Coding because I’m relatively better at it

During the analysis of the answers, we find that users require various kinds of skills to learn for which we have given them the option to choose a few skills during signup, users can also choose their teaching styles owing to what they feel comfortable with and again filtering is an option required by the users to use this platform and find their required skill.

3. How can the platform make skill-matching more effective to you?
    - By implementing more features like teleportation
    - Algorithms
    - Not applicable
    - Matching works best when it feels personal and adaptive not just “here’s a teacher,” but here’s the right teacher or peer for you, at this stage, with your goals.
    - Hello
    - By matching me with people who would want my skills to be shared
    - By matching my skills with the right opportunities.
    - Being attach to everyone preference
    - Filter & Al recommendations based on interests, goals and skill level
    - I don’t know
    - By giving personalized suggestions to people
    - By matching my interests
    - The platform can improve skill-matching through personalized suggestions, level checks, and filters.
    - Trial sessions and AI recommendation
    - Umm Umm
    - A trustworthy skill matching system
    - Reasonable
    - Can learn more from matching people
    - Clean profile
    - By matching me with people near my location for physical meetups

Respondents wanted the skill-trading engine to be made more effective through personalization, adapting to user needs and through enabling trust, practical features and reciprocal opportunities while skill sharing. Users want a proper algorithm to match fairly with other peers hence we need a proper recommendation-based algorithm to work with SkillShare.

4. Suggest one feature you think would make this platform stand out
    - The video call and free to use app makes it stand out compared to others
    - Gamification
    - Simple interface
    - Combine it with AI suggestions the platform could recommend projects suited to a user's current skill level, helping them grow faster and more confidently
    - Good boy
    - A smooth gamified platform
    - A personalized skill-matching dashboard
    - Make it available for everyone without any premiums
    - Rewards & e-certification system to motivate learners & recognize their progress
    - 1
    - Live sessions with experts
    - Easy to use interface
    - Personalized learning paths.
    - Skill journey tracker.
    - Something we can track everything and see if we have to learn more or make some difference in learning.
    - Ahhhhh
    - Skill Matching API
    - Reaching out
    - Chatting
    - Customisable profile
    - Event panel where people can meet and discuss topics

Standout features suggested by the users include gamification, personalisation, accessibility for all users, community features like live sessions and event panels, progress tracking via dashboards to know what level of mastery a user has based on the skill trackers and finally the ability to customize profiles while also keeping the UI/UX as simple as possible.

Resource Identification

Resource Management

Sprints with weekly planning rituals, daily stand ups and sprint reviews were utilized. And a simple RACI matrix helped ensure accountability:

- Project manager is accountable for scope and delivery.
- Developer is responsible for data/API contracts.
- QA checks the releases.
- Engineer fixes the matching engine.

Quality assurance is guaranteed through unit tests, API integration tests, and session feature tests along with accessibility checks. Furthermore, all risks were managed and neutralised using frequent checks, tests and moderation workflows ensuring operation success.

Resource Allocation

SkillShare will be built by a small functional team which includes,

- Project Manager
- UI/UX designer
- Frontend Engineers
- Backend Engineers
- DevOps Engineer
- QA lead

Workflow is already underway in 8 one-week sprints.

Fundamentals → profiles/skills → matching → scheduling → sessions → messaging → safety → hardening process.

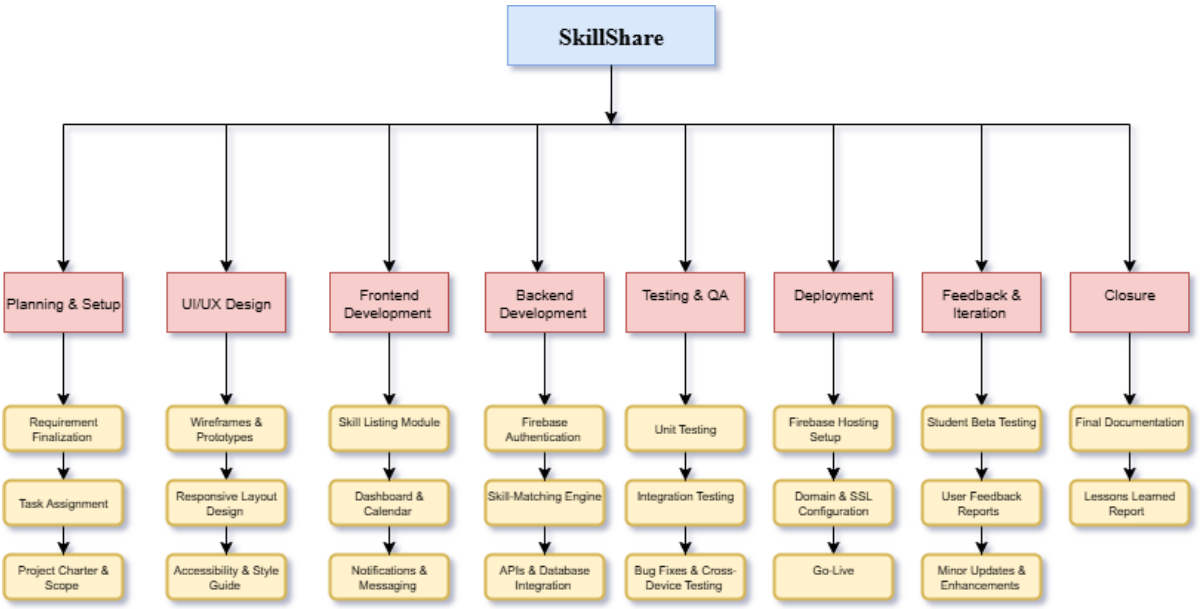
Environments are using development, staged and production with managed Postgres, object storage for media, and Redis for caching/queues. Total third-party costs are minimized (ideally free/student tiers) with email, limited bandwidth and hosting as the main budget; 10–15% sprint capacity is allocated for refactors/bugs.

Category	Item	Description	Cost (LKR)
Development	Front-end Development	UI components (React.js, HTML5, CSS3, GSAP animations)	60 000
	Back-end Development	APIs, business logic (Node.js/Express)	50 000
	UI/UX Design	Prototypes, user testing, style guide creation	20 000
	Testing & QA	Manual and automated test cases, cross-device/browser tests	15 000
Tools & Licenses	GitHub	Version control & open-source hosting	0
	Firebase Free Tier	Authentication, Realtime DB, Cloud Functions	0
	Netlify / Vercel	Free hosting for static assets & serverless APIs	0
	Figma (Free Plan)	UI/UX design & prototyping	0
	Google Meet / Zoom	Team meetings & stand-ups	0
Infrastructure & Hosting	Domain (.lk)	One-year registration	3 000
	SSL Certificate	Let’s Encrypt HTTPS	0
Contingency & Miscellaneous	Marketing (Organic)	Campus flyers, social media materials	10 000
	Contingency Buffer	Unplanned expenses	42 000

Grand Total	—	**All development, infrastructure & misc. costs**	**200 000**

Work Breakdown Structure

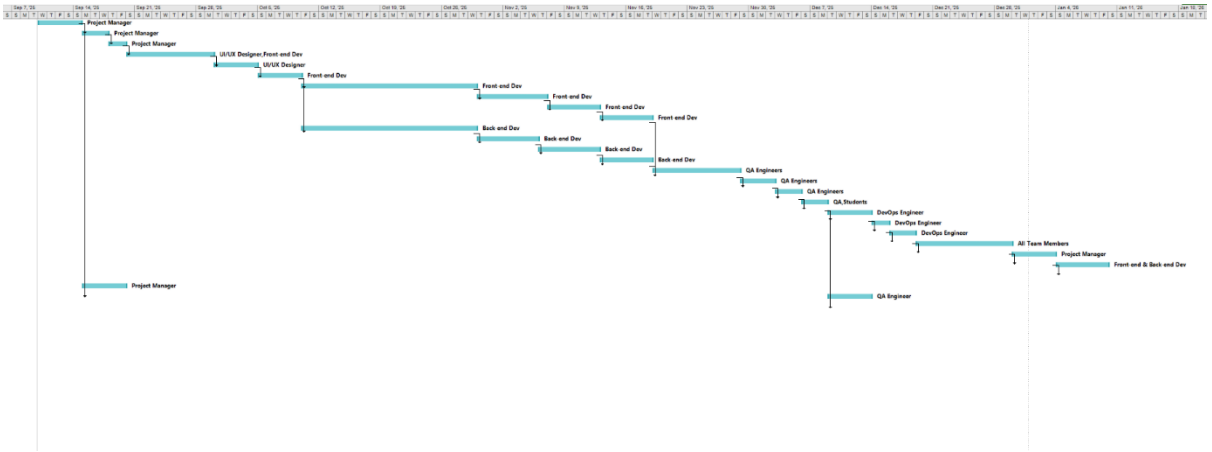
WBS Graphical Format



WBS Tabular Format

ID	Task Name	Duration	Start	Finish	Predecessors	Resource Names
1	Planning & Setup	5 days	10-Sep-25	14-Sep-25	-	Project Manager
2	Requirement Finalization	3 days	10-Sep-25	12-Sep-25	1	Project Manager
3	Task Assignment	2 days	13-Sep-25	14-Sep-25	2	Project Manager
4	UI/UX Design	10 days	15-Sep-25	24-Sep-25	3	UI/UX Designer, Front-end Dev
5	Wireframes & Prototypes	5 days	15-Sep-25	19-Sep-25	4	UI/UX Designer
6	Final Responsive UI	5 days	20-Sep-25	24-Sep-25	5	Front-end Dev
7	Frontend Development	20 days	25-Sep-25	14-Oct-25	6	Front-end Dev
8	Skill Listing Module	8 days	25-Sep-25	02-Oct-25	7	Front-end Dev
9	Dashboard Implementation	6 days	03-Oct-25	08-Oct-25	8	Front-end Dev
10	Calendar Integration	6 days	09-Oct-25	14-Oct-25	9	Front-end Dev
11	Backend Development	20 days	25-Sep-25	14-Oct-25	6	Back-end Dev
12	Firebase Authentication	7 days	25-Sep-25	01-Oct-25	11	Back-end Dev
13	API Development	7 days	02-Oct-25	08-Oct-25	12	Back-end Dev
14	Database Setup (MongoDB/MySQL)	6 days	09-Oct-25	14-Oct-25	13	Back-end Dev
15	Testing & QA	10 days	15-Oct-25	24-Oct-25	10,14	QA Engineers
16	Unit Testing	4 days	15-Oct-25	18-Oct-25	15	QA Engineers
17	Integration Testing	3 days	19-Oct-25	21-Oct-25	16	QA Engineers
18	Usability Testing (Pilot)	3 days	22-Oct-25	24-Oct-25	17	QA, Students
19	Deployment	5 days	25-Oct-25	29-Oct-25	18	DevOps Engineer
20	Firebase Hosting Setup	2 days	25-Oct-25	26-Oct-25	19	DevOps Engineer
21	Domain & SSL Configuration	3 days	27-Oct-25	29-Oct-25	20	DevOps Engineer
22	Feedback & Iteration	10 days	30-Oct-25	09-Nov-25	21	All Team Members
23	Collect User Feedback	5 days	30-Oct-25	03-Nov-25	22	Project Manager
24	Minor Updates & Improvements	5 days	04-Nov-25	09-Nov-25	23	Front-end & Back-end Dev
25	Documentation & Reports	Ongoing	Sep–Nov	Continuous	-	Project Manager, QA
26	SRS & Architecture Docs	5 days	10-Sep-25	14-Sep-25	1	Project Manager
27	User Guide	5 days	22-Oct-25	26-Oct-25	18	QA Engineer
28	Sprint & Testing Reports	Ongoing	Sep–Nov	Continuous	-	QA Engineer

Gantt Chart


Project Manager
Project Manager
Project Manager
UI/UX Designer,Front-end Dev
UI/UX Designer
Front-end Dev
Front-end Dev
Front-end Dev
Front-end Dev
Front-end Dev
Back-end Dev
Back-end Dev
Back-end Dev
Back-end Dev
QA Engineers
QA Engineers
QA Engineers
QA,Students
DevOps Engineer
DevOps Engineer
DevOps Engineer
All Team Members
Project Manager
Front-end & Back-end Dev
Project Manager
QA Engineer

Risk Analysis

Technical Risks

Risk	Mitigation	Contingency Plan	Context
Scalability issues	Use cloud infrastructure	Scale servers, optimize code	Critical for growth, demand can strain resources.
API failures	Backup APIs, test regularly	Switch to alternative APIs	APIs are external dependencies; redundancy is essential.
Server downtime	Cloud hosting	Failover servers, monitoring	Downtime hurts user trust; resilience measures must be in place.

Financial Risks

Risk	Mitigation	Contingency Plan	Context
Budget overruns	Cost tracking, prioritize MVP	Delay features, seek funding	Cost discipline ensures sustainability during early stages.
Funding shortages	Use free/open-source tools	Apply for grants, crowdfunding	Alternative funding helps avoid dependency on investors.

User Risks

Risk	Mitigation	Contingency Plan	Context
Low adoption	Campus marketing, referrals	Expand marketing, outreach	Adoption drives growth.
High churn	Gamification, engagement tools	Collect feedback, improve UX	Retention is as important as acquisition for long-term success.

Operational Risks

Risk	Mitigation	Contingency Plan	Context
Skill gaps	Training, agile sprints	Hire consultants/interns	Knowledge gaps can slow development.
Project delays	Plan milestones, monitor	Adjust scope or timelines	Delays are common; agile adjustments help stay on track.

Security Risks

Risk	Mitigation	Contingency Plan	Context
Data breaches	Encryption, SSL, auth	Incident response plan	Sensitive data protection is critical for compliance and trust.
Unauthorized access	Role-based access, MFA	Enforce stronger policies	Internal misuse is a high-risk area; strict controls are vital.

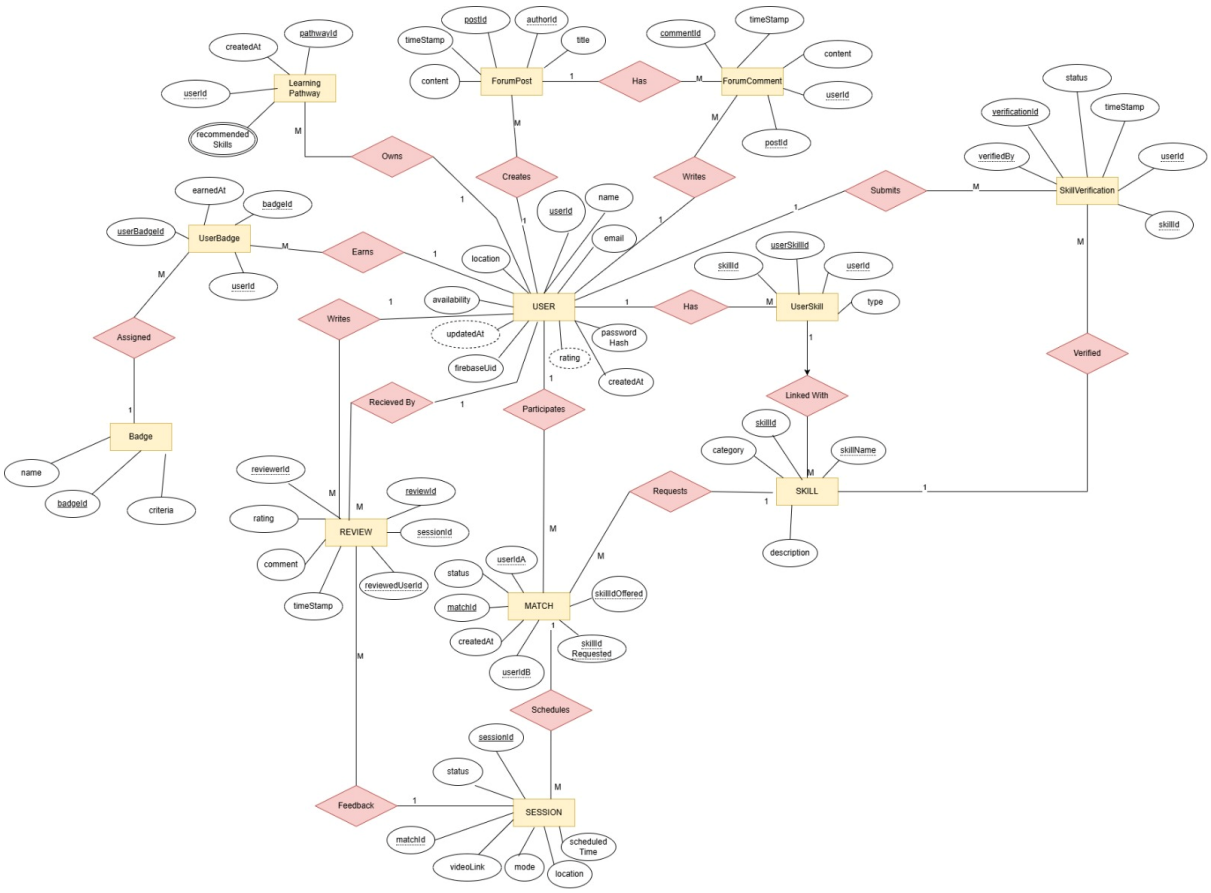
Regulatory Risks

Risk	Mitigation	Contingency Plan	Context
Compliance issues	Legal review, GDPR compliance	Update policies, consult legal	Global operations require regulatory adaptability.
Data privacy	Clear policies, user consent	Remedial action, notify users	Transparency and user trust are central to platform credibility.

Chapter 03

Design of the System

Entity Relationship Diagram



Normalized Relational Schema

User (user_id, name, email, password_hash, location, availability, createdAt, updatedAt)

Skill (skill_id, skillName, category, description)

UserSkill (userSkill_id, user_id, skill_id, type, createdAt)

SkillVerification (verification_id, userSkill_id, user_id, skill_id, verifiedBy, status, timeStamp)

Match (match_id, user_idA, user_idB, skillOffered_id, skillRequested_id, status, createdAt)

Review (review_id, session_id, reviewerUser_id, reviewedUser_id, rating, comment, timeStamp)

LearningPathway (pathway_id, user_id, createdAt)

Database

MongoDB has been chosen as the database solution for SkillShare due to its scalability and the ability to handle unstructured data. Several data model collections derived from the ER diagram is used as the base for the system which are given below,

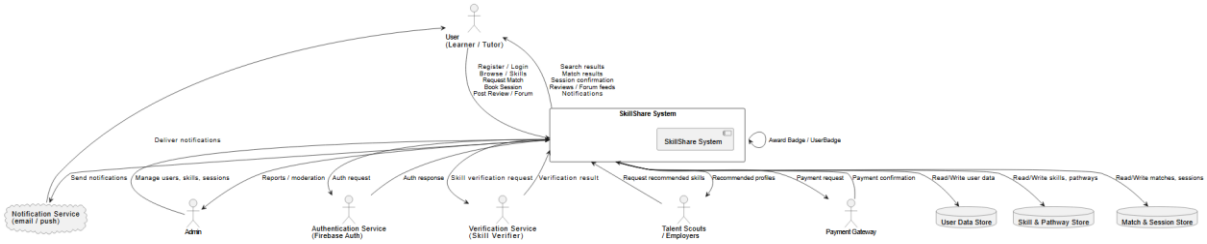
1) Users: Personal Information including login credentials, skills and learning pathways.
2) Skills: Skills along with descriptions and users who have chosen it.
3) Matches: Stores user matches data.
4) Reviews: Stores the comments from all users.
5) Learning Pathways: Stores learning pathway data chosen users.

Due to the usage of MongoDB, extensive scaling, high availability and tremendous uptime can be guaranteed.

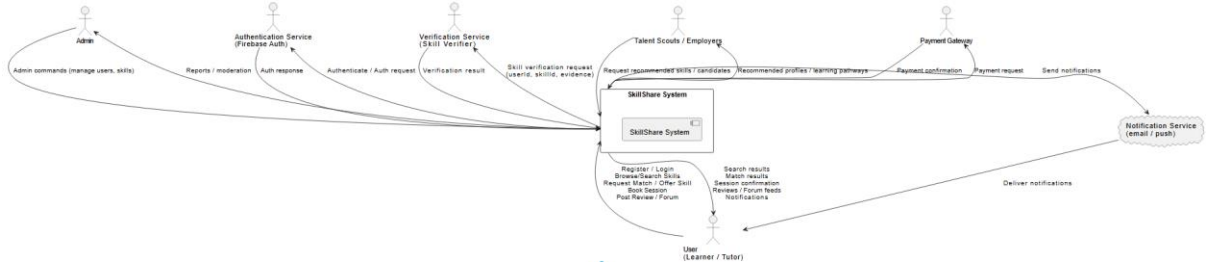
System Design

DFD Diagrams

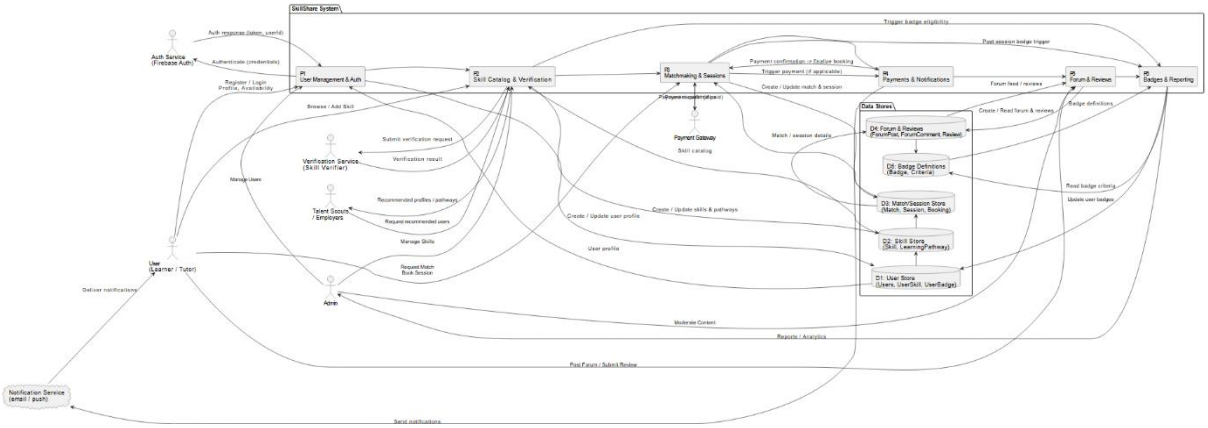
Context Diagram



Level 0 Diagram

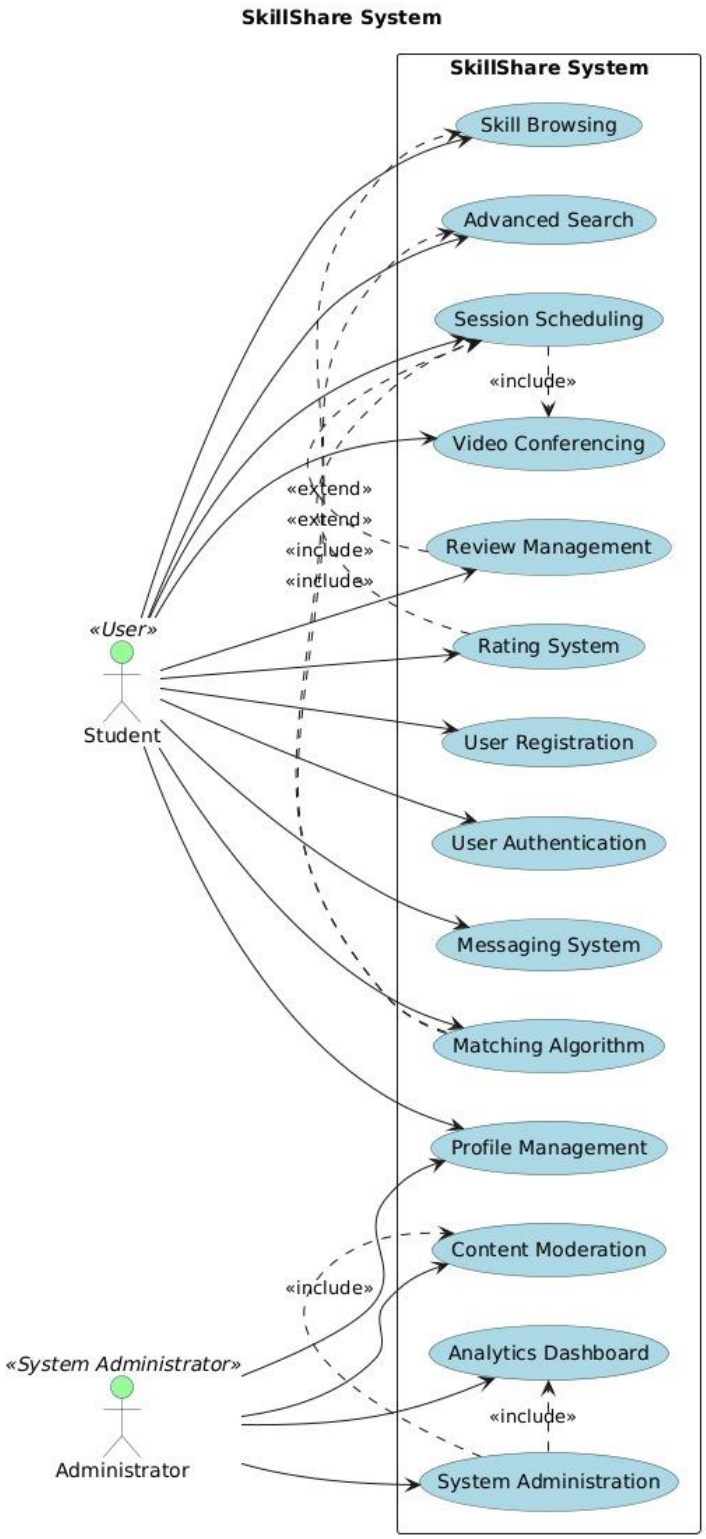


Level 1 Diagram

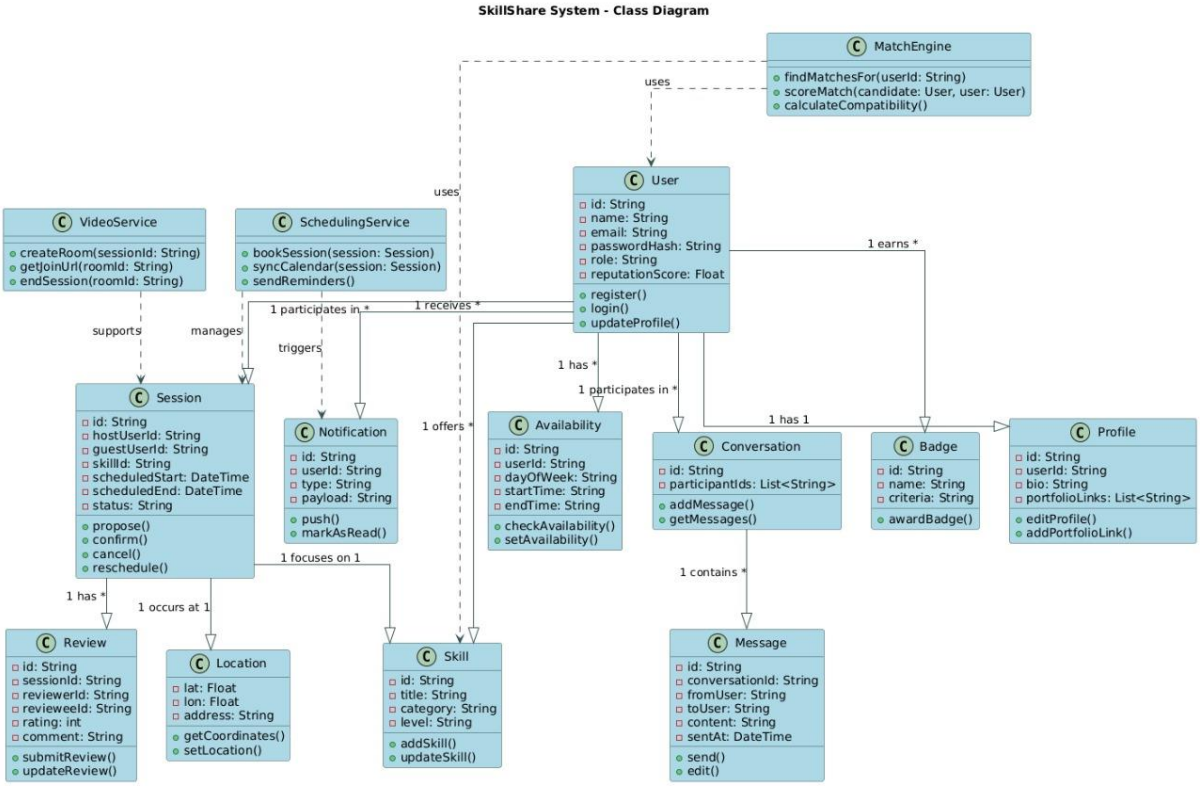


UML Diagrams

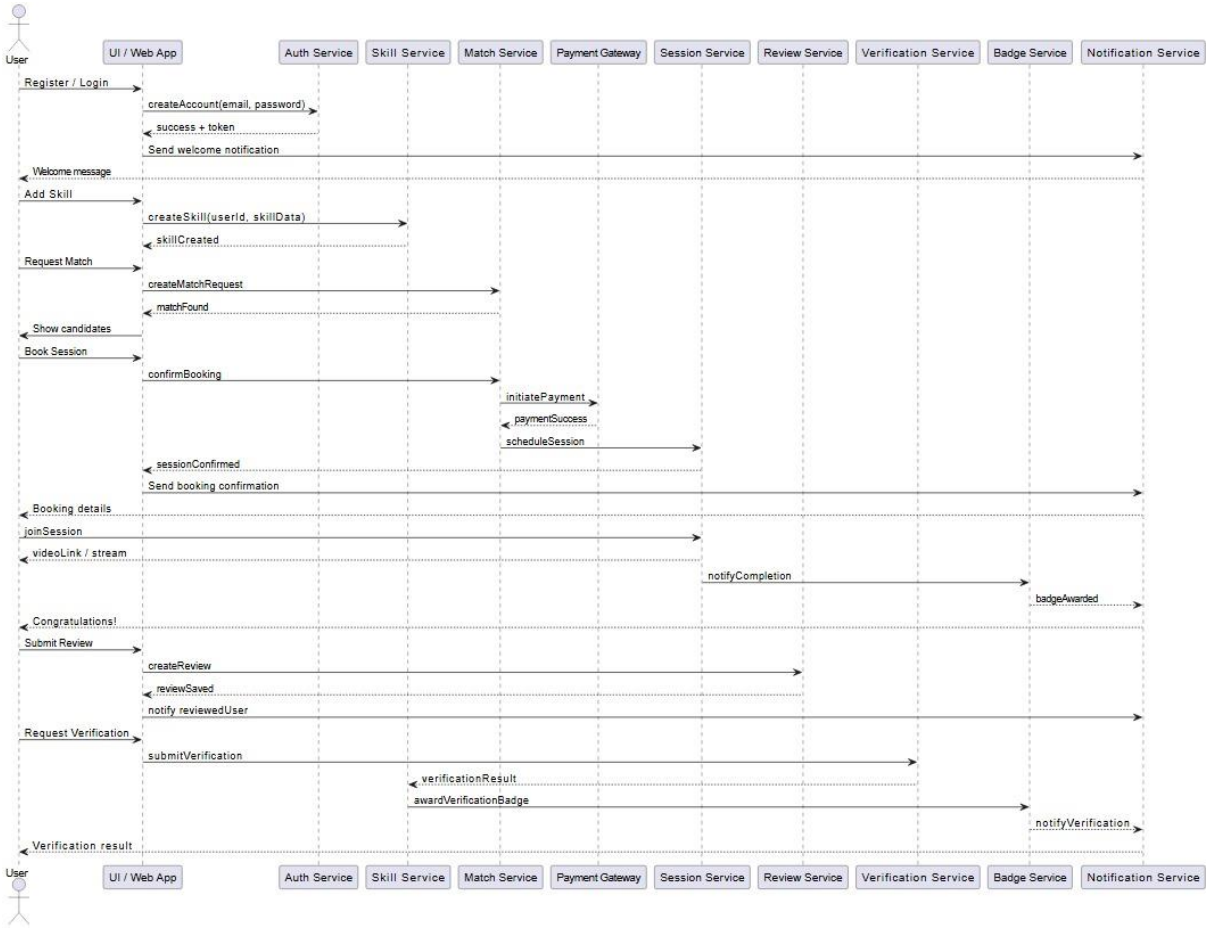
Use Case Diagram



Class Diagram



Sequence Diagram



User Interfaces/Navigations

User Interfaces

Wireframes

Navigation

SkillShare navigation is designed in a user-friendly manner. The process is described below,

- Users start at the homepage where they can register as a user. Then they will be taken into the application.
- Users will access their profile dashboard which has everything from skill matching engine, calendars and learning pathways included within.
- Admins will also register as the user but will get some additional features in the admin dashboard.

Chapter 04

Co-Functionalities with Evidence

Chapter 05

Testing

Testing Methodology and Justification

SkillShare uses agile software testing methodology throughout its development. This approach helps developers to identify, mitigate and resolve defects early on. Functional and non-functional testing were conducted manually to validate usability and performance. Furthermore, unit tests, systems tests, integration tests and user acceptance tests were done to verify real-world usability.

Agile software testing methodology was chosen to align with the development methodology in use as it helps gain ongoing validation through feedback and testing, ensuring that feature iterations are quick and reliable.

Test Plan