COMP 2606
PROJECT

15/04/2025

Skill-Swap Platform

The Null Pointers

ADANA BRYAN	816040793
DANIELLE GONZALES	816039321
MYKEL DEDIER	816042057
REYNEKA JOE	816036807

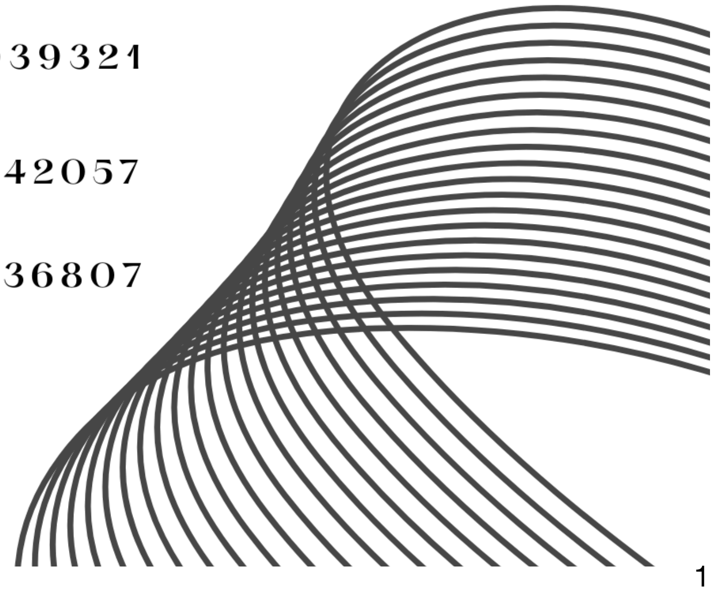


Table of Contents

Table of Figures

Glossary of Terms

Term	**Definition**
User	Someone that uses a product or service
System Admin/System Administrator	Person that manages and maintains computer systems
Moderator	Person that enforces rules on the communities that they oversee
Matchmaking	The process of connecting 2 or more users together based on the specified criteria
Two-Factor Authentication	The use of two different authentication factors for verification
Stakeholders	Person or organizations affected by decisions made.
Acceptance Testing	Acceptance testing is a quality assurance process where software is evaluated by end-users or customer representatives to ensure it meets their needs and specified requirements
Acceptance Criteria	Acceptance criteria are specific, testable conditions that a product or feature must meet to be considered complete and acceptable to users or stakeholders
Beta testing	Beta testing is a stage in software development where a select group of real users test a pre-release version of a product in a real-world environment to identify potential issues, gather feedback, and improve the product's quality before a full launch
User-Centric	User-centric means prioritizing the user's needs, preferences, and experiences throughout the design, development, and delivery of a product or service
End-Users	An end user is a person who ultimately uses or is intended to ultimately use a product.
Requirements	Requirements define the capabilities and characteristics a software system must possess to fulfill stakeholder needs and solve specific problems
Functional-Requirements	What the system should do
Non-Functional Requirements	Constraints on the system as a whole or the properties specific services and functions of the system must have/meet

Scenario	A sequence of imagined events
Use Case	Describes how a user interacts with a system or product
User Story	Short explanation of a software feature written from the perspective of the end-user

Introduction

Problem

The Skill Swap platform addresses perceived issues with the current learning system. Mainly, its barter-esque approach to learning where individuals ‘swap’ skills, is meant to break down the established barriers to learning. With it, persons without the necessary financial resources can still learn from other experts so long as they are in possession of a skill that can be traded.

Goals/Aims of System

The Skill Swap platform aims to revolutionize the way people learn. It allows persons to freely trade their expertise by teaching one another, without the involvement of any form of monetary payment.

Target User

The Skill Swap platform’s target users are individuals with a desire to learn, those who also possess an expertise or skill that can be taught to others in exchange for the acquisition of a new skill. It is not for persons seeking to gain without giving anything in return, as the purpose of the platform is to trade skills. The intention is for the platform to start out with a small closed community of users. This core user-base will be expanded to a wider audience as the system is improved.

User Stories

User Stories

1. As a user/learner, I want to create a profile/account on the app to be able to learn a skill and also teach a skill I am proficient in.
2. As a user, I want to be able to chat with the person I am skill swapping with to be able to make arrangements on how teaching and learning will take place.
3. As a system admin, I want to be able to get feedback on how the app is working to see if there are any changes that could be made to make the app run seamlessly for the users.
4. As a user,I want to use filters to search for possible skills I would like to learn so that I can easily find prospective skill swap partners.
5. As a user, I want to be able to have the option to create a profile for the first time using my gmail information so that I don’t have to manually put in my credentials.
6. As a user, I want to receive skill swap suggestions based on my listed interests so that I can easily find potential partners.
7. As a learner, I want to be able to match with skill providers in my area so that I do not have to travel long distances.
8. As a skill provider, I want to be able to display a portfolio on my profile so that others can see my proficiency in the skill I am advertising.
9. As a moderator, I want to ban and/or suspend the account of persons who falsely advertise skills (persons who lie about their skills) so that I can prevent them from circumventing the purpose of the app.
10. As a learner, I want to make comments on those I've skill swapped with and see the comments of other learners so that others can be made aware of the experience I've had with a particular skill provider and vice versa.
11. As a moderator, I want to be able to moderate the review system so I can foster a respectful online environment

Requirements

Functional Requirements

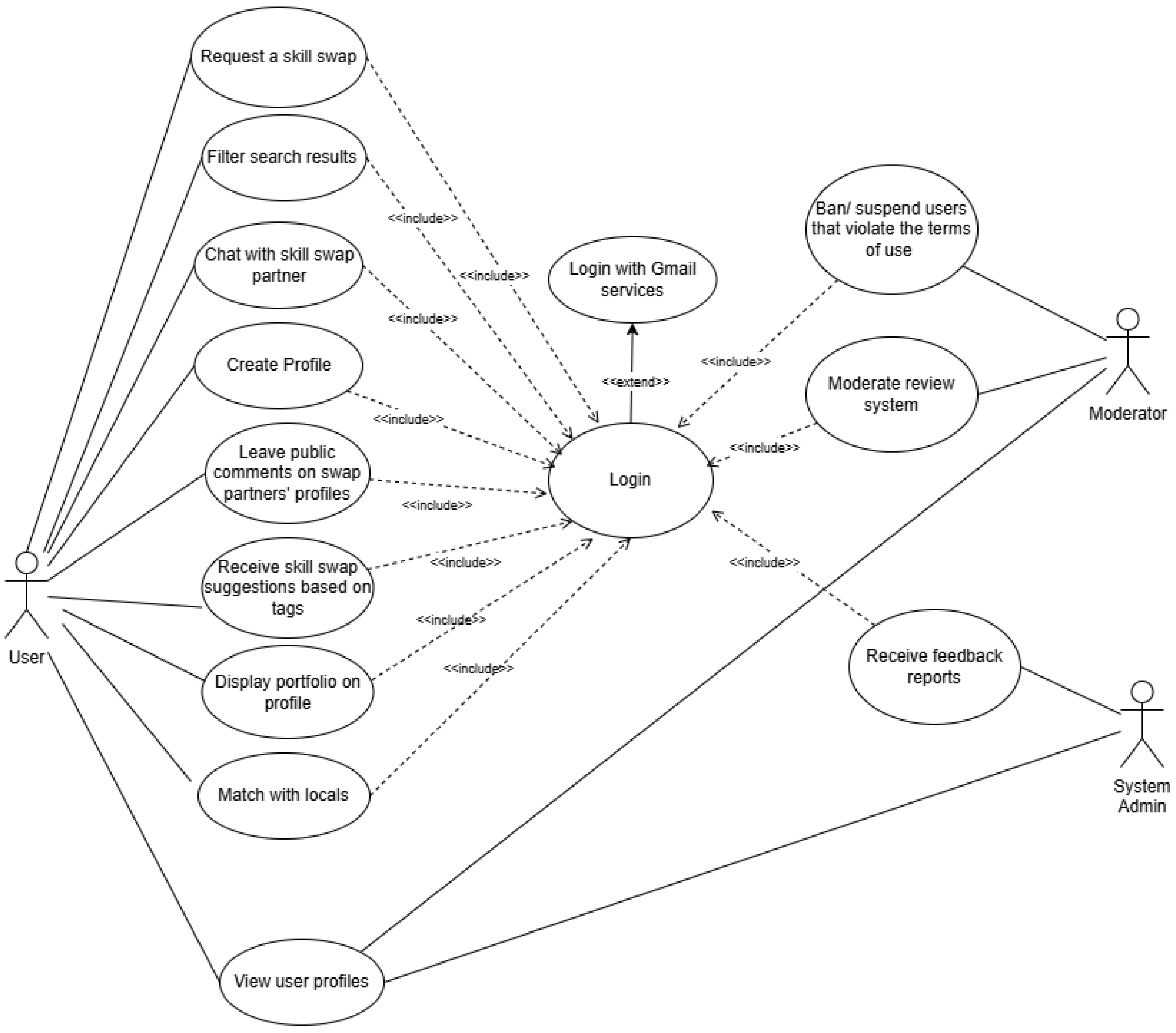
1. The system shall not allow users to only learn a skill, they must also teach one.
2. The system shall allow first time users to complete a matchmaking questionnaire/quiz to determine what skills should be placed at the top of the app for the user to see.
3. The system shall allow users to create an account before gaining full access to the platform.
4. The system shall allow only users without an account to browse the existing skill listings and profiles.
5. The system shall allow users to upload PDF, image and video files to their accounts as proof of their skills.
6. There shall be secure communication (messaging) for users to discuss exchanges before confirming.
7. The system shall allow users to view other user profiles.
8. The system shall allow users to send and receive swap requests.
9. The system shall allow both automated and manual matchmaking for the swapping of skills.
10. The system shall group skills based on tags/categories which can be used in filters to refine the search for a particular set of skills.

Non-Functional Requirements

1. The system shall have fast response times for searches and matchmaking.
2. The system shall be optimised for mobile use.
3. The system shall provide a seamless matchmaking experience.
4. The system shall encrypt user messages.
5. The system shall have Two-Factor Authentication integrated with the authenticator app for security.
6. The system shall have monthly scheduled downtime for maintenance between 2:00AM and 5:00AM.
7. The system shall be capable of efficiently managing increased user demand while maintaining performance and stability.
8. The system shall keep user data anonymous in analytics.
9. The system shall support 24/7 access to core features except during scheduled maintenance.
10. The system shall ensure data backups are performed daily and stored securely.

Use-Cases

Use Case Diagram



Request a skill swap
Filter search results
Chat with skill swap partner
Create Profile
Leave public comments on swap partners' profiles
Receive skill swap suggestions based on tags
Display portfolio on profile
Match with locals
View user profiles
User
Login with Gmail services
<<extend>>
Login
<<include>>
Ban/ suspend users that violate the terms of use
Moderate review system
Receive feedback reports
Moderator
System Admin

Figure 1: Use Case Diagram

Ranking of Use Cases

Use Case Name	Ranking criteria						Total score	Priority
	1	**2**	**3**	**4**	**5**	**6**	**30**	
Create profile	5	5	2	1	5	4	22	High
Filter search results	5	4	4	3	5	3	24	High
Request a skill swap	5	4	4	3	5	4	25	High
Leave public comments on Swap partners' profiles	3	3	2	2	3	3	16	Medium
Chat with Skill Swap Partner	4	5	4	4	4	3	24	High
Receive Skill Swap suggestions based on tags.	5	4	3	3	5	4	24	High
Ban/suspend users that violate the terms of use.	3	3	3	2	1	1	13	Medium
Receive feedback reports	2	2	2	1	1	1	9	Low
View user profiles	5	4	2	1	5	4	22	High
Display portfolio on profile	3	4	2	2	2	2	15	Medium
Match with Locals	3	3	4	4	5	4	23	High
Receive feedback reports	4	4	3	3	3	4	21	High
Moderate review system	4	3	2	2	4	3	18	Medium

Table 1: Ranking of Use Cases

Expanded Use-Case

Skill Exchange System

Author(s): <u>Mykel Dedier</u> **Date: <u>08/04/2025</u>**

Version: 1.1

Use-Case Name:	Request Skill Swap	**Use-Case Type:** **Primary Use Case**
Use-Case ID:	USS-RSS002.00	
Priority:	High	
Source:	User Made Skill Swap Request	
Primary Business Actor:	User	
Other Participating Actors:	Skill Learners (A User) Skill Providers (A User)	
Other Interested Stakeholders:	System Admin - Interested in the proper functioning of this feature to ensure the application operates as intended without error and inconvenience to the user Moderator - Interested in the smooth transaction of skills in order to fulfill the purpose of this application	
Description:	This use case describes the event of a user requesting to engage in a skill swap with another user. A user can request a skill swap after receiving automated matchmaking suggestions or by browsing user profiles and skill listings. If a match is found, the other party is notified and can accept or reject the request. Acceptance confirms the swap, allowing scheduling via private messages and a built-in calendar. If no suitable	

	teacher is available, an AI system generates lessons to help the user learn the skill.

Precondition:	The individual requesting a skill swap must be a registered user with an account on the platform.	
Trigger:	This use case is initiated when a request for a skill swap is made.	
Typical Course of Events:	Actor Action	System Response
	Step 1: The user initiates a skill swap request using the automated matchmaking system, through user profiles or searching on the skill listings.	Step 2: The system verifies the request details and ensures all required information is provided. Step 3: If using automated matchmaking, the system suggests potential skill swap partners based on prior user submitted information. Step 4: If browsing manually, the system retrieves and displays relevant user profiles and skill listings as requested.
	Step 5: The user selects a potential skill swap partner and submits the request.	Step 6: The system notifies the selected user of the pending request. Step 7: The notified user reviews the request and either accepts or rejects it. Step 8: If accepted, the system confirms the swap and enables scheduling features.

	Step 9: The users coordinate via private messages and the built-in calendar to arrange a date and time for the skill swap.	Step 10: The system records the scheduled session and updates the users' calendars accordingly.
Alternate Courses:	Alt-Step 2: If no suitable partner is found teaching the desired skill, the system generates AI-driven lessons to help the user learn the desired skill. Alt-Step 3: If the individual receiving the skill swap request rejects it, the user has to find another person to request a skill swap with.	
Conclusion:	This use case concludes on the successful arrangement of skill swapping between two users, the rejection of a skill swap or the fallback option of AI-generated lessons.	
Postcondition:	Regardless of the conclusion, the system logs the event and ensures no pending actions remain.	
Business Rules:	There can be no monetary gain involved in these transactions of skills	
Implementation Constraints and Specifications:	The system should handle a large number of concurrent users without performance issues. The matchmaking algorithm should consider user preferences, skill levels, and availability.	
Assumptions:	Users accurately list the skills they can teach and the skills they want to learn. The AI-generated lessons are an acceptable alternative if no human teacher is available The built-in calendar and scheduling features function without major technical issues. Users have stable internet access to use real-time features like messaging and scheduling	

Open Issues:	What happens if a user agrees to a skill swap but does not follow through? How effective and personalized will the AI-generated lessons be? How will the system handle users in different time zones?

Figure 2 : Expanded Use Case

Sequence Diagram

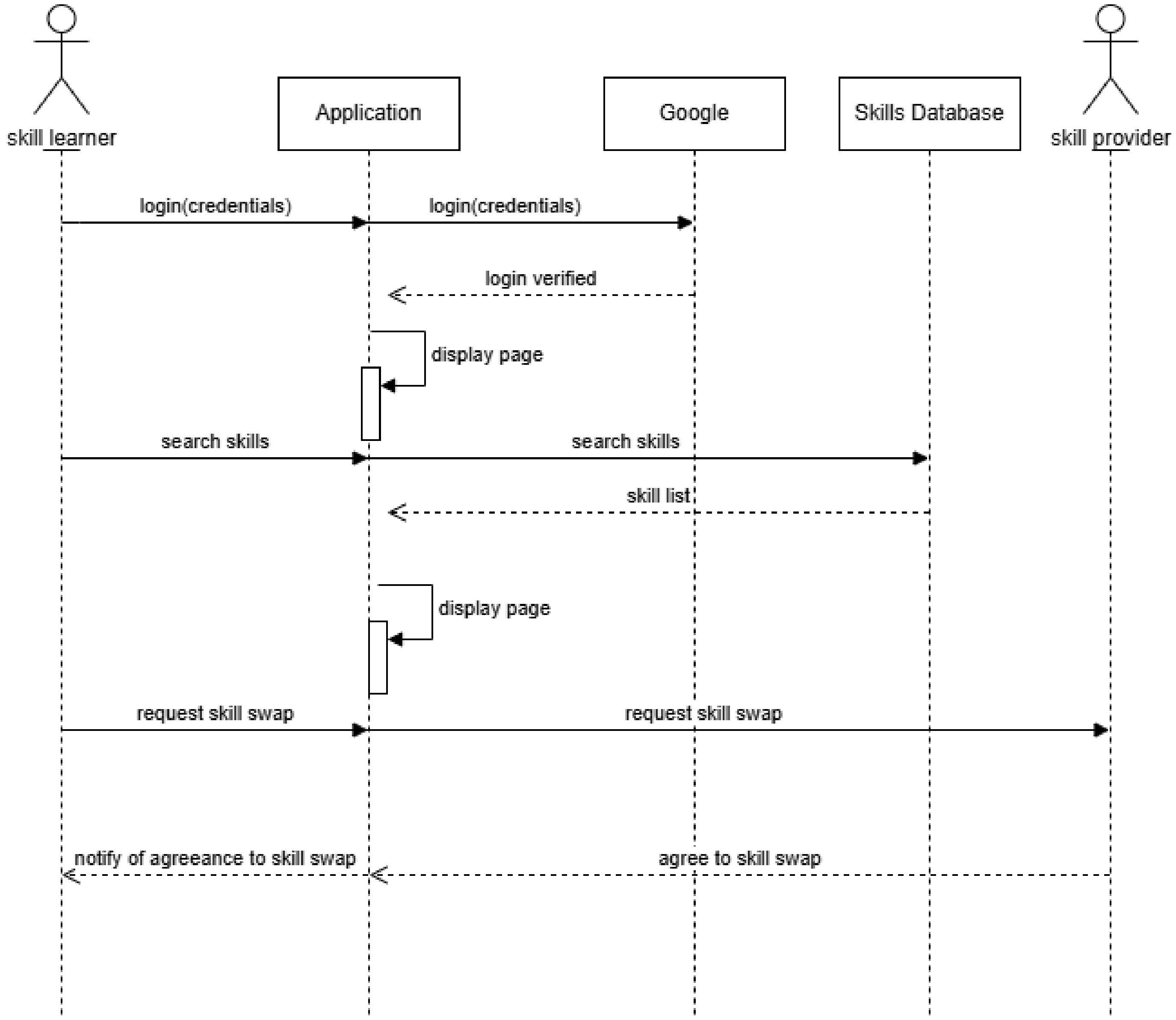


Figure 3: Sequence Diagram

Class Diagram

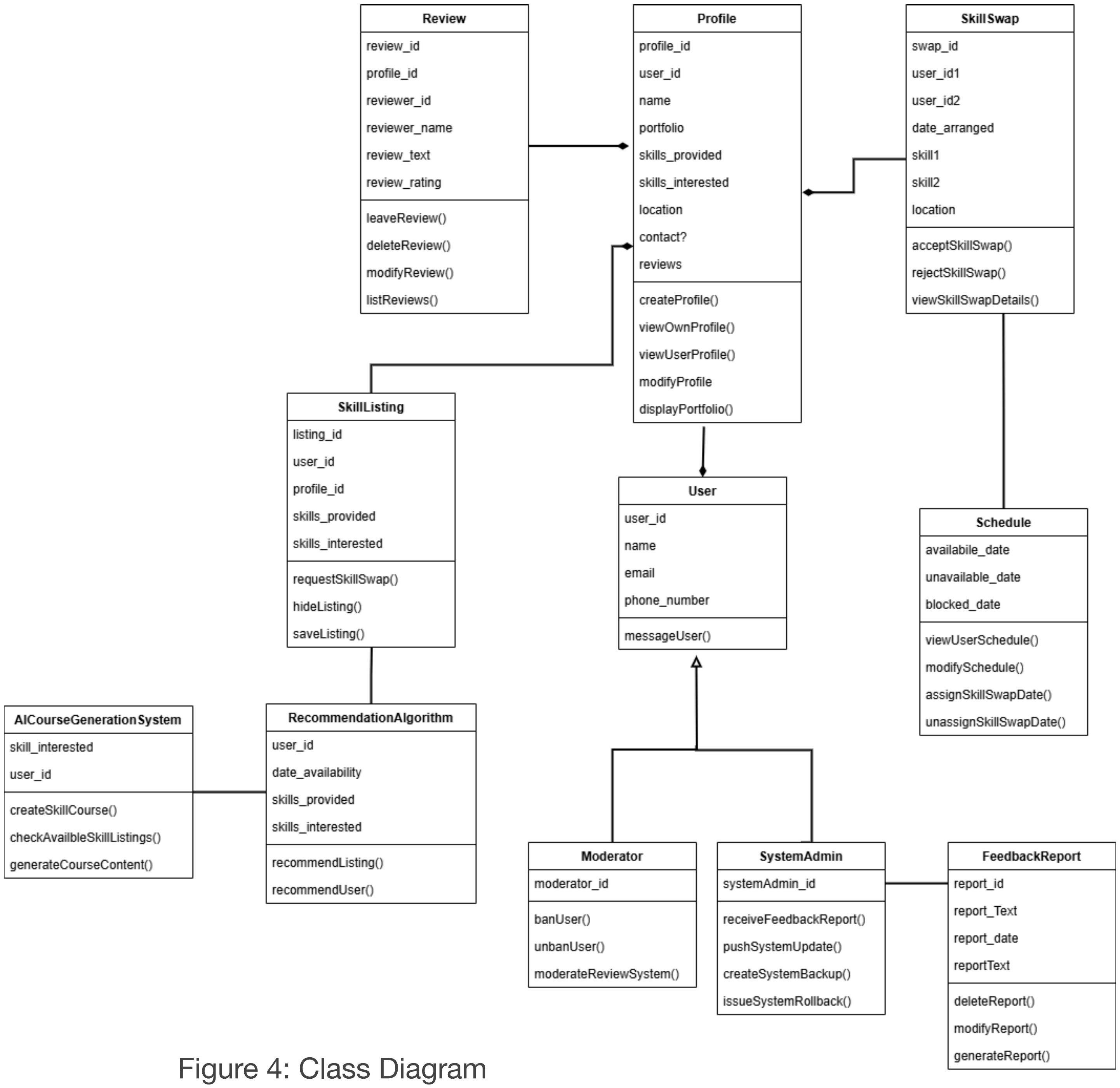


Figure 4: Class Diagram

Testing Plans

Project Name: Skill Share Platform

Test Case ID: SS-1	**Test Designed by:** Reyneka Joe
Test Priority (Low/Medium/High): Medium	**Test Designed date:** 11/4/25
Module Name: Skill Swap Modify Report	**Test Executed by:** Reyneka Joe
Test Title: Modify a Report	**Test Execution date:** 13/4/25
Description: Test the Skill Swap report modifier feature	

Pre-conditions: System Admin has already logged into the app

Dependencies:

Test No	Test Description	Test Steps	Test Data	Expected Result	Actual Result	Status (Pass/ Fail)	Notes
TU01	System admin modifying the date on a Report with valid date	1. Select a report	15/4/ 2025	System admin would be allowed to save the report.	As expecte d.	Pass	
		2. Change the date of the report					
		3. Save report					

Test No	Test Description	Test Steps	Test Data	Expected Result	Actual Result	Status (Pass/Fail)	Notes
	System admin modifying the date on a Report with invalid date	1. Select a report	15/4/ 2028	System admin should see the error message for invalid date	As expecte d.	Pass	
		2. Change the date of the report					
		3. Save report					

TU02							

Figure 5: Testing Diagram SS-1

Project Name: Skill Share Platform

Test Case ID: SS-2	**Test Designed by:** Reyneka Joe
Test Priority (Low/Medium/High): High	**Test Designed date:** 11/4/25
Module Name: Skill Listing Recommendation	**Test Executed by:** Reyneka Joe
Test Title: Recommend Users based on skills	**Test Execution date:** 12/4/25
Description: Test the recommendation algorithm	

Pre-conditions: User has already logged into the system

Dependencies:

Test No	Test Description	Test Steps	Test Data	Expected Result	Actual Result	Status (Pass/Fail)	Notes
TU01	User viewing skill listing	1. Enter skill into search bar	Playing the guitar	User should see a page loaded listing the available skill providers offering to teach a user how to play the guitar	As expected.	Pass	
		2. Click search bar icon					

Test No	Test Description	Test Steps	Test Data	Expected Result	Actual Result	Status (Pass/Fail)	Notes
	User viewing skill listing	1. Enter skill into search bar	Baking a cake	User should see a page loaded	The user was taken to the	fail	There was a bug in

TU02		2. Click search bar icon		listing the available skill providers offering to teach a user how to bake a cake	home page		the code

Figure 6: Testing Diagram SS-2

Project Name: Skill Share Platform

Test Case ID: SS-3	**Test Designed by:** Reyneka Joe
Test Priority (Low/Medium/High): High	**Test Designed date:** 11/4/25
Module Name: Skill swap accepting	**Test Executed by:** Reyneka Joe
Test Title: Accept Skill Swap	**Test Execution date:** 13/4/25
Description: Test the accept skill swap function	

Pre-conditions: Skill provider has already logged into system

Dependencies:

Test No	Test Description	Test Steps	Test Data	Expected Result	Actual Result	Status (Pass/ Fail)	Notes
TU01	A skill provider accepting a skill swap with another user	1. Open notifications	Jane Doe- Candle making	Skill provider will see a page loaded that says you've got a match	As expected.	Pass	
		2. Check skill swap requests					
		3. View each user from the requests					
		4.Accept a skill swap					

Figure 7: Testing Diagram SS-3

Project Name: Skill Share Platform

Test Case ID: SS-4	**Test Designed by:** Reyneka Joe
Test Priority (Low/Medium/High): High	**Test Designed date:** 11/4/25
Module Name: Message user	**Test Executed by:** Reyneka Joe
Test Title: Message a user in app	**Test Execution date:** 11/4/25
Description: Test the message function	

Pre-conditions: User has valid username and password

Dependencies:

Test No	Test Description	Test Steps	Test Data	Expected Result	Actual Result	Status (Pass/ Fail)	Notes
TU01	User viewing skill listing	1. Enter skill into search bar	Playing the guitar	User should see a page loaded listing the available skill providers offering to teach a user how to play the guitar	As expecte d.	Pass	
		2. Click search bar icon					

Figure 8: Testing Diagram SS-4

Project Name: Skill Share Platform

Test Case ID: SS-5	**Test Designed by:** Reyneka Joe
Test Priority (Low/Medium/High): High	**Test Designed date:** 11/4/25
Module Name: Creating a profile	**Test Executed by:** Reyneka Joe
Test Title: Create User Profile	**Test Execution date:** 13/4/25
Description: Test the create user profile page	

Pre-conditions: User has never created a profile before

Dependencies:

Test No	Test Description	Test Steps	Test Data	Expected Result	Actual Result	Status (Pass/ Fail)	Notes

TU01	User creating a profile	1. Click the next button on loading page	Jane Doe 5/4/2001 555-5556 Arima jdoe@gmail.com 12345678 Baking, carpentry Jewelry making, essay writing	User should be taken to the home screen	As expected.	Pass	
		2. Enter credentials on sign up page then click sign up button					
		3. Select and or add the skills you have to offer then click next					
		4. Select and or add the skills you want to learn then click next					

Figure 9: Testing Diagram SS-5

Acceptance Testing

"Acceptance testing is an inherent part of custom systems development. Customers test a system, using their own data, and decide if it should be accepted from the system developer." — Software Engineering Tenth Edition by Ian Sommerville

Overview
The Skill Swap platform is a user-facing, interactive application. The process of acceptance testing for the platform should be both functional and user-centric. In other words, it should ascertain whether the system meets its requirements and behaves as users expect.

Define Acceptance Criteria
Stakeholders would be intimately involved in the definition of acceptance criteria. For the Skill Swap platform, the basis of its acceptance criteria will come about from the already established functional and non-functional requirements, after negotiations between the developers, product owner and end-users. As a precaution, the most critical and necessary functionality should be prioritized, with secondary functionality taking a backseat in terms of determining whether the system is acceptable. This would allow the system to be operational at an earlier time frame, and those secondary requirements would be implemented with the deployment of later versions.

Plan Acceptance Testing
Testing schedules would be established with a detailed account of the resources allocated for conducting acceptance tests. The specifics of this process would be determined largely by the developers, with input from stakeholders such as the product owner on matters such as budget allocations, timeframes, testing order, etc.

The approach to testing would be scenario-based. Stakeholders such as users, moderators and admins are to be consulted in the creation of scenarios that models their expected interaction with the system and the outcome of that interaction. These scenarios act as test cases, while simulating realistic workflows and ensuring the entire path of interaction from start to end is covered. Each of such test cases must have clear established acceptance criteria that testing will use to validate whether requirements are met.

The developers should also outline potential risks to testing and discuss mitigation strategies.

Derive Acceptance Tests
Test cases are developed around requirements that have been translated into acceptance criteria, with stakeholders corroborating the exact scenario. Fully-fledged scenarios, user stories, use cases and their variants are to be utilized as necessary for this stage of the process.

> *Example Scenario (User Story):*
> As a user, I want to search for skill offers by location and rating, so that I can find high-quality, nearby matches.
>
> *Acceptance Criteria:*
> Results are able to be sorted and filtered.
> Location filter shows only relevant results.
> Rating filter shows only relevant results.
> Results fulfill all filtered criteria.
> Results are displayed within no more than 5 seconds.

These tests would aim to validate requirements, ideally encompassing both functional and non-functional requirements.

Run Acceptance Tests
The system would be deployed within an appropriate environment to facilitate the execution of acceptance tests. The ideal environment would be that within which the system is intended to operate. As the Skill Swap platform is a mobile application, there is no true singular environment where the platform is supposed to operate in.

As such an approach similar to beta-testing could be taken, where a select pool of end-users are granted access to download and use the platform on their personal mobile device. In that case, users can report issues of their own volition and there should be some means of remotely monitoring user activity and the outcomes. This also serves to highlight problems that might arise from a particular mobile platform.

However, there would surely be a dedicated testing environment populated with user data, that end-users can interact with to run through scenarios as developers monitor and document the interaction. Where possible automated testing would be employed for test cases such as registration, logging in, proposing skill swaps, etc. More manual tests would be required for cases such as accessing the user interface and overall user experience.

Negotiate Test Results
It is expected that not all acceptance tests will be passed. If requirements were prioritized accordingly during the definition of acceptance criteria (e.g. via a priority matrix), the product owner and developers can utilize this information to determine if the system is in an acceptable state for deployment, despite the system's failure to succeed in all test cases. These negotiations should also encompass necessary fixes to critical functionality if needed and a timeline for the correction of features which failed to pass their test cases.

Reject/Accept System
It must be noted that acceptance testing is not black and white. If the system does not exactly match up to specification and user expectation, it is still negotiable for the system to be deployed. Of course, that may be on the basis that urgent issues be fixed and any missing or mismatched functionality would be corrected and implemented in future versions of the system.

Risk Management

Risk	Probability	Effects	Affects	Strategy Category	Strategies to be Employed
Media uploads exceed storage limits.	Medium	Serious	Product	Mitigation	Enforce file size limits and monitor upload activity.
Location permissions are denied, affecting search results.	Medium	Tolerable	Product	Contingency	Offer manual location entry.
Abuse of feedback system.	Medium	Tolerable	Product	Avoidance	Implement flagging/reporting and introduce review thresholds.
Key team members leave mid-project, causing delays	Medium	Serious	Project	Mitigation	Cross-train team members and document processes.
Unavailability of third-part services(e.g., scheduling, encryption, messaging).	Medium	Serious	Product	Contingency	Implement simple backup communication and scheduling options.
Two-Factor Authentication fails due to third-party service disruption.	Low	Serious	Product	Mitigation	Provide alternative two-factor authentication methods such as email or SMS-based recovery.
Scheduling fails due to time-zone mismatches.	Medium	Tolerable	Product	Mitigation	Integrate automatic time-zone detection and test scheduling features with international users.
User data breach.	Low	Catastrophic	Business	Mitigation	Use strong

					encryption and regular security audits.
Bugs in matchmaking and scheduling due to insufficient testing.	High	Serious	Product	Mitigation	Write detailed test cases and perform user acceptance testing.
Exceeding cloud usage limits.	Medium	Serious	Project	Mitigation	Set usage alerts and utilise auto-scaling to manage user traffic.
Cloud service downtime.	Medium	Serious	Product	Contingency	Use cloud service providers with high uptime and availability.
Delay in content population.	Medium	Tolerable	Product	Mitigation	Pre-populate the platform with sample listings and encourage early users to contribute content.
Inaccurate or biased AI matchmaking recommendations.	Medium	Serious	Product	Avoidance	Use well-documented algorithms and conduct regular fairness and bias audits.
App Crashes under concurrent user load during peak hours.	Medium	Serious	Product	Mitigation	Perform stress and load testing and use cloud auto-scaling for high-traffic moments.
Low user adoption due to competition or lack of demand in the market.	Medium	Serious	Business	Mitigation	Conduct market research and offer incentive for early users.

Table 2: Risk Management

Cost Estimation

The Skill Swap Platform, as a modern software, requires a cost modeling technique that aligns with its software life cycles. COCOMO II (Constructive Cost Model) is well-suited for this platform, as it is designed to estimate costs for modern software projects by building upon the original COCOMO model. This enhanced version offers a more flexible and tailored approach to estimation, delivering precise and comprehensive predictions regarding effort, schedule, and expenses. By adapting its estimations across different phases of the software development lifecycle, COCOMO II provides a more customized and effective strategy for managing costs throughout the project.

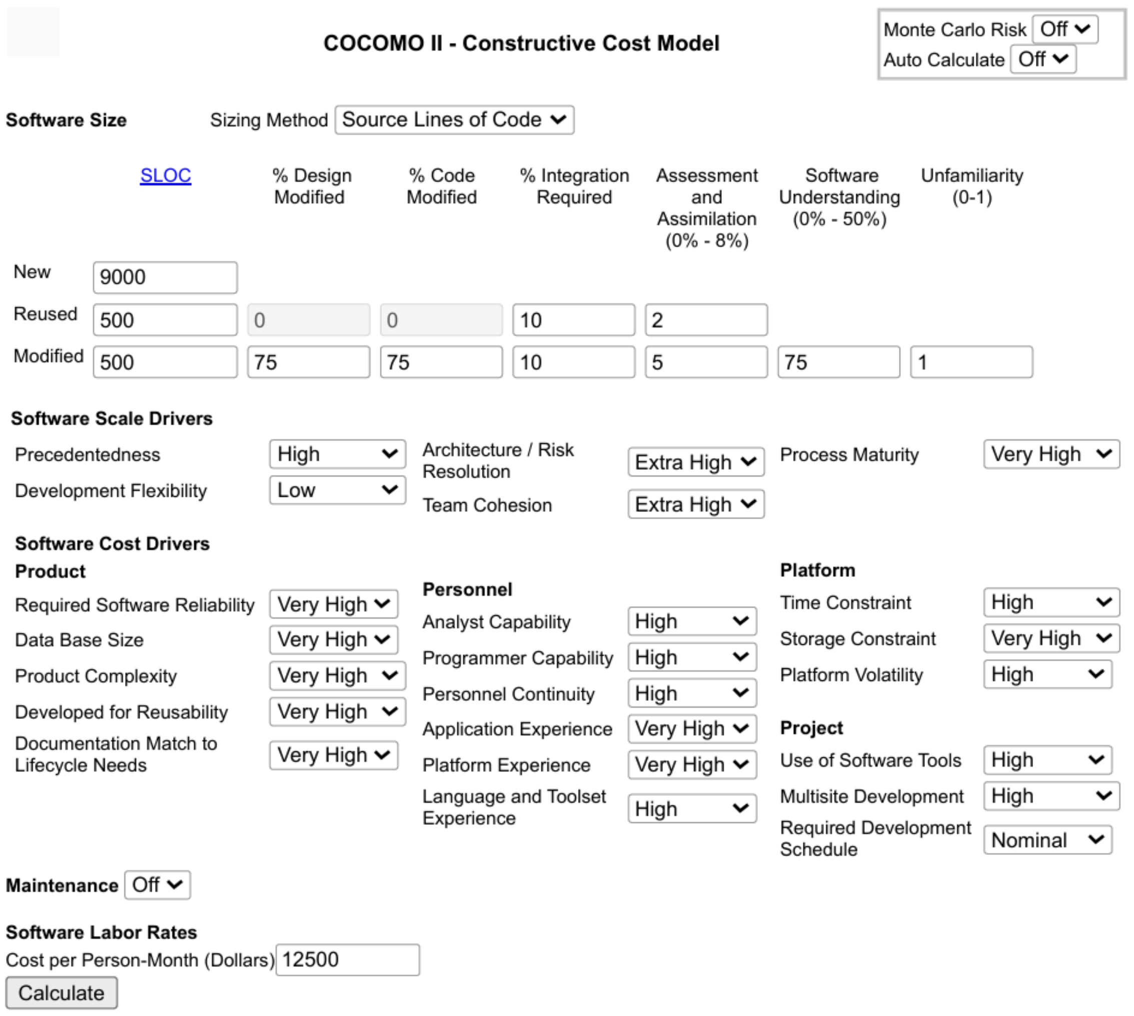
Specification

COCOMO II features a modular structure consisting of three submodels: Application Composition, which focuses on prototyping to address high-risk issues like user interfaces and performance; Early Design, which explores alternative software architectures and concepts of operation; and Post-Architecture, which centers on software development and maintenance, proceeding cost-effectively when the software life cycle is well-developed. It includes a detailed array of cost drivers to improve effort estimation, follows openness principles by making all relationships and algorithms publicly accessible for transparency and integration, and is tailorable to suit factors like project type, development methodology, and software reuse strategy. Furthermore, it offers size estimation models, such as Object Points (counting screens, reports, and third-generation language models), Function Points (measuring information processing functionality), and Source Lines of Code (SLOC).

Justification

COCOMO II offers improved accuracy by factoring in various elements that impact project cost and effort, enhancing its precision. Its modular and tailorable design ensures flexibility, making it adaptable to diverse software development projects and methodologies. Transparency is promoted through the public availability of its core components, enabling better understanding, analysis, and customization. As a framework for continuous improvement, COCOMO II relies on ongoing data collection and analysis. Additionally, it supports modern software development practices such as software reuse, re-engineering, and application generator usage.

Project Cost (Breakdown)



COCOMO II - Constructive Cost Model

Monte Carlo Risk: Off
Auto Calculate: Off

Software Size Sizing Method: Source Lines of Code

	SLOC	% Design Modified	% Code Modified	% Integration Required	Assessment and Assimilation (0% - 8%)	Software Understanding (0% - 50%)	Unfamiliarity (0-1)
New	9000						
Reused	500	0	0	10	2		
Modified	500	75	75	10	5	75	1

Software Scale Drivers

Precedentedness	High
Development Flexibility	Low
Architecture / Risk Resolution	Extra High
Team Cohesion	Extra High
Process Maturity	Very High

Software Cost Drivers

Product

Required Software Reliability	Very High
Data Base Size	Very High
Product Complexity	Very High
Developed for Reusability	Very High
Documentation Match to Lifecycle Needs	Very High

Personnel

Analyst Capability	High
Programmer Capability	High
Personnel Continuity	High
Application Experience	Very High
Platform Experience	Very High
Language and Toolset Experience	High

Platform

Time Constraint	High
Storage Constraint	Very High
Platform Volatility	High

Project

Use of Software Tools	High
Multisite Development	High
Required Development Schedule	Nominal

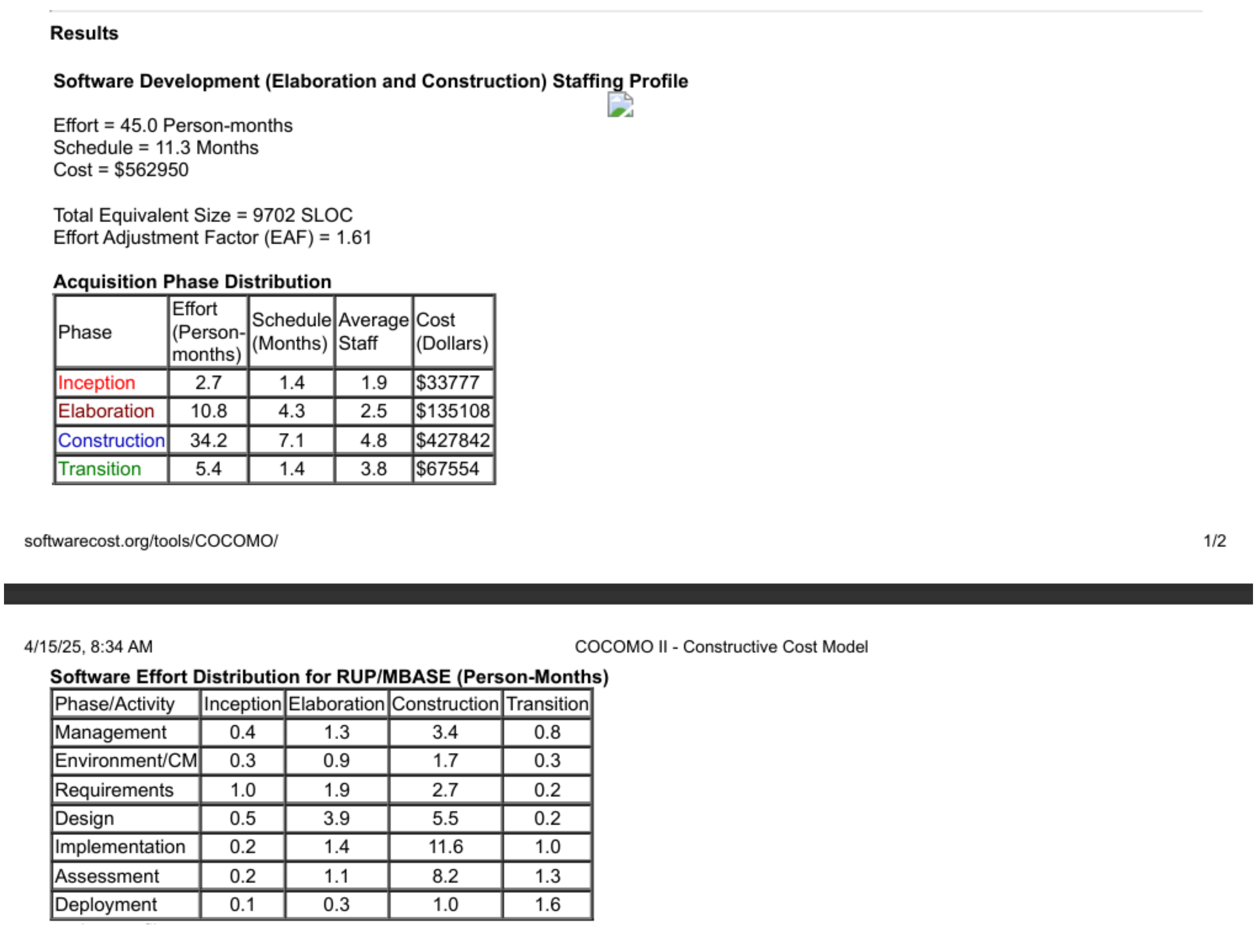
Maintenance Off

Software Labor Rates

Cost per Person-Month (Dollars) 12500

Calculate

Figure 10: Project Cost (Breakdown) part 1



Results

Software Development (Elaboration and Construction) Staffing Profile

Effort = 45.0 Person-months
Schedule = 11.3 Months
Cost = $562950

Total Equivalent Size = 9702 SLOC
Effort Adjustment Factor (EAF) = 1.61

Acquisition Phase Distribution

Phase	Effort (Person-months)	Schedule (Months)	Average Staff	Cost (Dollars)
Inception	2.7	1.4	1.9	$33777
Elaboration	10.8	4.3	2.5	$135108
Construction	34.2	7.1	4.8	$427842
Transition	5.4	1.4	3.8	$67554

softwarecost.org/tools/COCOMO/ 1/2

4/15/25, 8:34 AM COCOMO II - Constructive Cost Model

Software Effort Distribution for RUP/MBASE (Person-Months)

Phase/Activity	Inception	Elaboration	Construction	Transition
Management	0.4	1.3	3.4	0.8
Environment/CM	0.3	0.9	1.7	0.3
Requirements	1.0	1.9	2.7	0.2
Design	0.5	3.9	5.5	0.2
Implementation	0.2	1.4	11.6	1.0
Assessment	0.2	1.1	8.2	1.3
Deployment	0.1	0.3	1.0	1.6

Figure 11: Project Cost (Breakdown) part 2

User Interface Design

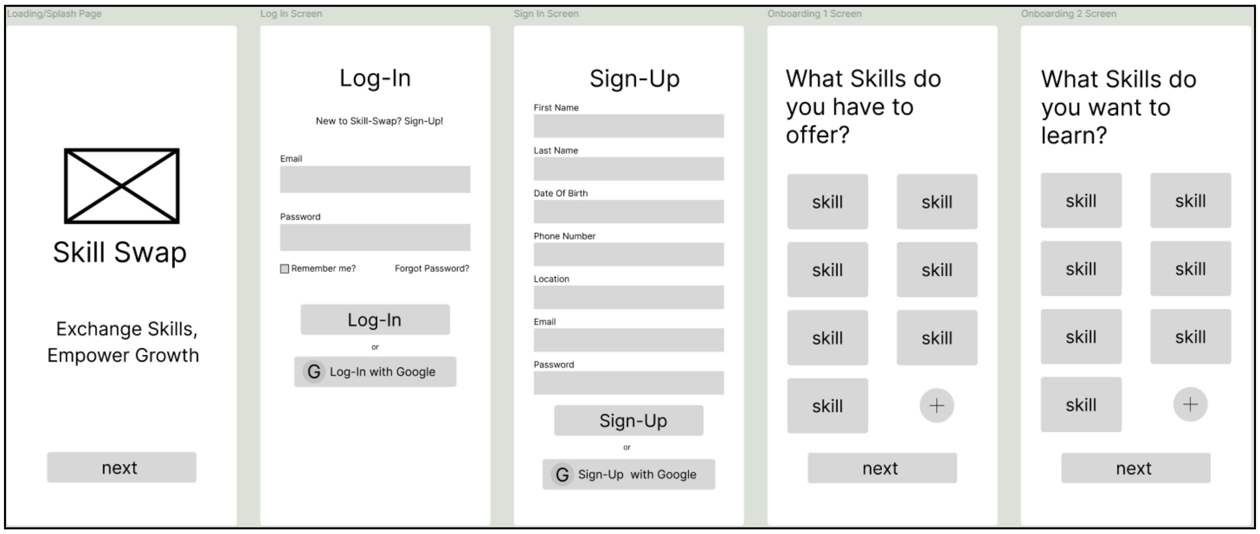


Figure 12: Showing the Loading/Splash Page, Log-In, Sign-Up and Onboarding screens.

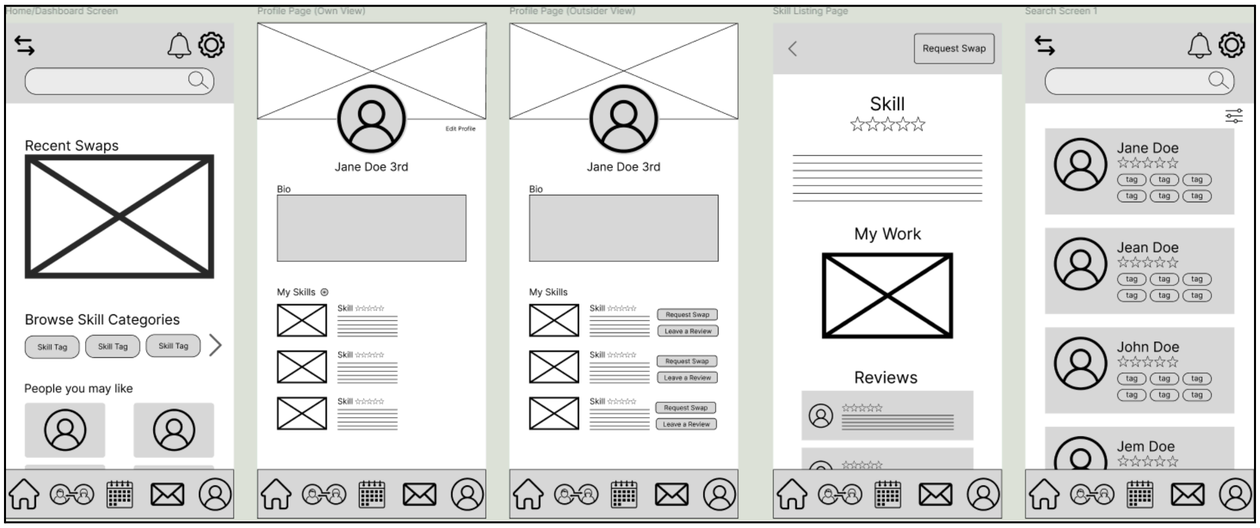


Figure 13: Showing the Home/Dashboard, Profile Screens (both the personal view and external view), Skill Listing Screen and one of the Search Screens.

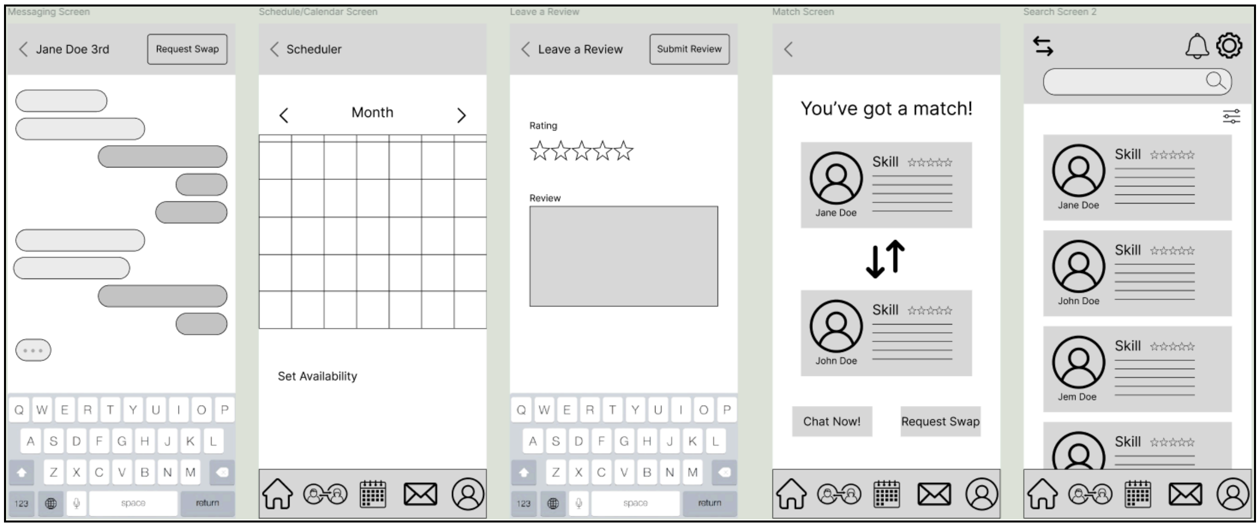


Figure 14: Showing the second Search Screen, Messaging, Scheduling, Feedback and Matchmaking Screens.

Video

Link to Video:
https://youtu.be/5Ebh9QtE9Wg

References

APA 7th Edition

Rahaman, M. (2023, November 30). *How Do I Become an Online Moderator? - Riseup Labs*. Riseup Labs. https://riseuplabs.com/how-do-i-become-an-online-moderator/#:~:text=What%20Does%20an%20Online%20Moderator,adhere%20to%20the%20platform's%20rules.

What Does a System Administrator Do? (With Average Salary). (2025). Indeed Career Guide. https://www.indeed.com/career-advice/careers/what-does-a-system-administrator-do

Fanchi, C. (2022, August 17). *What Is App Scaling and Why It Matters*. Backendless; Backendless Corp. https://backendless.com/what-is-app-scaling-and-why-it-matters/

an. (2010, December 12). *What is the best Time-Of-Day to run the maintenance of an international (English) website?* Server Fault. https://serverfault.com/questions/211701/what-is-the-best-time-of-day-to-run-the-maintenance-of-an-international-english

COCOMO II Model Definition Manual Acknowledgments. (n.d.). https://athena.ecs.csus.edu/~buckley/CSc231_files/Cocomo_II_Manual.pdf

ERI Economic Research Institute. (2025). *Computer Software Engineer*. Salary Expert. https://www.salaryexpert.com/salary/job/computer-software-engineer/trinidad-and-tobago

Boehm, B., Clark, B., Horowitz, E., Westland, C., Madachy, R., & Selby, R. (1995). Cost models for future software life cycle processes: COCOMO 2.0. *Annals of Software Engineering*, *1*(1), 57–94. https://doi.org/10.1007/bf02249046

COCOMO 2. (2025). Softstarsystems.com. https://www.softstarsystems.com/cocomo2.htm

COCOMO II - Constructive Cost Model. (2025). Softwarecost.org. http://softwarecost.org/tools/COCOMO/

IEEE Recommended Practice for Software Requirements SpeciÞcations Software Engineering Standards Committee of the IEEE Computer Society IEEE-SA Standards Board. (1998). https://www.math.uaa.alaska.edu/~afkjm/cs401/IEEE830.pdf

Merriam-Webster Dictionary. (2025, April 6). Merriam-Webster.com. https://www.merriam-webster.com/dictionary/scenario#:~:text=%3A%20a%20sequence%20of%20events%20especially,course%20of%20action%20or%20events