

STUDYMATCH: Connecting Minds, Shaping Futures



College of Computer and Information Science

Mapua Malayan Colleges Mindanao

Gen. Douglas Mac Arthur Hwy, Talomo,

Davao City, 8000 Davao del Sur

Submitted by:

LOBATON, Gwyn Ann S.

RUSIANA, Jonas Gabriel L.

Project Description:

Study Match is a user-friendly application available on both Android and iOS platforms, developed by Gwyn Lobaton and Jonas Rusiana of Team Iced Tea Pioneers. This app facilitates a connection between tutors and students, providing support in subjects where students need help. Its primary goal is to assist students struggling with specific subjects while enabling tutors to earn money by offering their tutoring services at their chosen hourly rates. The app matches students with tutors based on location, the tutor's expertise, special needs cases, and budget. Additionally, it includes a search option allowing users to find tutors by location or skill.

Requirements Summary:

		iOS	Android
MINIMUM REQUIREMENTS	Processor Cores	Dual-core Processor	Dual-core processor
	OS	iOS 12 or later	Android 8.0(Oreo) or later
	RAM	2 GB	2 GB
RECOMMENDED REQUIRMENTS	Processor Cores	Quad-core processor	Quad-core processor
	OS	iOS 14 or later	Android 10.0 (Pie) or later
	RAM	4 GB	4 GB
OTHER REQUIREMENTS	Permissions	Location, Notifications, and Storage	Location, Notifications, and Storage

Table 1. System Requirements

Study Match requires a minimum of a dual-core processor and 2 GB of RAM for smooth performance on both iOS and Android devices. The app is compatible with iOS 12 or later and Android 8.0 (Oreo) or later. For optimal performance, a quad-core processor and 4 GB of RAM are recommended, along with iOS 14 or later and Android 10.0 (Pie) or later. Permissions for location, notifications, and storage are necessary to match students with nearby tutors and securely store user data.

Prototype Description:

The prototype was developed using Figma, an interactive prototyping software and website known for its robust design and collaboration features. Figma allows developers to create dynamic and engaging prototypes that can be easily shared with testers through generated links. This facilitates seamless feedback and iteration, enhancing the overall development process.

STUDYMATCH Figma Link

The provided link is the Figma mockup and prototype of the StudyMatch Prototype:

<https://www.figma.com/design/dLYFuQqLpuyu9GH9i7cMF9/STUDYMATCH?node-id=0-1&t=o3vEy9GAIp0VJhtu-1>

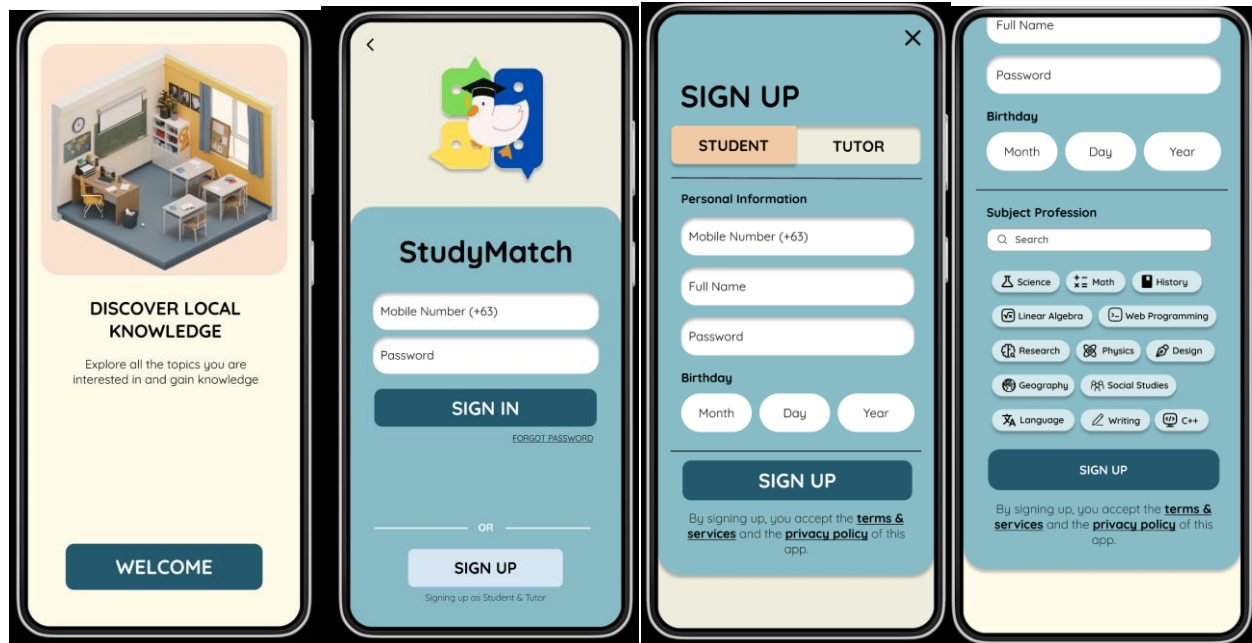
User Scenario:

Yuki and Kaori have been struggling to find a suitable tutor, which has been impacting their school performance. They prefer in-person tutoring, but finding local tutors has been challenging. Online tutoring is not their preferred method because they want an in-depth learning with the subject along with the tutor beside them to guide.

One day, Kaori stumbled upon an app called STUDYMATCH while searching for tutoring options online. They were amazed to find a wide range of tutors available in their area through the app. Impressed by its functionality, they shared STUDYMATCH with

Yuki, who was initially skeptical but ended up finding it remarkable. Yuki even started recommending the app to colleagues in need of tutoring help.

STUDYMATCH Mock-up/Prototype:



Prototype Flow:

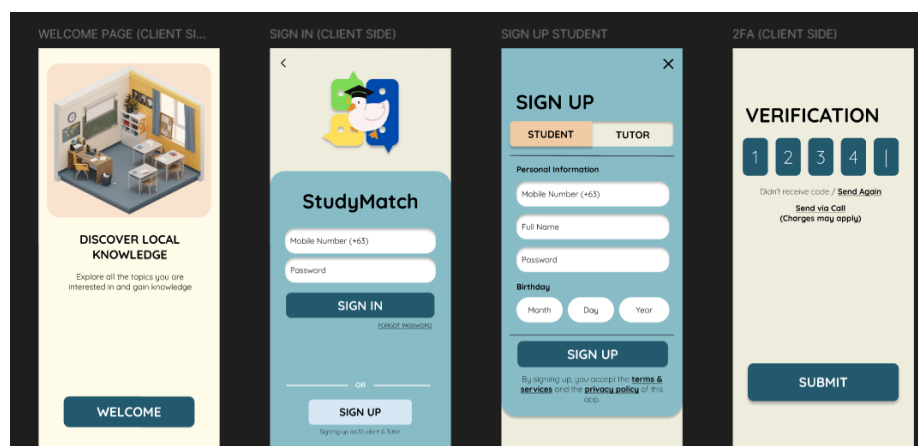


Figure 1. User Authentication

Figure 1 shows the user authentication process of the two user type categories.

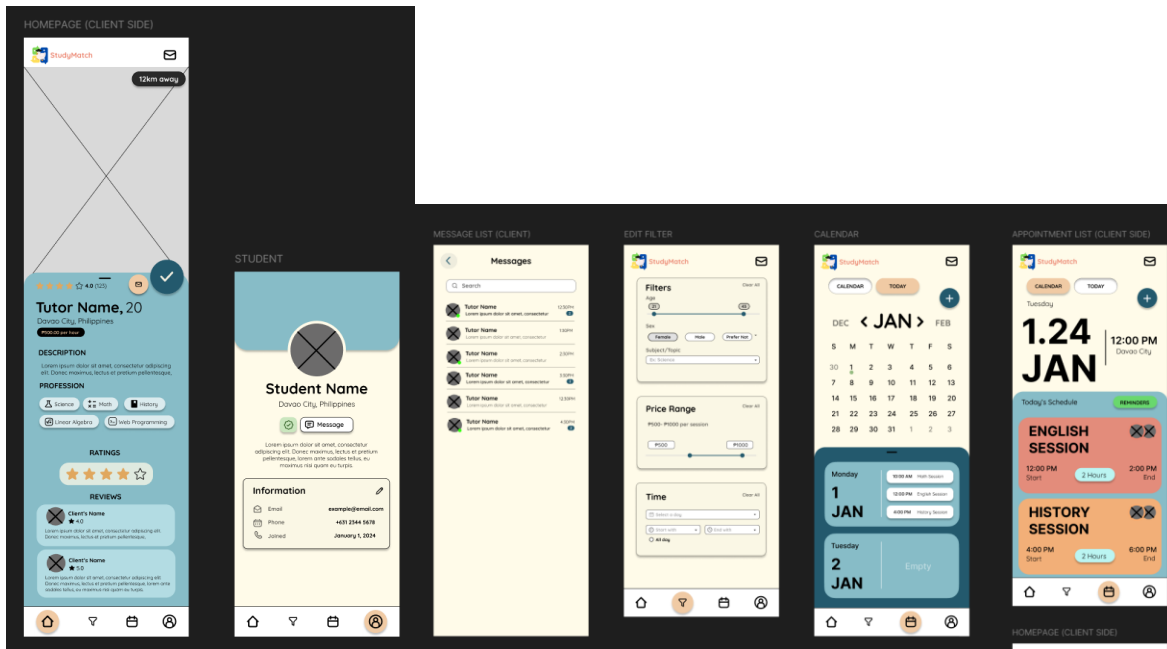


Figure 2. Student Standard View

Figure 2 shows the standard view on the student side. With the navigation bar at the bottom, the flow of the said views is arbitrary and doesn't really matter, as any of those buttons can transfer to each other.

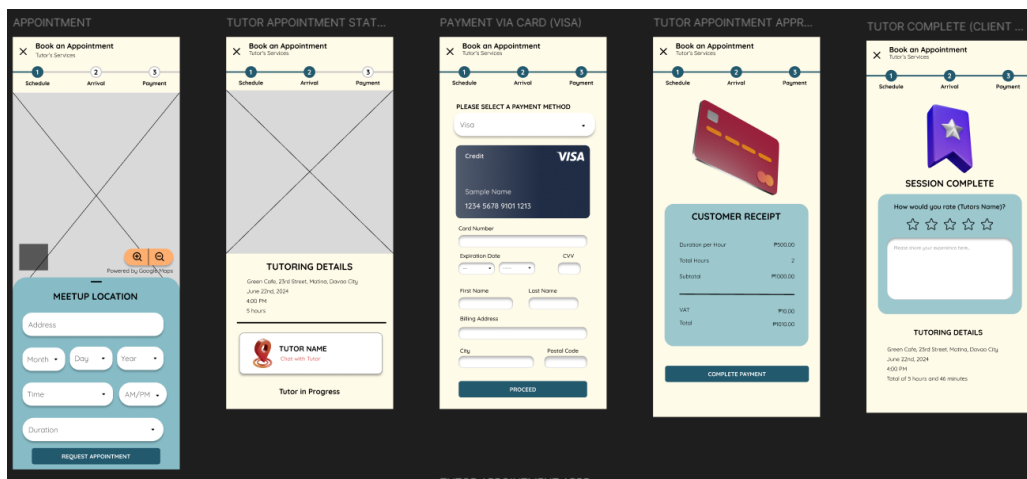


Figure 3. Student Booking Flow

Figure 3 shows the booking procedure with one of the tutors. It also demonstrates when a tutoring session is completed, along with the payment flow and review flow.

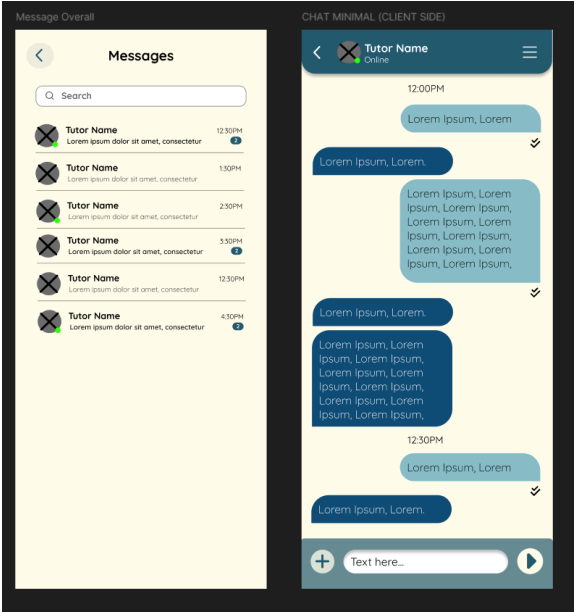


Figure 4. Communication Flow

Figure 4 is the views involved with communication, and its basic flow on communicating with different students and tutors. Both the student and tutor view share the same communication flow and view.



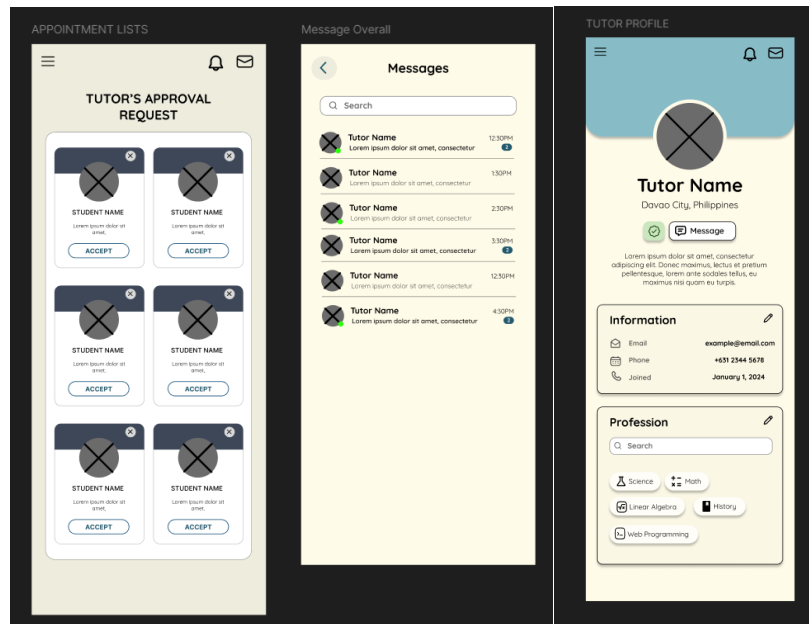


Figure 5. Tutor Standard Flow

Figure 5 shows the standard view on the tutor side. With the navigation bar at the side (from the three bars icon), the flow of the said views is arbitrary and doesn't really matter, as any of those buttons can transfer to each other. The difference between student and tutor is that the tutor view maximizes space with the exclusion of the navigation bar at the bottom.

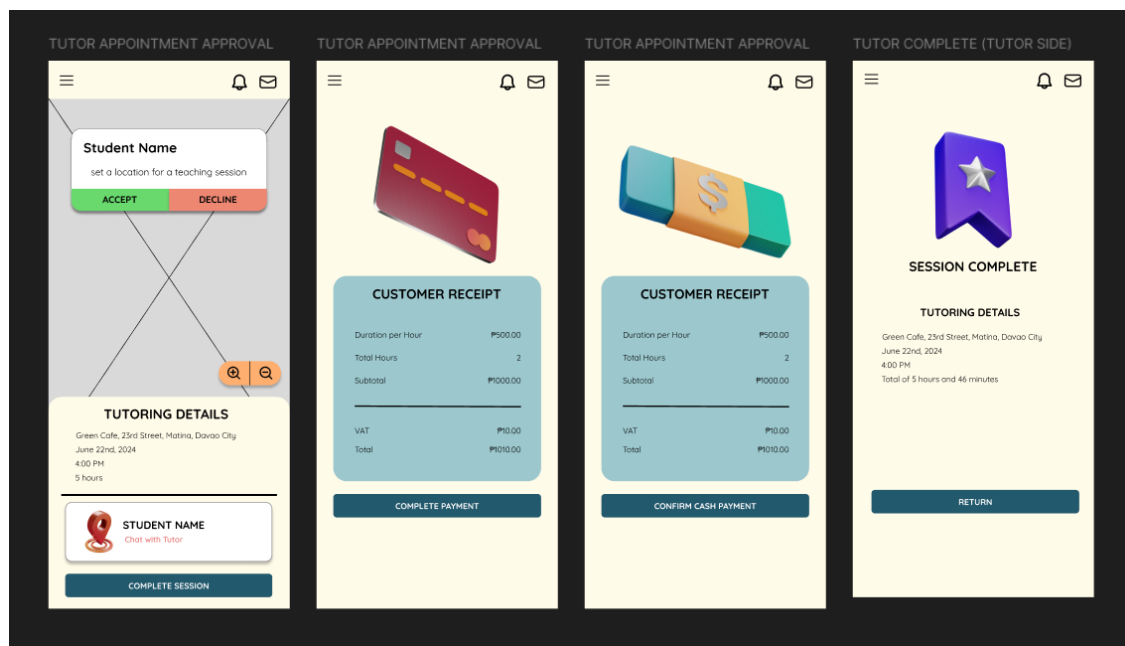


Figure 6. Tutor Booking Flow

Figure 6 shows the booking flow of the tutor. The tutor controls the appointment details in terms of whether the tutoring session is completed or not, and whether or not the tutor received the cash payment if ever the student chooses a cash payment option.

Rationale:

The team utilized Figma as the design platform for websites and prototypes due to its collaborative features, cross-platform accessibility, robust prototyping tools, version control capabilities, support for design systems, and ease of sharing. Figma enables real-time collaboration among team members, regardless of their location or device, streamlines the prototyping process with interactive tools, ensures version control and consistency with design systems, and simplifies sharing designs for feedback and collaboration, making it a comprehensive and efficient solution for our design needs.

Changes to the Requirements:

The changes made to the requirements are only towards the user requirements, which refers to more ease of navigation when it comes to flow, and a much more easier way to create appointments without getting confused. These additional changes made the prototype a bit more ergonomic in terms of user experience but may require a lot of UI views in regard to the app being more user friendly. Apart from that, no further changes has been made to the requirements of the design.

Initial Evaluation Plan:

Usability Specifications

The following objectives will be pursued in the development of this prototype when it comes to user appeal:

- **User-friendly Interface:** Verify that the application is accessible to users of all ages with parental supervision for the children and technical skills using intuitive and simple navigation on the user interface.
- **Personalization:** Let users to personalize their profiles and preferences to get notifications and recommendations from different tutors that are specifically catered to the people in need.
- **Efficient Matching System:** Provide a reliable algorithm that pairs students with qualified tutors according to criteria including budget, specific requirements, area of expertise, and proximity.
- **Feedback Mechanism:** Establish a feedback system that allows students to review and rate tutors, to ensure that best quality and continuous improvement of service.
- **Security and Privacy:** By implementing rigorous data protection guidelines and open policies, the team can ensure the protection towards security and privacy.
- **Business Oriented System:** Provide features that assist with the business side of tutoring, like safe payment processing, financial tracking for tutors, and marketing resources to assist tutors in expanding their client base.

Population

The minimum population selected students and working students that will participate in using the prototype will be 15 people, while the minimum population will be

20 people. 75% of the population will be doing generic tasks that correspond to the generic system flow of the application. While 25% of the population are required to do certain tasks that correspond to business management as a tutor. The StudyMatch prototype must be able to do every single task that was proposed to be considered complete.

Prototype Tasks

The tasks for the prototype will be split into two major categories: (1) the student view, and (2) the tutor view. Furthermore, the student view three (3) sections of tasks: (1) finding & communicating with a tutor, (2) accomplishing a service and transaction with a tutor, and (3) rating and service reoccurrence. As for the tutor view, there are three sections of tasks: (1) client communication, (2) business setup, and (3) service transaction. The following table are some tasks under these categories and sections:

Student View	
Finding & Communicating with a Tutor	Choosing Tutors from a local area.
Accomplishing a Service & Transaction with a Tutor	Paying the tutor after the tutoring.
Rating & Service Occurrences	Planning a future tutor.

Tutor View	
Client Communication	Chatting with your clients.
Business Setup	Setting up your business profile.
Service Transaction	Completing a transaction and receiving the cash from the app.

Heuristic Evaluation

The evaluation of the StudyMatch will be evaluated with the 10 Usability Heuristic Method of Evaluation. The following are the evaluation of the design:

Visibility of System Status

The prototype will be brief and straightforward when giving certain information to the users, enabling the users to be informed easily without having to think too much about the given information.

Match Between the System and the Real World

The prototype doesn't use any informal words and jargons, instead, it uses simple words for those who don't understand English enough. Universally accepted words, phrases, and concept to certain features are used to easily identify those features.

User Control and Freedom

The prototype has an intuitive navigation system that guides the users to different sections of the prototype without getting lost. Communicating with service providers to clients uses a platform's standard keyboard and media system to use not only text but also multimedia for an effective communication between service providers and clients.

Consistency and Standard

The used symbolism are different enough to not confuse the users, and properly integrates the universally accepted symbolism for user interface and interaction. The elements of the UI also follow a certain design standard implemented during the prototyping phase for reusability and consistency.

Error Prevention

Error messages use simple words, and any commonly encountered error will have their own distinct symbolism in order to communicate the problem with the User and may be able to fix the problem on their end. The application is also not complex

enough to break it easily, and the entire flow of the user is completely guided as to not be astray to their original purpose (i.e., idiot proofing).

Recognition Rather than Recall

The prototype uses commonly used symbols and universally accepted object and actions in order for the user to be familiar with at first use. So, the user does not have to give a lot of effort just to use the app.

Flexibility and Efficiency of Use

The prototype makes use of simple UI component as to not waste the system resource of the user and does provide a comprehensive menu for them to navigate to the major aspects of the application.

Aesthetic and Minimalist Design

The aesthetic follows a flat design, with the heavy emphasis of colors and simple design elements to help make things interesting. The prototype shouldn't also provide a lot of information in one page, and only shows the relevant and important information of the said section.

Help Users Recognize, Diagnose, and Recover from Errors

The error handling of the prototype uses simple words and provides links to resources on how to solve the error in the user's end. Like solving the network connectivity of the app. It also does not provide any error codes, and each distinct error should be described and documented in order to recover from errors.

Help and Documentation

The UI design of the prototype is a guiding force to use the app, however in cases where the UI is not enough, a help section is provided for certain or generic actions to be used. And apart from that, any online information will be provided as resources to help the user from different problems.

Participant Survey and Feedback

After the testing of the prototype by the designers, a participant survey and feedback will be conducted:

Data Gathering Method	Description
Survey – Quantitative	A survey will be handed out to the participants to gather the participant's user experience when handling the prototype. Which will be interpreted from a 5-point Likert Scale.
Feedback – Qualitative	After the survey, the participants will be asked of their feedback in regard to their experience. This is useful for further insights that might not be covered by the survey.

The following table is the content of the survey and their method of answer (either a long answer or a points-based answer):

This table can be accessed to a Google Form's link, which is the following:

<https://forms.gle/Qk4zAoUf7fYhHigq6>

The following table is the interpretation of the survey questions that are given to the participants. The interpretation gives insights on whether or not the system is acceptable and successful in terms of design:

Scale	Range Value	Interpretation	Classification
5	4.50-5.00	Highly Acceptable	Successful
4	3.50-4.49	Acceptable	
3	2.50-3.49	Moderately Acceptable	Neutral
2	1.50-2.49	Fairly Acceptable	Unsuccessful
1	1.00-1.49	Not Acceptable	