Team Name - PayNothing

Project/App/Service Name - PayNothing

Video: Demonstration Video

Xiangsheng Gu

GT username: xgu304

xgu304@gatech.edu

Chen Zhang

GT username: czhang708

czhang708@gatech.edu

Zhining Zhang

GT username: zzhang3180

zzhang3180@gatech.edu

Suhas Kumar

GT username: skumar646

skumar646@gatech.edu

PayNothing GitHub link:
https://github.com/XiangshengGu/PayNothing

PayNothingBackend GitHub link:
https://github.com/SuhasKumar04/PayNothingDashboardBackend

Team Agreement

Participation: Team members are expected to work on the project at least 10 hours weekly.

Engagement: Team members are expected to check and respond to group messages in Microsoft Teams within 12 hours.

Short In-person Discussion: Team members are expected to meet after class at the study space in front of the lecture room.

Remote Meeting: Team members are expected to attend online meetings via Zoom Meetings.

Decision-Making: Done by majority votes within the team. Xiangsheng Gu takes the role of tiebreaker.

Roles & Responsibilities:

- **Xiangsheng Gu:** Oversees project progress, leads the core development, arranges meetings, and ensures deadlines are met.
- **Suhas Kumar:** Co-lead team meetings, organize deadlines and team management, oversee back-end development and database operations.
- **Chen Zhang:** Leads front-end development, ensuring an intuitive and user-friendly interface. Implements core UI/UX features, integrates front-end components with the back-end, and collaborates with the team to enhance overall application functionality.
- **Zhining Zhang:** Assist in project coordination, contribute to brainstorming and feature development, and adapt responsibilities based on evolving team needs.

Tools for Collaboration

- **Version Control:** Team members can push updates on codes to the project repository.
- **Communication:** Team members can chat, schedule a time to meet, and share their progress in MS Teams.
- **Project Notebook:** Team members can document together on Overleaf.
- **Demonstrations:** Team members can construct presentations together on Google slides.

Dispute-resolving process: Team members are expected to address issues during remote meetings or in-person discussion. If unresolved, we will seek guidance from instructors.

Problem Overview

Sprint 5:

Existing online platforms for buying and selling used items — including eBay, Facebook Marketplace, OfferUp, and Craigslist — have long dominated the e-commerce space. However, they increasingly frustrate modern users due to several persistent issues:

- **Time-consuming, repetitive listing processes**: Sellers must take high-quality photos (often from multiple angles), upload them carefully, write detailed descriptions, set prices, specify item conditions, and sometimes even handle shipping logistics. This must be repeated for every individual item, making it tedious and overwhelming, especially for casual users simply looking to declutter.
- **Hidden or mandatory fees**: Many platforms add unexpected costs such as seller fees, shipping charges, sales tax, or service fees. These additional costs reduce the perceived value of transactions and discourage user participation.
- **Security and authenticity concerns**: Scams are common, ranging from duplicate or fake listings to sellers who misrepresent items or fail to deliver after payment. Users often encounter stolen images, counterfeit products, or bait-and-switch tactics, making trust a major barrier to successful transactions.

These issues create significant friction, making it difficult for consumers to engage in item exchanges without frustration.

Solution: PayNothing

PayNothing was intentionally designed to remove these barriers and reimagine how people exchange items locally. It introduces a modern, frictionless alternative to traditional resale platforms by focusing on video-first listings, zero-cost exchanges, and trust-centered local connections.

Key innovations include:

- Short-form video listings instead of static photos and long descriptions: Sellers quickly record a short video showcasing their item, making the process faster, more intuitive, and more authentic. Buyers get a real-time, honest view of the item's condition, reducing misunderstandings and scams.
- Zero transaction fees: PayNothing eliminates financial challenges: no seller fees, no shipping costs, and no taxes. Every transaction is direct and cost-free.
- Local, in-person exchanges: The platform prioritizes connecting users in the same city or within driving distance. This local-first model eliminates the need for shipping, reduces the risk of lost or delayed items, and enables immediate, convenient transactions.
- In-app messaging for communication and negotiation: Users can easily chat, ask questions, and arrange safe, public meetups through secure in-app messaging, enhancing transparency and safety.

- Market Need: Whether users are looking to declutter furniture, donate clothing, trade electronics, or give away books, PayNothing offers a zero-cost, low-effort, and high-trust alternative to the outdated models of peer-to-peer commerce.

By solving critical pain points — transaction fees, listing friction, and scam prevention — PayNothing fills a vital gap in the market and redefines how people discover, trade, and give value to everyday items.

By Sprint 4, we had confirmed that the biggest frustrations with legacy marketplaces—tedious multi-step listings, hidden fees, and rampant scams—were only magnified by static photos and disjointed UIs. Our A/B tests showed users loved the swipe-based feed and tag filters we introduced, but still struggled to trust recycled images, locate items nearby, and track conversations once they "liked" a post. Survey data and prototype interviews made it clear: authenticity, immediacy, and seamless messaging were the missing links.

In Sprint 5, we tackled those gaps head-on. We locked uploads to the in-app camera with real-time geotagging—eliminating photo reuse and guaranteeing every video is fresh and local. Our home feed now pushes new posts instantly, no manual refresh/logout required, and auto-matches mutual "likes" into read-receipt–enabled chats. Read/unread indicators and timestamps in the inbox give users clear visibility into every negotiation. Early ad-placement experiments let us validate non-disruptive monetization. Unlike photo-only platforms or fee-laden services, PayNothing's video-first, zero-cost barter model fuses local discovery into a single, trust-centric experience that users helped us refine at every step.

Sprint 4:

Current online platforms that exist for buying and selling used items (e-commerce) such as eBay, Facebook Marketplace, OfferUp, and Craigslist have long dominated the peer-to-peer e-commerce space. However, while these platforms offer access to a wide network of users, they also come with significant drawbacks that increasingly frustrate modern users, such as:

1) Time-consuming and repetitive listing process: Sellers must take high-quality photos (often from multiple angles), upload each image in a specific order, write lengthy and descriptive product details, set prices, specify item conditions, and sometimes even calculate shipping. This entire process must be repeated for every individual item a seller wants to list, making it overwhelming—especially for casual users who are simply trying to declutter their homes or give away a few items.
2) Layers of hidden or mandatory fees: Additional fees are tacked onto customers such as seller fees, shipping fees, sales tax, or service charges. These extra costs reduce incentives for users to participate and often leave both buyers and sellers dissatisfied with the final value they receive from a transaction.
3) Security and authenticity issues: Users frequently encounter scams, including duplicate listings, fake or misrepresented items, stolen product images, and instances of sellers

who accept payment but fail to deliver the product—or deliver something completely different.

These reasons make it difficult for consumers to be able to buy and sell items without facing pain points in the process.

PayNothing was intentionally designed to eliminate these barriers and reinvent the way people engage in local item exchanges. The platform serves as a **local bartering** platform.

Rather than relying on static photos and long descriptions, PayNothing uses **short-form video listing**s for open-to-trade items or giveaways that make the posting process not only faster and more intuitive but also more **authentic**. Users can showcase their items with a simple video, giving potential traders a clearer view of the item's condition in real time.

To avoid extra fees, PayNothing **removes all financial friction** from the exchange. There are no seller fees, no shipping costs, and no platform taxes.

PayNothing emphasizes **local, in-person** trades and focuses on connecting users in the **same city or drivable distance** to avoid logistics complications and eliminate delays or lost packages. By focusing on geographically constrained video feeds and item discovery based on location, users are immediately connected with people nearby, enabling faster, safer, and more convenient exchanges.

To enhance trust and communication, the platform supports **in-app messaging**, so users can chat, negotiate, and ask questions directly about items. Transactions can take place in public locations to ensure safety and prevent fraud.

Whether someone wants to offload extra furniture, exchange books, give away clothes, or trade electronics, PayNothing provides a **zero-cost, low-effort, high-trust** alternative to existing platforms. By solving key issues like transaction fees, scam prevention, and user effort, PayNothing fills a critical gap in the current market and redefines how people declutter, donate, and discover value in their everyday items.

<u>Sprint 3:</u>

Popular e-commerce platforms like eBay, Facebook Marketplace, OfferUp, and Craigslist allow users to buy and sell used items online. However, in these platforms:

1. Users are required to take photos, upload them sequentially, write detailed descriptions, and repeat this tedious process for each item.
2. After this time-consuming struggle, users often need to pay extra fees for their items, such as shipping fees, seller fees, and sales tax.

3. Disappointing experiences happen when photos are reused to create duplicate listings, fake items with images of real items, non-delivery of shipped items or delivery of wrong items.

PayNothing eliminates these issues by serving as a **local bartering** platform where users **record and post short videos** of open-to-trade items or giveaways **in** the app --- keeping the process simple and authentic. To avoid extra fees, it focuses on connecting users in **the same city or drivable distance**. Users can chat directly about items, and transactions take place in **public locations** to ensure safety and prevent fraud.

Sprint 1:

There exists popular applications that offer selling services to customers, such as customers selling used items to others. Popular applications include Ebay, Facebook Marketplace, and OfferUp that require users to take pictures of their item and write a full description. However, customers/sellers are oftentimes subject to pay many fees such as shipping fees and seller fees. Our app is designed to avoid all these issues, operating as a local bartering platform where users can post short videos of items they are open to trade or simply give away to others interested. To avoid shipping fees, our app will be geared to local communities within drivable ranges. In addition, users will have the ability to have conversations with other users related to items. Transactions will occur in public settings to ensure customer safety and fraud prevention.

Analysis of Survey Responses

Sprint 3:

According to responses from our conducted survey[1]participants appreciated the affordability of buying used items and some of them reported positive experiences with sellers being fair and honest about item conditions, but only a few of them found the process easy and had no complaints. Besides that, according to responses to the last survey question “Could you briefly share some thoughts or concerns on your experience of buying/selling/trading used items?”, concerns about the quality and authenticity of items were common and some of the participants suggested features like seller reviews and verification to improve trust. Moreover, concerns about meeting buyers or sellers in certain areas were mentioned, highlighting the need for safe and convenient transaction locations. Many participants called for a more streamlined and less tedious process for selling items, including reducing the number of steps and eliminating unnecessary fees.

[1] Please view Appendix 12.2: Responses on the Survey Questions

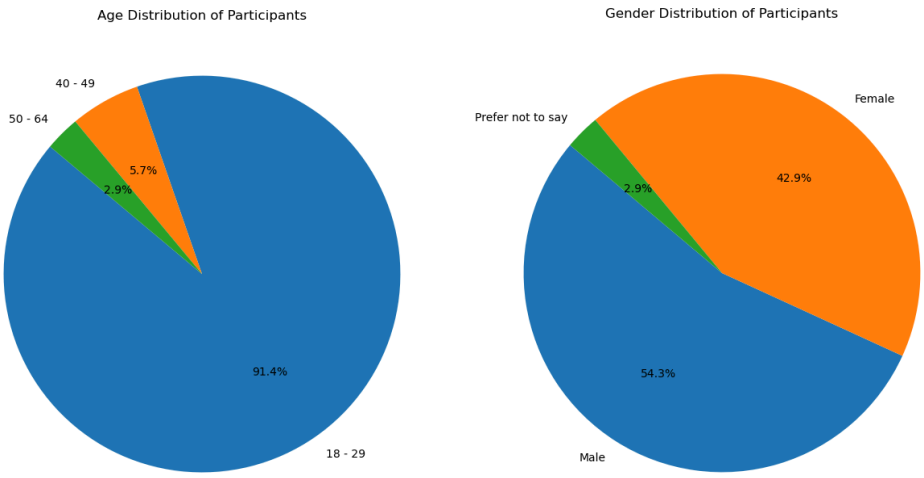


Figure 1: Demographic Distribution of Participants

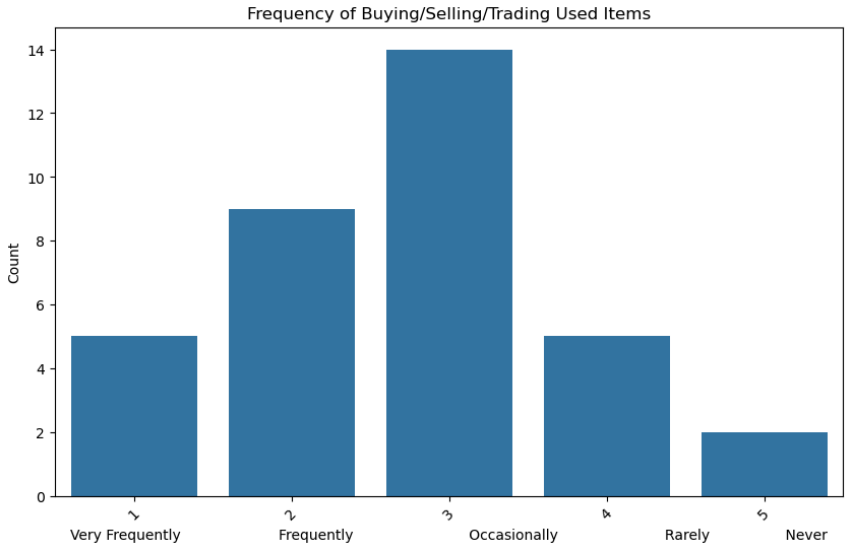


Figure 2: Many participants either occasionally or frequently buy, sell, and trade used items

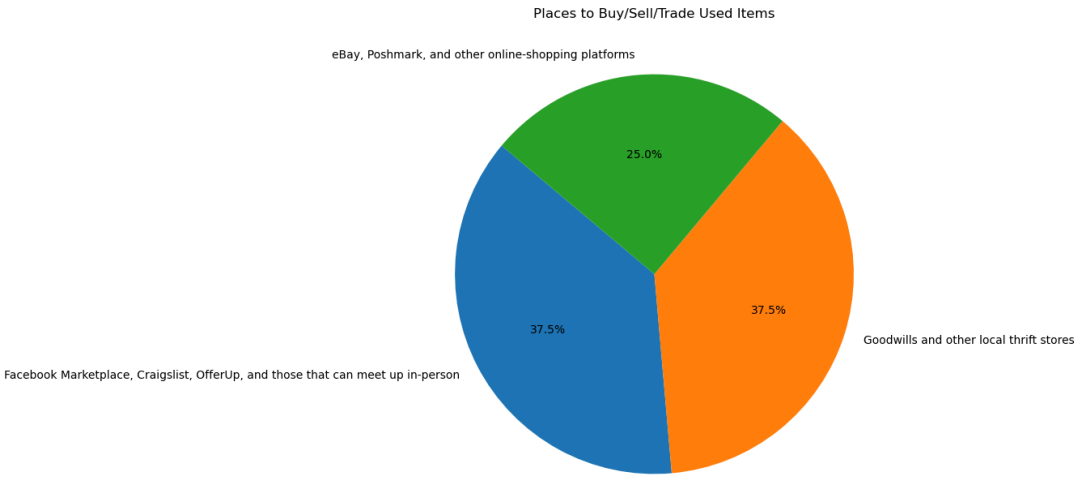


Figure 3: More than one third of participants found platforms that allow meeting up in-person are more reliable compared to online-shopping-only platforms

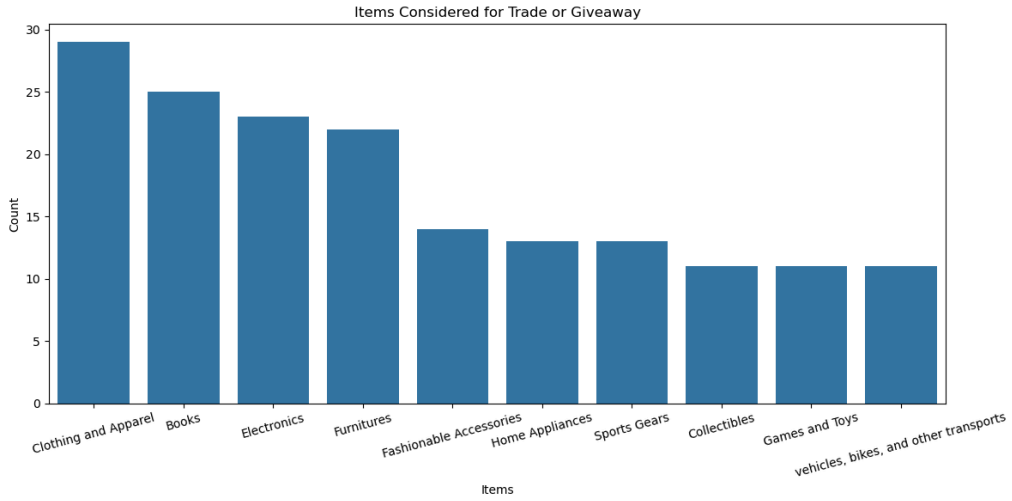


Figure 4: Clothing & Apparel, Books, Electronics, Furniture are the most popular used items that participants buy, sell, and trade online

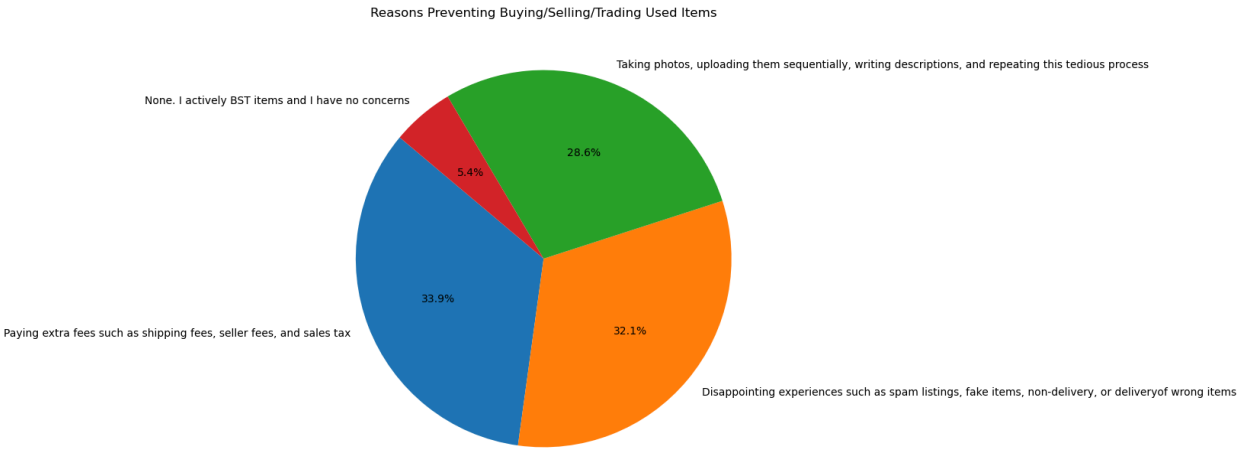


Figure 5: Many participants do not want to sell or trade used items not only because of additional fees, such as shipping fees, seller fees, and sales tax, but also the effort required to sell items, such as taking photos, uploading, and writing description

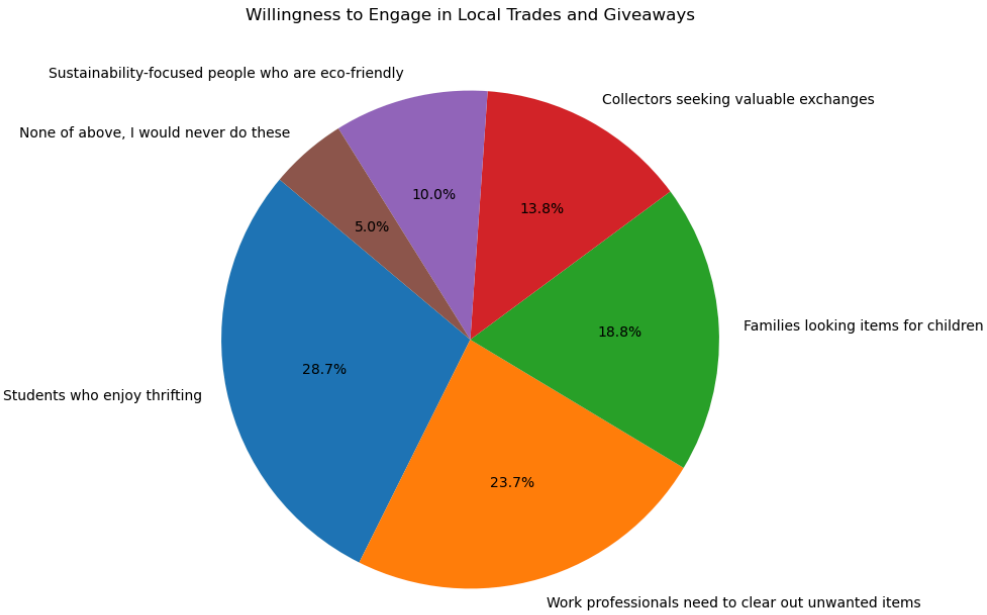


Figure 6: Primary users who engaging in buying, selling, and trading used items are students, work professionals, and families

Domain Research

After conducting local interviews[2], several key pain points emerged:

- People feel that selling items online involves too much effort, and handling inquiries can be time-consuming.
- Some are discouraged by seller fees, shipping costs, and long waiting times for delivery.
- They also shared concerns about scams, counterfeit items, and personal safety.
- Some also mentioned the importance of ensuring fairness and quality control to make bartering more reliable and trustworthy.

As a result, users expressed a desire for an easier, more interactive way to give away or exchange items without the hassle of creating lengthy posts or dealing with hidden fees.

Regarding academic papers, there are a multitude of sources that highlight the importance of a bartering style app in order to improve sustainability and community participation in this type of exchange. According to “E-barter Exchanging System: Toward a Smart and Sustainable Community”[3], a bartering system aims to provide public benefits, reduce environmental waste, and increase social cohesion. This shows a clear indication that a bartering app not only has a place in the market, but puts forth social benefits impacting the community positively.

Sprint 1:

Within the domain of selling used items online, there already exist many applications to service this field, such as Ebay, Facebook Marketplace, Depop, Craigslist, and the list goes on. However, what do all of these applications have in common that users have an issue with? After interviewing on the topic, many were frustrated with the numerous miscellaneous fees associated with these apps, as these apps are aimed to generate revenue for the company. Many times, ads are also filled within these applications, also deemed "annoying" by many users as it interferes with UI design and takes away from the true purpose of the application. Our app is completely free, ad free, no shipping fees, and no strings attached. These solutions address the research conducted within the domain research, trying to appease the users as much as possible.

Competitive Analysis

Competitors have existed for many years, gaining traction, posting advertisements, and letting their services become known throughout the community. However, none of them is adapting

[2] Please view Appendix 12.3: The First Interview

[3] Published in: Intelligent Sustainable Systems, Lecture Notes in Networks and Systems, 2023, pp. 355–371 DOI: 10.1007/978-981-19-7660-5_31

video-based item listings, which show the item's true condition. These features we believe are a promising alternative to static photos and textual descriptions. As trading items on our app require no additional costs and it ensures local transaction safety, we target our app at the gap within the competitive industry.

Sprint 2:

Feature	**PayNothing**	**eBay**	**Facebook Marketplace**	**OfferUp**	**Craigslist**
Listing Type	Video-based	Image/Text-based	Image/Text-based	Image/Text-based	Image/Text-based
Item Types	Used, Tradeable, Giveaways	New, Used	New, Used	New, Used	New, Used
Primary Model	Bartering and Free Giveaways	Bidding, Fixed-Price Sales	Buy/Sell	Buy/Sell	Buy/Sell
Local Transactions	Yes	Yes	Yes	Yes	Yes
Shipping Available	No	Yes	Yes	Yes	No
Transaction Fees	None	12-15% Seller Fees	None	Service Fee for Shipping	None
Scam Prevention	Video-based verification	Buyer/Seller Ratings	Community Reports	Buyer/Seller Ratings	None
User Verification	Email, Phone Verification	Verified eBay Accounts	Facebook Profiles	TruYou Identity Verification	None
Privacy Concerns	No Personal Data in Listings	Shipping Addresses Required	Facebook Profile Visibility	Profile-Based Information	Email-Based Contact
Ease of Use	High (One-tap video posting)	Moderate (Detailed listings required)	High (Integrated with Facebook)	Moderate (App-based listings)	Low (Manual posting required)
Buyer/Seller Communication	In-app Messaging	eBay Messages	Facebook Messenger	In-app Chat	Email or Phone

Fraud and Scam Risks	Low (Authenticity via Video)	High (Fake Listings, Payment Scams)	Medium (Fake Profiles)	Medium (Fake Listings)	High (No Verification or Moderation)
Ads and Promotions	No Ads, Community-Based Leaderboard	Paid Promotions	Paid Promotions	Paid Promotions	No Paid Promotions
Main Drawback	Limited by local supply and demand	High seller fees	Privacy concerns due to public profiles	Limited user base	No user protections

Table 1: Competitive Analysis: PayNothing vs. Existing Marketplaces

Sprint 1:

Competitions already exist in this domain, as mentioned before, popular applications such as Ebay already have a customer base with millions of users. Depop, for example, is geared to only clothes, but has the same principle as Ebay, selling used clothing/shoes/jewelry, etc. All of these competitors have big customer bases, but also have existed for many years, gaining traction, posting advertisements, and letting their services become known throughout the community. However, as our app is to be geared to be completely free, offering a bartering feature, ensuring transaction safety, and within drivable distance. This allows our app to target the gap within the competitive industry, addressing customer concerns and servicing a useful process within communities.

Solution Approaches

Sprint 5:

In Sprint 5, we focused on continuous improvement, building on user feedback from A/B testing conducted during Sprint 4. On the UI side, we enhanced the inbox page by adding conversation status details, including timestamps and read/unread indicators, to make communication more seamless. For publishing and collection records, we also optimized the UI by displaying cover images, titles, descriptions, geographic locations, and comprehensive video interactions after clicking. We also enabled location services APIs, allowing users to display the item's location when creating a post, making it easier for others to assess proximity during discovery. On the functionality side, the app now supports real-time updates — when a user uploads a new post, it

instantly appears for all users without requiring them to refresh or log out and back in. This creates a more dynamic and interactive experience. Additionally, the matching algorithm is now fully operational: if two users like each other's posts, they are immediately connected via the inbox page, enabling direct communication and faster exchanges.

Solution Approach 1: Posting Item and Video Feeds (selected)

Approach:

- Users access the "Post" screen to **record short-form videos *directly within the app*** showcasing items they want to barter or give away. To ensure authenticity, **only videos captured through the in-app camera with geolocation data enabled** can be uploaded. External uploads are not allowed, preventing tampering or reused content.
- The app curates these verified video posts into personalized feeds on the "Home" screen based on user location, search tags, and activity history. Users can then initiate one-on-one conversations using the message icon on the "Home" screen, with ongoing chats managed in the "Inbox" screen.

Pros:

- **Authenticity by design**: Mandatory in-app video recording and geotagging ensures listings are recent, local, and unaltered, helping eliminate fake or recycled content.
- **Dynamic, trustworthy presentation**: Videos allow real-time demonstration of item condition, background context, and usability—providing a richer, more reliable user experience than static images.
- **Encourages local engagement**: Users are shown geographically relevant posts to promote hyperlocal exchanges, minimizing reliance on shipping and third-party logistics.
- **Cost-effective for users**: No seller fees, platform commissions, or delivery charges; everything is negotiated and exchanged locally.
- **Streamlined chat and negotiation**: In-app messaging reduces spam, increases privacy, and builds trust between parties.

Cons:

- Video data requires more cloud storage, processing power, bandwidth, and costs to our Startup. May introduce scalability issues and increase operational costs.
- Contents are geographically constrained by locations and users cannot find items in other cities.
- Users may be hesitant to share videos with real voices or backgrounds. Reliance on location data can raise security questions if not transparently managed.

How can it be differentiated from existing solutions?

- Unlike platforms such as eBay, Facebook Marketplace, or OfferUp, which primarily rely on uploaded photos, our approach mandates *real-time, in-app video capture with*

location tagging. This eliminates issues like photo reuse scams, misleading listings, and duplicate posts. The system emphasizes **verified presence and location-based authenticity**—ensuring that what users see is what they get, and that items are actually available nearby.

Use Cases:

1. **Givers:**
    a. **Scenario description:** Tom has two pairs of used sneakers in good condition and some clothes he doesn't wear anymore. He wants to give away clothes to people who truly need them, and two pairs of shoes to trade for something he might be interested in.
    b. **Steps:**
        i. Registers via the “Auth” screen and sets up location services.
        ii. Navigates to the “Post” screen and **records a video using the in-app camera**, showcasing each item.
        iii. Adds title, description, tags (e.g., “sneakers,” “size 10”), and **location** then posts it.
        iv. The system shares his post with users in the same city.
        v. Tom starts browsing videos on the home page, and swipes right to mark the videos he likes.
        vi. Once matched, he receives messages in the “Inbox” screen, negotiates with interested users.
        vii. They agree to meet at a public location and complete the exchange.
2. **Systems:**
    a. **Scenario description:** The app system organizes and moderates video posts, ensuring content quality and safety for users. It prioritizes local posts in the feed based on user location.
    b. **Steps:**
        i. Collects and verifies each uploaded video’s timestamp and location metadata.
        ii. Blocks uploads from non-app sources to enforce authenticity.
        iii. Collected location data will be used to check if both the giver and receiver are in the same city.
        iv. Item search history from users can be used to personalize the video feed.
3. **Receivers:**
    a. **Scenario description:** Jerry wants to trade two vintage shirts for shoes but wishes to avoid fake or irrelevant listings.
    b. **Steps:**
        i. He selects “shoes” tag, searches, “size 10”, and other keywords to find relevant posts in the “Home” screen.
        ii. The system sorts the most relevant posts based on keywords, and he views the results.
        iii. Views **video posts confirmed to be recent and local**, not old or reused.

iv. He swipes right to mark the videos he likes, hoping to be matched with another user.
v. Once Jerry matches with another user, they will communicate in the inbox.
vi. The system crosschecks user locations with, and informs safety warnings to each user in the “Inbox” screen.
vii. Jerry acknowledges the alert and negotiates the meetup time and location with the other user.
viii. He arrives at the agreed-upon public location and safely completes the exchange.

Sprint 4:

In Sprint 4, our team continued refining and iterating on the previously selected solution direction, with a primary focus on solution approach 1. We leveraged insights from our initial prototype, including user feedback and survey results, and we concentrated on improving user interface aesthetics, interaction flow, and feature clarity in different functions. This sprint emphasized the polish and performance of the video posting and content discovery experiences. Also, we compared our improved prototype with the initial prototype through A/B testing to validate our enhancements. We used comprehensive questionnaires to collect user feedback to determine whether the changes we made in Sprint 4 were effective and whether they needed to be further improved in subsequent Sprints.

Solution Approach 1: Posting Item and Video Feeds (selected)

Approach:

- Users navigate to the “Post” screen to record and upload short-form videos showcasing their items for bartering or giveaway purposes. The system curates these video posts into personalized feeds on the “Home” screen based on the user’s location, search preferences, and interaction history.
- Users can communicate with the poster in the "Chat" screen through the letter icon on the "Home" screen. And all chat sessions are displayed in the "Inbox" screen for users to manage conversations.

Pros:

- In-App video posts provide a more authentic and transparent way of showcasing real-time product conditions compared to static images, which discourage fake or reused listings.
- Dynamic video content is more engaging and encourages trust and quicker decisions.
- Users are encouraged to connect through in-app conversations to clarify item information and follow-up details.

- Encourages more local trades and giveaways in drivable distance as users accessing video feeds, removing reliance on shipping or third-party logistics.
- Users avoid common marketplace costs such as seller fee, sale tax, or delivery charges.

Cons:

- Video data requires more cloud storage, processing power, bandwidth, and costs to our Startup. May introduce scalability issues and increase operational costs.
- Contents are geographically constrained by locations and users cannot find items in other cities.
- Users may be hesitant to share videos with real voices or backgrounds. Reliance on location data can raise security questions if not transparently managed.

How can it be differentiated from existing solutions?

- Unlike eBay, Facebook Marketplace, or OfferUp, this approach emphasizes live presentation of an item through videos rather than static photos, which eliminates photo reuse scams, fake listings, and duplicate posts. Use video feeds to attract user participation through dynamic videos and interactive animations.

Use Cases:

4. **Givers:**
    a. **Scenario description:** Tom has two pairs of used sneakers in good condition and some clothes he doesn't wear anymore. He wants to give away clothes to people who truly need them, and two pairs of shoes to trade for something he might be interested in.
    b. **Steps:**
        i. He creates his user profile in the “Auth” screen with location setup, and navigates to the “Post” screen of the app.
        ii. He records a short video to showcase, introduces all his items, adds title, description with tags, and then posts it to the “Home” screen.
        iii. Alternatively, he can record videos for each item and post them separately to the “Home” screen.
        iv. System notified users in the same city to view this new post.
        v. On the “Inbox” screen, Tom receives messages from users and negotiates the meetup time and location.
        vi. He arrives at the agreed-upon public location and safely completes the exchange.
5. **Systems:**
    a. **Scenario description:** The app system organizes and moderates video posts, ensuring content quality and safety for users. It prioritizes local posts in the feed based on user location.
    b. **Steps:**
        i. System collects user video data, location data, and other activities.
        ii. Collected video data will be viewable for other users from the same city.

iii. Collected location data will be used to check if both the giver and receiver are in the same city.
iv. Item search history from users can be used to personalize the video feed.

6. **Receivers:**
    a. **Scenario description:** Jerry is a newly-enrolled user who recently posted two vintage t-shirts. He wants to trade them for a pair of shoes, but also wants to avoid spam posts and rude users.
    b. **Steps:**
        i. He selects “shoes” tag, searches, “size 10”, and other keywords to find relevant posts in the “Home” screen.
        ii. The system sorts the most relevant posts based on keywords, and he views the results.
        iii. He taps the post to message users who posted shoes he wants to trade with.
        iv. The system crosschecks user locations with, and informs safety warnings to each user in the “Inbox” screen.
        v. Jerry acknowledges the alert and negotiates the meetup time and location with the other user.
        vi. He arrives at the agreed-upon public location and safely completes the exchange.

Sprint 3:

For Sprint 3, we continue to use the previous three solutions of our application. According to the previous survey, we chose the 1st solution (Posting Item and Video Feeds) to implement for codes. Also, we further surveyed the other solutions with users of our first prototype to figure out whether we need to achieve these functions in subsequent sprints.

Sprint 2:

For Sprint 2, we proposed three solutions (functions) to further improve the functions of our application. Considering the core degree of the functions, we finally chose method 1 and method 2 for development.

Solution Approach 1: Posting Item and Video Feeds (selected)

Approach:

- Users navigate to the “Post” screen to record and upload short videos of their items for bartering or giveaway purposes. The system curates these video posts into personalized feeds on the “Home” screen based on user location and previously liked items.

Pros:

- In-App video posts provide a more authentic and transparent way of showcasing items compared to static images.
- Encourages more local trades and giveaways in drivable distance as users accessing video feeds.
- No shipping fee, seller fee, sale tax, or wait on delivery.
- Enhances user engagement with a dynamic video-based feed.

Cons:

- Video data requires more storage, bandwidth, and costs to our Startup.
- Contents are limited by locations and users cannot find items in other cities.
- Privacy concerns related to user location services and recordings of actual voice and surroundings.

How can it be differentiated from existing solutions?

- Unlike eBay, Facebook Marketplace, or OfferUp, this approach emphasizes live presentation of an item through videos rather than static photos, which eliminates photo reuse scams, fake listings, and duplicate posts.

Use Cases:

7. **Givers:**
    a. **Scenario description:** Tom has two pairs of used sneakers in good condition and some clothes he doesn't wear anymore. He wants to give away clothes to people who truly need them, and two pairs of shoes to trade for something he might be interested in.
    b. **Steps:**
        i. He creates his user profile with location setup, and navigates to the “Post” screen of the app.
        ii. He records a short video to showcase and introduce all his items, then posts it to the “Home” screen.
        iii. Alternatively, he can record videos for each item and post them separately to the “Home” screen.
        iv. System notified users in the same city to view this new post.
        v. On the “Inbox” screen, Tom receives messages from users and negotiates the meetup time and location.
        vi. He arrives at the agreed-upon public location and safely completes the exchange.
8. **Systems:**
    a. **Scenario description:** The app system organizes and moderates video posts, ensuring content quality and safety for users. It prioritizes local posts in the feed based on user location.
    b. **Steps:**

i. System collects user video data, location data, and other activities.
ii. Collected video data will be viewable for other users from the same city.
iii. Collected location data will be used to check if both the giver and receiver are in the same city.
iv. Item search history from users can be used to personalize the video feed.

9. **Receivers:**
    a. **Scenario description:** Jerry is a newly-enrolled user who recently posted two vintage t-shirts. He wants to trade them for a pair of shoes, but also wants to avoid spam posts and rude users.
    b. **Steps:**
        i. He searches “shoes”, “size 10”, and other keywords to find relevant posts in the app.
        ii. The system sorts the most relevant posts based on keywords, and he views the results.
        iii. He taps the post to message users who posted shoes he wants to trade with.
        iv. The system crosschecks user locations with, and informs safety warnings to each user in the “Inbox” screen.
        v. Jerry acknowledges the alert and negotiates the meetup time and location with the other user.
        vi. He arrives at the agreed-upon public location and safely completes the exchange.

Solution Approach 2: Gamification and Reward System with LoopCoin

Approach:

- To enhance user engagement and reward active participation, we introduce LoopCoin, a platform-specific cryptocurrency. Users earn LoopCoins by keeping the app active in the foreground for at least 1 hour or completing successful trades confirmed by both parties with detailed reviews. LoopCoins can be spent on various perks, such as bumping posts to the top of the latest feed or instantly revealing pickup locations for giveaway items.

Pros:

- Encourages consistent app usage and active participation.
- Adds real value and utility through the LoopCoin economy.
- Creates an incentive-driven community with positive interactions.

Cons:

- Users may attempt to game the system for excessive LoopCoin accumulation.
- Requires secure and scalable blockchain integration for crypto management.

How can it be differentiated from existing solutions?

- Unlike traditional local bartering platforms, PayNothing gamifies the experience through LoopCoin incentives, creating a thriving virtual economy. Current platforms rely on organic engagement without providing clear motivation to participate actively. Our cryptocurrency-driven approach differentiates PayNothing from platforms like Craigslist and Facebook Marketplace by blending blockchain technology with real-life community interactions.

Use Cases:

1. **Active Trader Using LoopCoins:**
    a. **Scenario description:** Emily is an active user who frequently trades items. She often keeps the PayNothing app open for an hour while browsing posts and messaging others. She also uses the LoopCoins she earned to promote her new video posts.
    b. **Steps:**
        i. Keep the app open for at least 1 hour to earn 1 LoopCoin.
        ii. Post an item and complete a successful trade with positive feedback.
        iii. Earn an additional LoopCoin for the successful transaction.
        iv. Spend 1 LoopCoin to bump her new item video to the top of the latest video feeds.
2. **New User Incentivized by LoopCoin:**
    a. **Scenario description:** Joshua is a new user hesitant to participate until he learns that he can earn LoopCoins for verifying identity and post his first video. Motivated by the opportunity to earn and spend these coins, he lists an item and completes his first transaction.
    b. **Steps:**
        i. He creates his user account and explores the platform.
        ii. He discovers that verifying identity and posting the first item video can earn 10 LoopCoins.
        iii. He then posts an item video for trade or giveaway and finds a match.
        iv. Successfully completed his first transaction and earned 10 LoopCoins.
        v. He learns more about spending the LoopCoins to bump his future posts for greater visibility.
3. **First Come First Serve:**
    a. **Scenario description:** Sarah is interested in furniture listed on PayNothing. Instead of waiting for the video poster to respond to her with pickup location details, she spends 1 LoopCoin to reveal the pickup location immediately and go pick it up right away.
    b. **Steps:**
        i. She browses videos on the “Home” screen of the app and finds the furniture she wants.
        ii. This furniture is still available as its hidden pickup location has not been revealed by others.
        iii. She decides to spend 1 LoopCoin to reveal the location instantly to her, while the post is marked as “Location Revealed”.

iv. She drives to pick up the furniture and the transaction is completed.

Solution Approach 3: Community-based Leaderboards (Needs and Haves)

Approach:

- Because we focus on bartering methods, it is important to know the current demand and supply of the nearby community. Therefore, users should add tags when posting so that we can count the supply. By counting the search content on the main interface, we can understand the needs of users. Then, through the ranking, the community will know what items are currently needed or available.

Pros:

- Improve visibility by showing the most needed and readily available items.
- Increase matching efficiency by showing high-demand.
- Encourages community participation.

Cons:

- Need users to tag correctly to ensure data accuracy.
- Maintain an active user base to enhance the effectiveness of the leaderboard.

How can it be differentiated from existing solutions?

- Craigslist or HaveNeed do not provide a leaderboard feature and instead rely on keyword-based search to display posts. Our approach introduces a ranking system that highlights the most needed and available items, making it easier for users to identify trade opportunities.

Use Cases:

1. **Poster:**
    a. **Scenario description:** Mike has extra PlayStation games at home, and he hopes to exchange them for other games through the community's barter platform. Therefore, when he posts, he needs to fill in the items he can provide (Haves) and the items he wants to get (Needs).
    b. **Steps:**
        i. Log in to the app and enter the "Post" interface.
        ii. Enter the information and video of the items that can be provided.
        iii. Add appropriate item tags (such as: #PlayStation #Game).
        iv. Enter the type of items you want to exchange (such as: Games).
        v. Post the post and wait for responses from community members.
        vi. Receive private messages from interested users, negotiate exchange details, and complete the transaction.
2. **Modify Published Post:**

a. **Scenario description:** Jeff originally posted a thread hoping to trade a printer card for a gift card. However, after checking the community leaderboard, he saw that gift cards were in low supply, while PlayStation games were in ample supply. Therefore, he decided to amend his thread.
b. **Steps:**
    i. Log into the app. Check the leaderboard and find that gift cards have scarcity but PlayStation games are easier to get.
    ii. Go to the "My Posts" interface.
    iii. Find the previously posted post
    iv. Edit the needs (Needs): change from "Gift Card" to "PlayStation Game"
    v. Update the tag (such as: \#PlayStation Game).
    vi. Save the changes and repost the post, waiting for the community members to respond.

3. **Potential Participants:**
    a. **Scenario description:** Murry usually does not actively publish demand or supply, but she likes to check the leaderboard to understand the demand and supply situation in the community. One day, she found that the leaderboard showed "High demand item: NIP haircare", and she happened to have an extra one at home.
    b. **Steps:**
        i. Open the app and enter the "Ranking" interface to view the current popular needs and haves in the community.
        ii. Look through videos and find a video that matches the items you have (e.g., #haircare demand is high).
        iii. Click the "trade button" to create a thread in the "inbox screen". And communicate with the poster of the video to finish the transaction.

Sprint 1:

Solution Approach: AI-Powered Video Feeds and Item Matching

Approach:

- Use AI models/algorithms to protect users from scams and scammers' video posts. To avoid hit-and-miss situations, feed users with personalized video posts based on their preferences or previous posted contents.

Pros:

- Reduces spam and fraudulent posts, increasing user trust.
- Enhances engagement by showing relevant items to users.
- Encourages more trades and giveaways through personalized feeds.

Cons:

- Requires continuous model training to remain accurate and efficient.
- Potential bias in recommendations if AI models are not properly optimized.
- Privacy concerns related to user behavior tracking.

How can it be differentiated from existing solutions?

- Unlike Offerup and Craigslist, this approach should significantly eliminate scammers' video posts. Most popular platforms did not integrate AI-driven technology for local bartering, instead they are doing commercial selling.

Use Cases:

1. **Givers:**
    a. **Scenario description:** Tom has two pairs of used sneakers but in great condition. He wants to give away one pair to people who truly need it, and another pair to trade for something he might be interested in.
    b. **Steps:**
        i. He records and uploads a video showcasing his sneakers.
        ii. AI classified the posted items to the 'Shoes' category.
        iii. System notify users who recently liked posts in that category to view this new post.
        iv. AI analyzed Tom's recently liked videos and crosscheck users who posted these items with users who recently liked posts in the 'Shoes' category.
        v. Items appear in the feeds of targeted users, increasing the likelihood of a successful giveaway or exchange.
2. **Intelligent systems:**
    a. **Scenario description:** The AI system automatically filters and ranks video posts based on relevance and user trust scores.
    b. Steps:
        i. AI detects and flags suspicious or spam posts based on content analysis and user reports.
        ii. Posts from verified and frequently engaged users receive higher rankings.
        iii. Personalized recommendations adjust dynamically based on user engagement.
        iv. Users can provide feedback on recommendations, further refining AI learning.
3. **Receivers:**
    a. **Scenario description:** Jerry recently posted two vintage t-shirts. He wants to trade them from a pair of shoes but wants to avoid spam posts and rude users.
    b. **Steps:**
        i. He can open the app and search 'shoes', 'size 10', and other keywords to find relevant posts.
        ii. AI sorts the most relevant posts based on keywords, likes, and accounts activity.

iii. He can tap the post to message users who posted shoes he wants to trade with.
iv. AI crosschecks video post locations with user actual locations, and informs transaction histories to each user in the 'Inbox' screen.
v. Jerry acknowledges the alert and will make a decision whether to continue chatting or block the user.

Sprint 1:

Solution Approach: Gamification and Reward System

Approach:

- To drive engagement, users can earn points for completing transactions, responding quickly, and maintaining a positive transaction history. Points can be redeemed for perks such as priority listings, exclusive trade opportunities, or digital badges.

Pros:

- Encourages participation and community involvement.
- Reduces inactive user base by incentivizing engagement.
- Adds a fun, competitive aspect to the platform.

Cons:

- Could lead to users exploiting the system solely for points.
- Requires a well-designed reward structure to make it effective.

How can it be differentiated from existing solutions?

- Unlike existing barter platforms like Craigslist or Facebook marketplace, which do not provide incentives beyond the transaction itself, our system introduces an interactive, game-like experience. Current platforms rely on organic engagement without providing clear motivation to participate actively. With our gamified approach, users can feel a sense of accomplishment and progression, which encourages them to continue using the platform.

Use Cases:

1. **Active Participant Earning Rewards:**
    a. **Scenario description:** Emily regularly trades items and wants to maximize her benefits. By completing transactions and maintaining positive interactions, she earns reward points that grant her access to premium features.
    b. **Steps:**
        i. Log into the app and browse available listings.
        ii. Initiate and complete a barter transaction.

iii. Receive positive feedback from the trade partner.
iv. Earn points that can be redeemed for priority listing placement.
v. Continue engaging on the platform to unlock additional perks.

2. **New User Incentivized to Participate:**
    a. **Scenario description:** Joshua is new to the app and hesitant to engage. He notices that first-time trades earn a significant bonus, motivating him to post his first listing.
    b. **Steps:**
        i. Create an account and explore the platform.
        ii. Discover the “First Trade Bonus” promotion.
        iii. List an item for trade and find a match.
        iv. Successfully complete his first transaction and earn bonus points.
        v. Use the points to enhance his visibility on the platform.
3. **Leaderboard Competitor:**
    a. **Scenario description:** Andy enjoys competition and wants to be a top-ranked user. He actively engages with the community to climb the leaderboard and unlock exclusive rewards.
    b. **Steps:**
        i. Check the leaderboard rankings on the app.
        ii. Engage in more trades and respond quickly to inquiries.
        iii. Earn additional points for consistent transactions and positive feedback.
        iv. See his ranking improve on the leaderboard.
        v. Redeem exclusive perks, such as featured trader status.

Sprint 2:

Solution Approach: Contactless Package Locker and Delivery Integration

Approach:

- Integrate our platform with contactless package lockers and delivery systems located in residential and commercial areas. This allows users to complete trades and giveaways contactlessly for convenience and safety. Once users agree on an exchange through Inbox in PayNothing app, they can drop off and collect items at nearby lockers using secure access codes provided by the system. It requires geo-location data to suggest nearby package lockers and collaborating with third-party locker providers (such as Luxer One and Amazon). Notifications are sent to users with pickup/drop-off instructions.

Pros:

- Enhances user safety by avoiding direct face-to-face exchanges.
- Increases convenience, allowing trades to happen at flexible times.
- Reduces the risk of failed transactions or missed meetings.

Cons:

- Requires partnerships with locker providers.
- Limited availability in rural or less populated areas.
- Potential logistical challenges during high-usage periods.

How can it be differentiated from existing solutions?

- Unlike traditional marketplaces, this approach provides contactless and secure item exchanges through modern infrastructure, increasing user trust and flexibility. No other major bartering platforms offer locker-based exchange services.

Learning Prototypes

Sprint 5:

In Sprint 5, our team continued to refine the user experience by addressing gaps identified through user feedback and A/B testing in Sprint 4. While the prior sprint focused heavily on UI polish and interaction mechanics (such as swipe gestures and authentication flow), Sprint 5 transitioned toward **enhancing trust, immediacy, and location-driven discovery**.

One of the most significant updates in this sprint was the **mandatory use of the in-app camera and location services when creating a post**. To combat fake listings and ensure content authenticity, users can now only upload videos if they are recorded directly within the app and paired with real-time geolocation data. This ensures that all item listings are both recent and local, helping to establish a strong baseline of trust across the platform. We are no longer allowing users to upload posts without including their location information, and the design decision rooted in direct user feedback favoring transparency and safety.

On the UI side, we expanded our inbox screen capabilities by integrating **read/unread indicators and timestamped messages**, offering a clearer overview of ongoing conversations. We also changed the display style of the user's previous posts and favorites in the profile screen. Now we will display the cover image, title, description and location information for each record. When users click on a record, they can browse the corresponding video.These small visual touches significantly improved user perception of app professionalism and usability.

From a system functionality perspective, we implemented **real-time synchronization** of posts. When a user uploads a new item, it now instantly appears in nearby users' feeds without requiring manual refresh or re-login.We have also implemented real-time updates of publishing records and favorite records on the profile screen, enhancing fluidity and interactivity. In parallel, our **matching algorithm is now fully operational**: if two users "like" each other's video posts, they are automatically connected in the inbox screen, enabling direct communication to accelerate item exchanges.

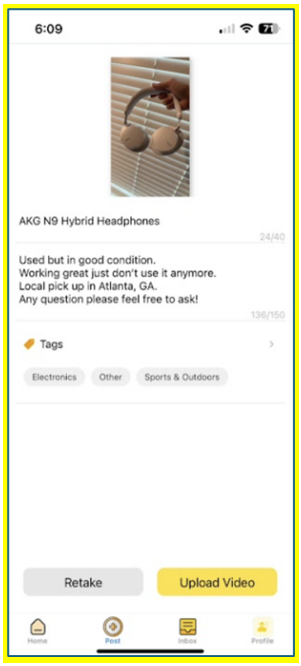
In addition, we introduced early-stage ad format testing in preparation for future monetization. The inbox screen now supports both banner ads and full-screen interstitial ads, enabling us to analyze user interaction and tolerance toward different ad types while maintaining a clean experience.

User Testing (A/B Testing Design and Observations)

To validate these improvements, we conducted a new round of structured A/B testing. Each user interacted with both the previous prototype and the new Sprint 5 version to compare efficiency, trust-building elements, and overall experience:

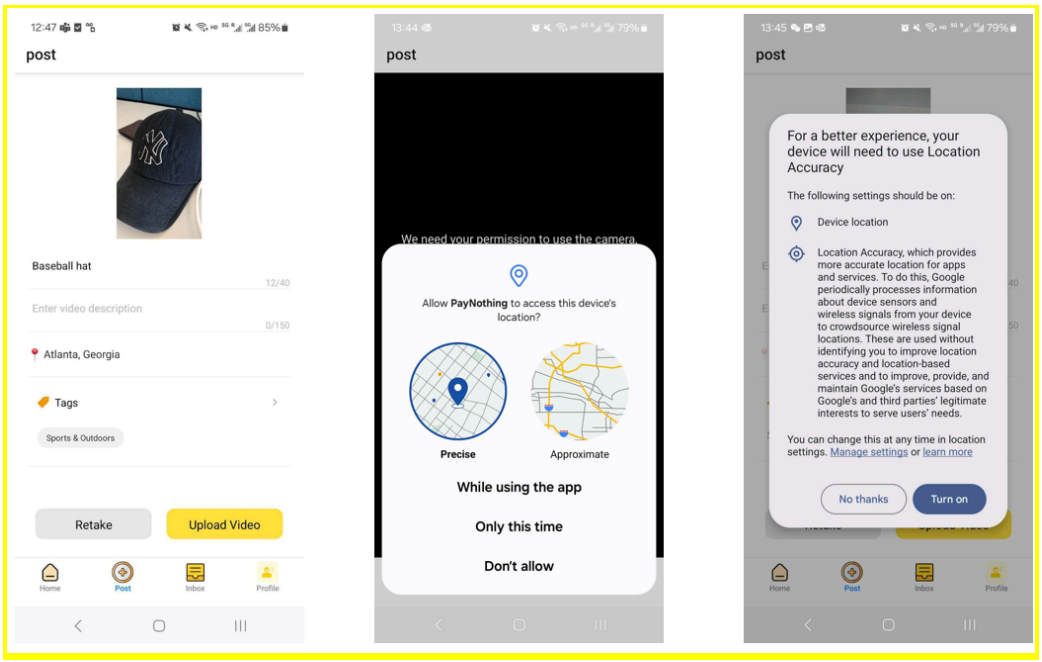
1. **How long does it take to post a video in 'post screen'?**
   → *Sprint 5 emphasizes speed and verification, tracking upload time with auto-location capture.*

   A:



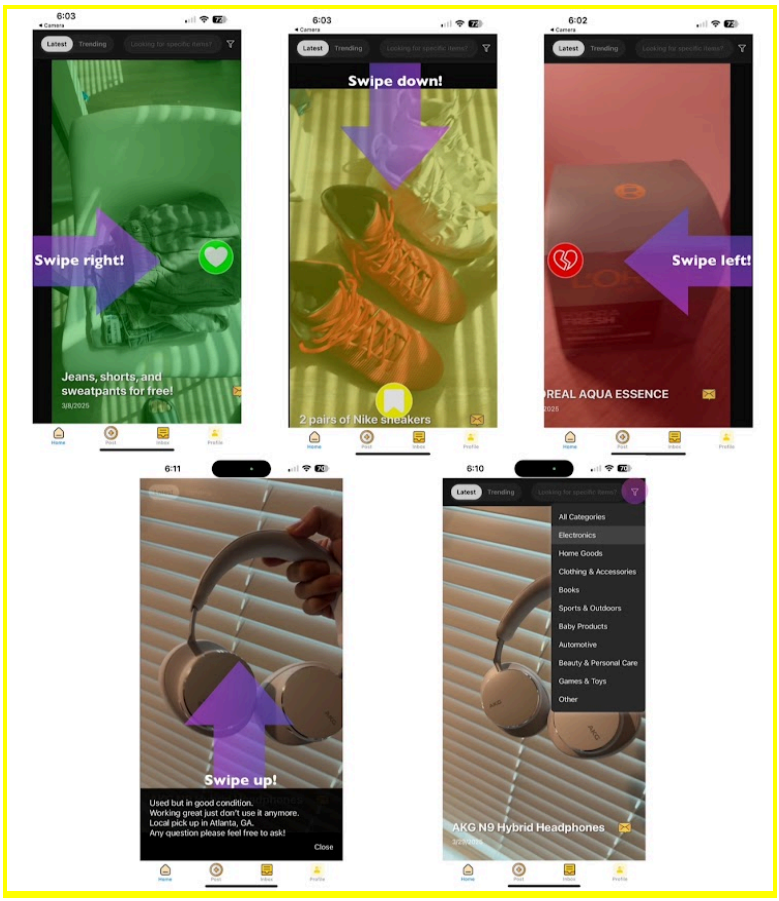
   - Users can post videos without including location information.

   B:



- Users need to agree to PayNothing to obtain location permissions when entering the post page. On the post information page, location permissions must be enabled to obtain the video shooting location.

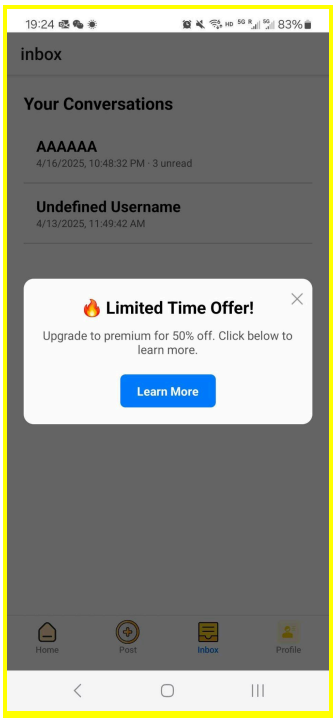
2. **How many times does a user match in 'home screen'?** *(Sprint 5 prototype only)*
   → Sprint 5 includes live matching and auto-connection to inbox, increasing match rate visibility.



- Matching Algorithm: if two users “like” each other’s video posts, they are automatically connected in the inbox screen, enabling direct communication to accelerate item exchanges.

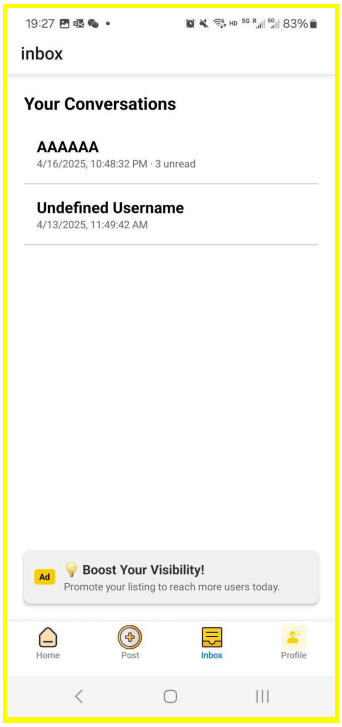
3. **How many times were banner and full-screen ads clicked in ‘inbox screen’?** *(Sprint 5 prototype only)*
   → We introduced ad placements to gauge impact without degrading user experience.

   A:



- Full-screen ads: Full-screen ads will be displayed when the user first enters the inbox page. After the user closes it, it will not be displayed again in this app.

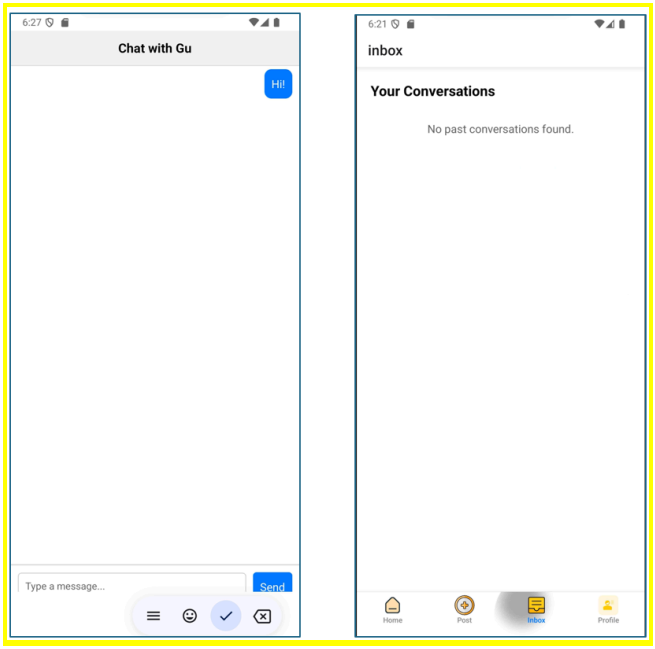
B:



- Banner ads: Banners will always be displayed on the inbox page, and users cannot close the ads directly.

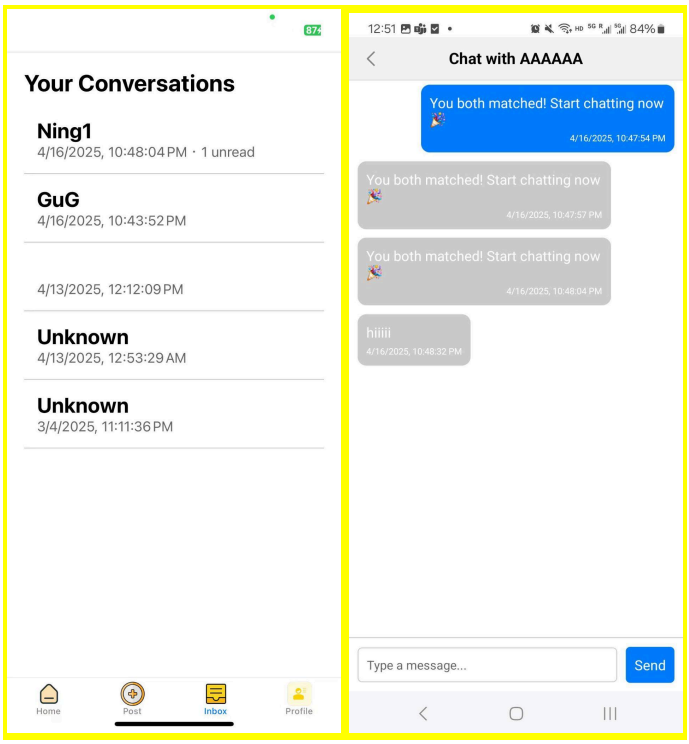
4. **How long does it take for the user to match/make an exchange with another user in the previous prototype vs. current MVP?**
   → Sprint 5 shows a decrease in time-to-match due to the real-time syncing and direct inbox linking.

   A:



- Users can view previous chat records on the inbox page and browse previous conversation information in each session.

B:



- Users can view previous chat records on the inbox page, and it can show whether they have been read. In the conversation, each message will contain a timestamp.

5. **How long does it take to create a new account and log in?**

   → *Sprint 5 retained previous improvements with one-time authentication at app launch.*

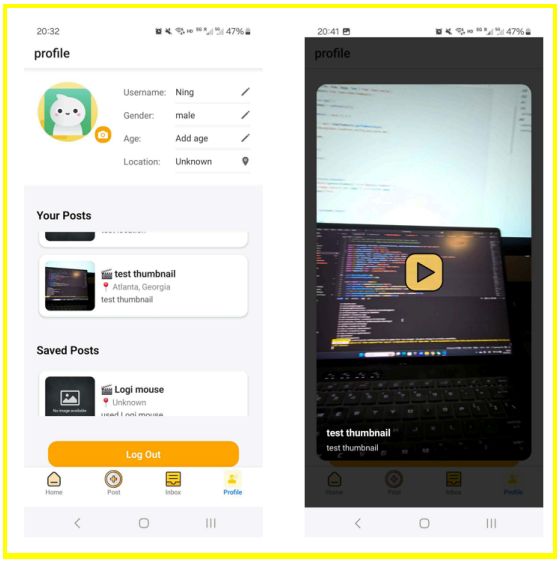
   - To evaluate the effectiveness and speed of our onboarding process. A seamless and quick authentication flow is essential to minimize user drop-off at the entry point of the app. By timing this process, we ensure our login experience remains efficient, especially as we layer in new features like real-time syncing and ad displays that could impact load times.

6. **Which page does the user spend the most time on?**

   → *Sprint 5 analytics show increased time on the "Home" feed due to more engaging, authentic videos.*

   - To understand where users find the most value and engagement within the app. By identifying high-retention areas like the "Home" screen, we can allocate design and engineering resources to further enhance the content experience. Increased time spent here validates the effectiveness of in-app video recording, geolocation filtering, and swipe-based discovery introduced in Sprint 5.

7. **Rate your overall user experience on a scale of 10, and any other comments?**
   → *Most testers noted higher satisfaction with video authenticity, real-time interaction, and location relevance.*
   - To gather a holistic, user-centric measure of satisfaction beyond individual features. This question captures both quantitative ratings and qualitative insights, helping us evaluate whether recent feature updates (e.g., location-locked posting, real-time feed updates, matching logic) are translating into meaningful improvements in perceived app value and usability.



   - In Profile screen, we display the cover image, title, description and location information for each record. When users click on a record, they can browse the corresponding video.

Sprint 4:

In Sprint 4, our team focused on developing the second learning prototype with the primary objective of refining the **user experience**, **user interface**, and ensuring **functional clarity** in how users interact with the Posting and Video Feed features. Building on the feedback from Sprint 3, we identified that while users appreciated the core idea of video-based item listings and localized bartering, there was a clear demand for improvements in **app fluidity, visual consistency, interaction flow, and ease of use**. As a result, Sprint 4 shifted our development and validation focus toward **UI improvements, feature implementation, and A/B testing**, prioritizing swiping mechanics and seamless authentication over deeper backend integrations like user rating weight systems. Officially have home screen functionality smoothened out as video interaction is done in a swiping manner (swipe left to dislike and swipe right to like, and swipe down to save video). Additionally, videos can now have tags associated with them that

can be placed by the seller to categorize videos, making finding a specific item easier. User authentication has now been finalized, as users must authenticate on app launch before accessing any other features.

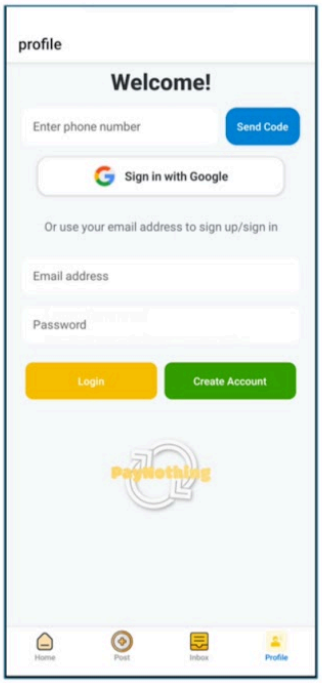
Originally, we wanted this learning prototype to incorporate major technical changes in code which included additional features such as an advanced search bar algorithm and a weight user rating system. We would later plan to incorporate this in our A/B testing to gauge user feedback on our additional features. However, Sprint 3 user testing yielded that these features are not a high priority and not what users wanted the most. Not only that, the technical incorporation into the app integration was quite technically complex, and with a smaller Sprint 4 window, we pivoted away from this approach. Instead, feedback from previous sprint was primarily based on UI changes, convenience of scrolling, and overall appearance of the app paired with ease of use. Therefore, this learning prototype incorporated those changes. We aimed to narrow our focus on **what users valued most**: **intuitive interactions, aesthetic enhancements, smoother scrolling, and instant engagement**. We redesigned the home screen interaction and implemented swipe-based mechanics (left to dislike, right to like, up for item details, down to save). In this way, we adopt familiar patterns inspired by popular platforms like TikTok, Instagram Reels, and Tinder. This is why we changed the approach to A/B testing regarding UI changes, app layout, and likeability of newer features (swiping left/right/up/down, user authentication, and video filters).

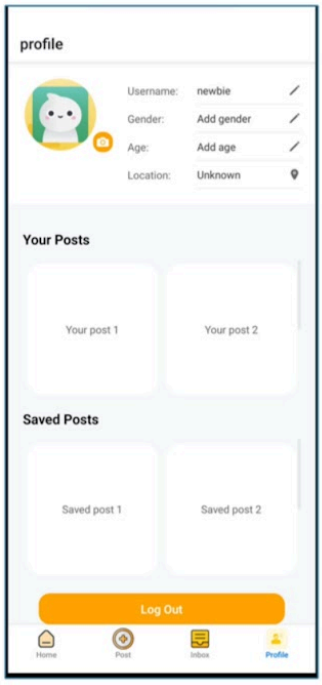
To validate these changes, we conducted structured A/B testing to compare older interaction models with the new prototype experience. Each version was assessed in user interviews to gather quantitative scores and qualitative feedback based on intuitiveness, engagement, and satisfaction.

A/B testing is listed below:

Authentication Flow

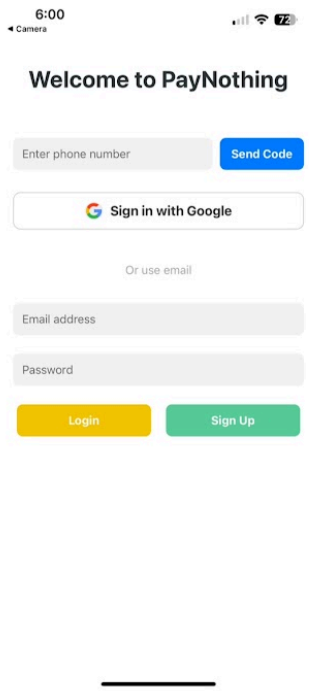
A:

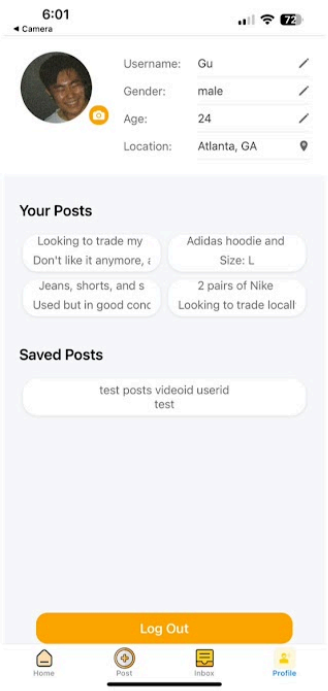




- Users could browse the feed before logging in, but access to core features like posting, liking, and messaging was restricted until they manually navigated to the Profile tab to sign in.

B:



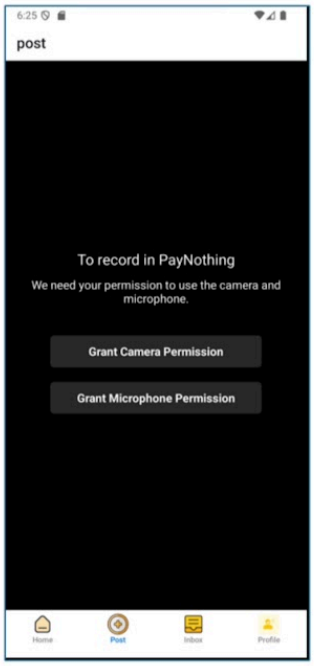


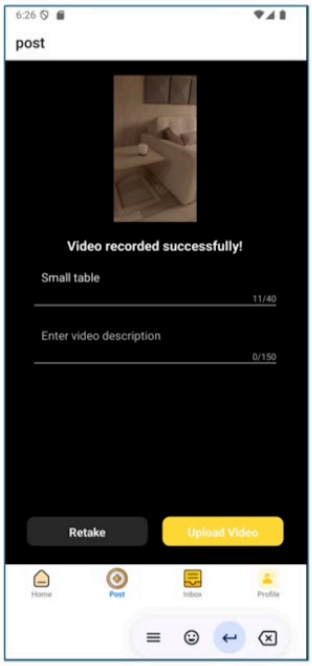
- Users were required to sign in or sign up at app launch before accessing any functionality. In the “Profile” page, all user data (past posts, saved items) is auto-loaded upon login, streamlining access to core features.

Outcome: Variant B was overwhelmingly preferred for its simplicity and clear access boundaries, ensuring users didn't interact halfway through and get blocked later.

Video Posting Interface

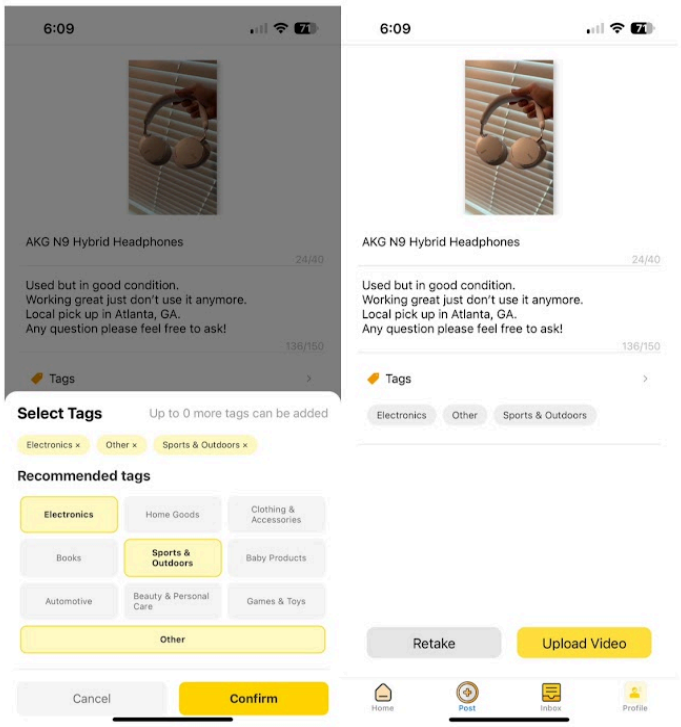
A:





- Users could record and post a short video of the item they want to trade or give away. Users can introduce the item via verbal explanation. They could also manually add a title and description post-recording.

B:

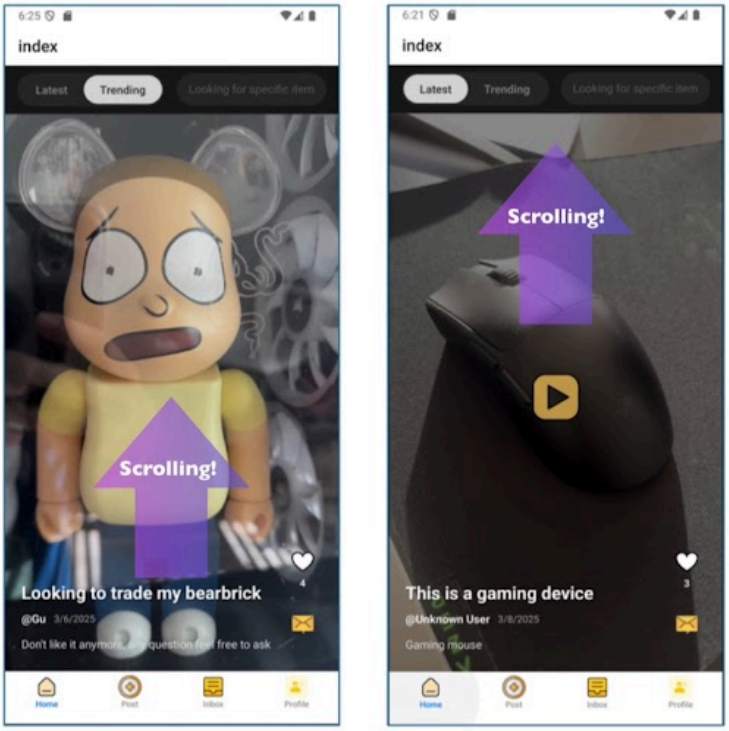


- The interface was redesigned with a white and yellow color scheme and introduced tagging functionality, allowing users to categorize their listings (e.g., #clothing, #electronics, #books).

Outcome: Variant B improved content organization and discoverability, with users noting that tags helped them “find relevant items faster” and made posts feel “cleaner and more organized.”

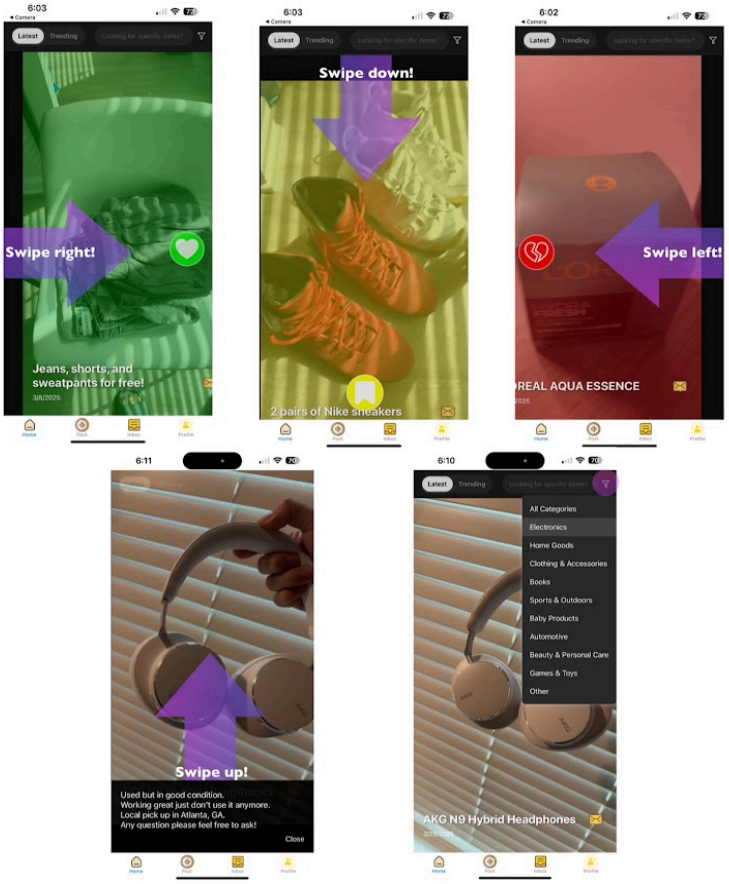
Video Feed Interaction

A:



- Users could scroll vertically through videos to view posted items.
- Users could tap the “heart” icon to like a post.
- Users could tap the “message” icon to chat with other users.
- Item details (e.g., title and description) were persistently displayed at the bottom of each video.

B:



- Users could filter items based on categories.
- Users could view and interact with one video post at a time with a gesture-based interaction model:
    - Swipe left to 'dislike' the item.
    - Swipe right to 'like' the item.
    - Swipe up to reveal full item details.
    - Swipe down to save the video for later.
- An "instant match" happens if both users like each other's posted items, enabling immediate chat initiation.

Outcome: Variant B significantly boosted engagement, with users describing it as "more fun," "interactive," and "more intuitive, like using a dating app." Many also appreciated the cleaner focus on one item at a time.

What we valued the most from this learning objective is to revise design and feature specifics based on the user feedback from our latest learning prototype. Planning of our next sprint will be better explained in learning prototype plans, see learning prototype plans for future plans.

Posting Item and Video Feeds (selected)

- **Develop & Test:**
    - Interviews and A/B test on the updated prototype (Done): Participants are asked questions to compare the first and the second learning prototypes and given feedback after experiencing our app. We also provide a comparison between the first prototype and the improved prototype to allow users to confirm whether our improved prototype is effective.
    - Updated storyboard & visuals (Done): Update storyboards and screenshots to provide the new Posting and Video Feeds scenario, and use it to conduct a wide range of user surveys.
    - UI enhancements, function development, and bug fixes (Done): Improve the Posting and Video Feeds functionality, fix bugs, use new UI/UX design and new video interaction mode.

Sprint 3:

Posting Item and Video Feeds (selected)

- **Unknowns:**
    - How convenient is “posting items and browsing video Feeds'” on a scale of 1-10? May compare with popular e-commerce platforms that you previously used.
    - How easy is it to browse videos on the home page, did you find any issues or some other features you wish it can have?
    - How was your experience of posting videos? Were there any issues or something you wish could be different?
    - How was your experience of messaging someone? Were there any issues or something you wish could be different?
    - If you like an item you saw, would you message them? Did your app make it easy for you to contact them?
    - For the overall functionality of the app, were you able to effectively browse video feeds, post videos, and message users? If not, why not?

- **Develop & Test:**
    - Interviews on the first learning prototype (Done): Participants are asked all the Unknowns and provided feedback after experiencing our app.
    - Storyboard survey (Done): Use storyboards to create a Posting and video feeds scenario and use it to conduct a number of user surveys[4].
    - Function development (Done): Initially implement the Posting and video feeds functions and allow users to experience them.

Community-based leaderboards (needs and haves)

[4] Please check “Learning prototype results” for more details.

- **Unknowns:**
    - Will the leaderboard provide useful insights to encourage user engagement?

- **Develop & Test:**
    - Storyboard survey (Done): Use storyboard to create a scenario and use the scenario to conduct a certain number of user surveys (Learning prototype results). Haves and Needs could be displayed.
    - Function development (Not done): Develop a simple ranking page and use firebase to track the number of clicks on the current page. Use user activity and clicks to understand whether users are willing to use the ranking page.

Gamification and Reward System

- **Unknowns:**
    - Will users be motivated by a crypto-coin-base system?
    - What rewards mechanisms are most effective in driving engagement?
    - Could users find ways to exploit the system unfairly?

- **Develop & Test:**
    - Launch a small scale crypto-coin reward with a limited amount.
    - Analyze transaction frequency before and after reward implementation.
    - Introduce different reward ways and track which incentives drive the most engagement.
    - Monitor potential abuse of the system and adjust rules as needed.

Sprint 2:

Posting Item and Video Feeds (selected)

- **Unknowns:**
    - Will users find this approach of posting and video feeds interesting?
    - Will users accept that short videos can be used as a better way to barter?

- **Develop & Test:**
    - Storyboard survey (Done): Use storyboard to create a Posting and video feeds scenario and use it to conduct a number of user surveys.
    - Function development (Done): Initially implement the Posting and video feeds functions and allow users to experience them.

Sprint 1:

AI-Powered Video Feeds and Item Matching

- **Unknowns:**
    - Will AI successfully detect spams or scammers' videos?
    - Will item recommendations actually be relevant to the user's preferences?

- **Develop & Test:**
    - For spam detection, we can design test cases that randomly create some user accounts to be in the role of scammers and post videos that do not match their item descriptions, or user locations are far away from video post locations. Testing against our basic detection algorithm and optimizing it for better performance.
    - For content recommendations, deploy a basic matching algorithm that collects user data such as titles from liked posts and generated transcripts from videos they posted. Utilize these data to feed personalized videos to users, and assess user satisfaction through surveys.

Technical Discussion

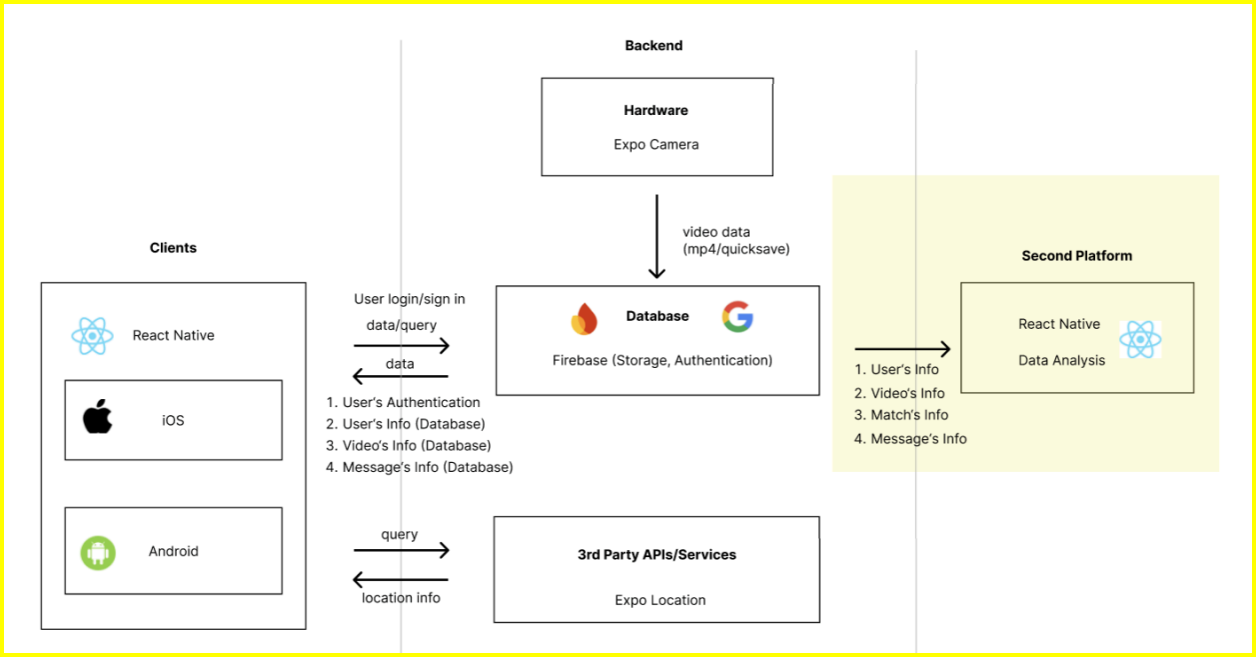
Sprint 5:

The PayNothing mobile app is built with React Native using the Expo Framework, ensuring cross-platform support for both iOS and Android devices. The stack emphasizes modularity, offline capability, and real-time data responsiveness, using the following technologies:

Tools	Purposes:
Firebase Authentication	User sign-up/sign-in with email/password, phone, and Google
Firebase Firestore	Real-time NoSQL database for storing user profiles, messages, and video post metadata
Firebase Storage	Upload and host video recordings (mp4) and thumbnails
Expo Camera	Capture video content directly from users
Expo Location	Fetch and reverse-geocode the userâ€™s city/state for tagging posts
Expo AV / Video	Playback for recorded and saved content
Expo Notifications	Push notifications for matches and interactions

React Native Reanimated + Gesture Handler	Smooth, physics-based swipe interaction on the Home feed

1. Architecture



2. Client–Backend Interactions

- User Authentication: Managed by Firebase Authentication SDKs across platforms.
- User & Video Data: Stored and retrieved from Firestore, using real-time listeners (onSnapshot) for chat and feed.
- Video Uploads: Handled via Expo Camera and uploaded to Firebase Storage.
- Location Services: Queried from the device using Expo Location and optionally stored alongside video metadata.
- Media Playback: Implemented using Expo AV and displayed in full-screen modals or swipeable lists.

3. Modular Feature Implementation (TypeScript)

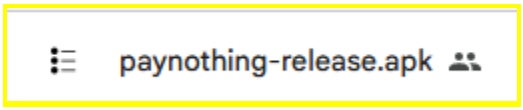
- Initial Layout & Navigation (app/_layout.tsx)
    - Conditionally routes users to /auth if not logged in.
    - Routes authenticated users to either the tab layout or chat screen.
    - Guards routes using onAuthStateChanged and useSegments.
- Authentication Page (app/auth.tsx)
    - Implements multiple auth flows: email/password, Google, phone.
    - On successful login, redirects to the tab navigator.
    - Creates default Firestore user record upon sign-up.

- Tab Navigator (app/(tabs)/_layout.tsx)
  - Defines bottom navigation for Home, Post, Inbox, and Profile.
  - Prevents unauthenticated users from posting (intercepts tab press).
- Home Screen (app/(tabs)/index.tsx)
  - Video posts are displayed one at a time.
  - Swipe-based gesture interactions:
    - Left = Dislike
    - Right = Like → checks for match and creates a chat
    - Up = View Description
    - Down = Save video
  - Sort and filter capabilities:
    - Search bar filters by title.
    - Tag filter uses TagSelector modal.
    - Trending vs. Latest sorting implemented by likes and timestamps.
- Post Creation (app/(tabs)/post.tsx)
  - Uses Expo Camera to record up to 60 seconds of video.
  - Captures thumbnail with expo-video-thumbnails.
  - Collects metadata: title, description, tags, location.
  - On upload:
    - Stores video in Firebase Storage.
    - Saves metadata to Firestore and links to the user's post history.
- Inbox Messaging (app/(tabs)/inbox.tsx)
  - Lists past conversations sorted by timestamp.
  - Displays unread message counts.
  - Shows either banner or interstitial ads via Ads.tsx.
- Chat Page (app/chat.tsx)
  - Two-person chat with real-time updates using onSnapshot.
  - Messages are stored in Firestore under a shared conversationId.
  - Unread messages marked as read on load.
- User Profile (app/(tabs)/profile.tsx)
  - Edit profile photo, username, gender, age
  - Retrieve location via expo-location
  - Browse previously posted and saved videos
  - Registers Expo Push Token after login for messaging/notifications.
- Ads & UI Components
  - app/components/Ads.tsx – Shows either banner or modal interstitial ads.
  - app/components/TagSelection.tsx – Tag selector modal with up to 3 custom tags.
  - app/components/VideoViewModal.tsx – Full-screen modal for saved or posted videos.

4. Android APK

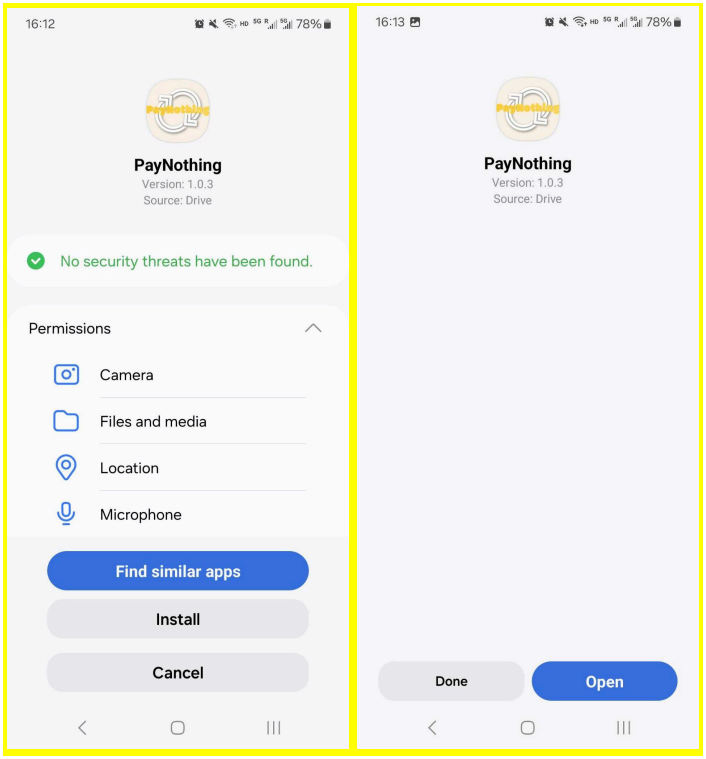
We successfully packaged the PayNothing app into a standalone APK file, allowing it to be installed and used directly on Android devices without relying on Metro bundling. By running **gradlew assembleRelease**, we generated the release version APK located at

android/app/build/outputs/apk/release/app-release.apk. This APK can be manually transferred to an Android phone for offline installation and full functionality testing, enabling smoother user testing and deployment preparation.



Apk Download link:

https://drive.google.com/file/d/1krR71HPrq8icnzXElwV-IgZCRz5pYV-9/view?usp=drive_link



5. Second Platform Backend Dashboard

App Layout & Navigation (_layout.tsx):

- Sets up the main app navigation with a Stack navigator.
- Applies dark/light themes automatically using useColorScheme().
- Prevents the splash screen from auto-hiding until fonts are loaded.
- Creates a tab navigator with "Home" and "Explore" tabs.

Home Dashboard Screen (index.tsx):

- Fetches backend metrics: total users, videos, messages, matches.
- Displays live stats using StatCard components.
- Implements pull-to-refresh functionality to reload backend data.
- Displays recent user activities with the ActivityList component.

- Shows time-series charts for video uploads, messages, matches, and active users.

Activity Feed Component (ActivityList.tsx):

- Renders a list of recent activities like new users, videos, and matches.
- Displays an empty state message when there are no activities.
- Formats timestamps for activities into readable dates.

Daily Active Users Chart Component (DAUChart.tsx):

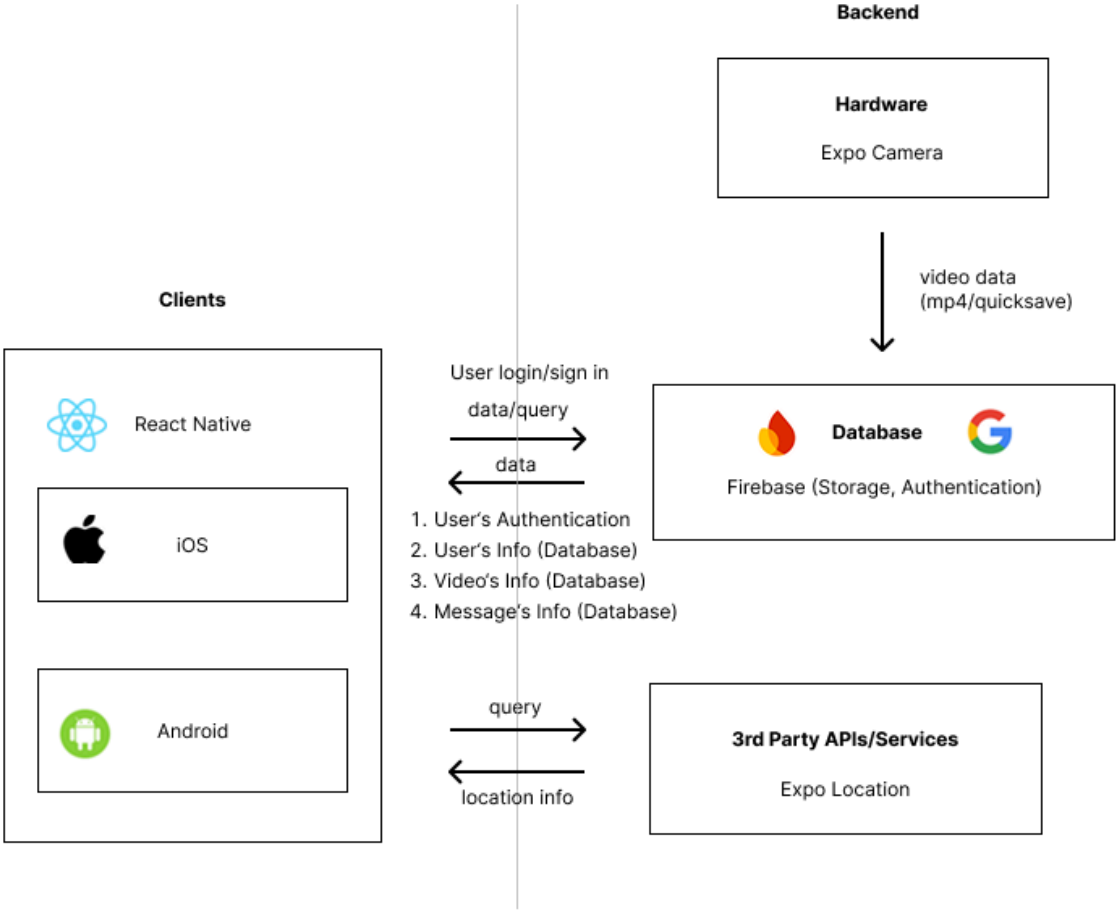
- Lists daily active user counts in a readable text format.
- Formats dates into MM/DD format for quick scanning.
- Provides a lightweight alternative to full graph-based charts.

Sprint 4:

Our App and its functionalities are developed using React Native with Expo Framework, making it cross-platform for both iOS and Android mobile devices.

1. Firebase for authentication such as sign-up, sign-in, and user profile management.
2. Firebase for database cloud storage such as storing user messaging data, videos, etc.
3. Expo Camera for media capture and Expo Location for location services.

PayNothing Architecture Diagram:



Features are implemented using TypeScript as following:

- **Initial layout and app navigation** (*_layout.tsx*):
    - Set up the initial page routing to conditionally display the Auth screen, chat screen, and subsequent (tabs) home, post, inbox and profile screens.
    - If the user is not logged in, redirect to the Auth screen to ensure the user registers/logs in.
    - If the user chooses to enter the chat, the Chat screen will be jumped and displayed in full screen.

- **Chat Screen** (*chat.tsx*):
    - Displays the chat screen between a user and another user in full screen.
    - User messages are on the right, and the other party's messages are on the left, and they are distinguished by different colors. The messages will be sorted in chronological order.
    - New messages will be stored in Firestore's "messages". Combined with each conversation id, ensure that both chat records are stored in the same session.

- **Authorization login registration page** (*auth.tsx*):
    - Integrated Firebase authentication that allows users to have multiple sign-up/sign-in methods as such:
        1. Login via phone number: Upon passing the reCAPTCHA verifier, it'll send a verification code to the user's phone so the user can sign in with the verification code.
        2. Sign in with Google using a Gmail account.
        3. Login/Create an account by entering email address and password.
    - Upon signing in, the user will be redirected to the (tabs) page, and the corresponding page will be displayed according to /(tabs)/_layout.tsx.

- **Basic layout and app navigation** (/(tabs)/*_layout.tsx*):
    - Established four screens (home, post, inbox, profile) and navigation between these screens by tapping their icons in the lower bar area. Make sure users are authorized in the login page first to access all other pages.

- **Home Screen** (/(tabs)*index.tsx*):
    - User interact with one video post at a time:
        1. Swipe left to 'dislike' the item, swipe right to 'like' the item.
        2. Swipe up to view the item description.
        3. Swipe down to save the video for later.
        4. Instant match happens when two posted items 'like' each other.
    - A search bar located in the top right corner of the screen to filter by title of posts.
    - A filter icon is on the right side of the search bar. When the user clicks it, a drop-down box of all tags will be displayed, and the user can select the corresponding tag to filter the video.
    - Touchable buttons such as “Latest” and “Trending” are available to tap, and are fully functional in this sprint.
    - Trending sort by highest likes, and latest sort by upload time. Allow users to sort by relevance and popularity.
    - Message buttons also exist on the post, allowing users to directly click the message button. This will automatically take this to the inbox page, allowing direct messaging with the seller.

- **Post Screen** (*/(tabs)/post.tsx*):
    - When entering the post page in the logged-in state, for users who have not authorized camera and microphone permissions, the camera and microphone permission application windows will automatically pop up in sequence; if the user does not click the permission window that pops up automatically, the page will also display the current authorization status of the two permissions, and the user can click to apply for permissions again.
    - Upon approval, it can show the back camera view, timer and a touchable button for users to tap to record or stop recording.
    - After recording, display the current video cover, and let the user add a title, description, and tags before posting.

    - Modify the UI and UX styles and change the overall background to white. Add tag selection function: After the user clicks on a tag, a pop-up window will be displayed to dynamically modify the tag corresponding to the video.
    - Once uploaded, recordings are saved as user data to Firebase Storage. Save the video information and add the record to the user's history posts.

- **Inbox Screen** (*/(tabs)/inbox.tsx*):
    - The inbox screen has now been implemented with a proper UI that allows for direct messaging capabilities between two users.
    - A user can send messages to other users and receive messages from others.
    - The past conversations that the user had with other users are displayed as a list on the inbox screen, each with the title of the other user's username.
    - The user is able to select any past conversation from the list and open a new chat window to continue the conversation.
    - The user is required to log in first before viewing the chat history.
    - All messages are now stored in Firestore Database that are linked to the user.

- **Profile Screen** (*/(tabs)/profile.tsx*):
    - Upon signing in, users can upload images from local photos to edit their profile picture, and they can edit their username, gender, and ages. User's location retrieval is also implemented via Expo Location, which asks users for location services permission for the first time and will display (city, state) upon approval.
    - Upon signing in, users can browse their previously published posts and saved posts.

<u>Sprint 3:</u>

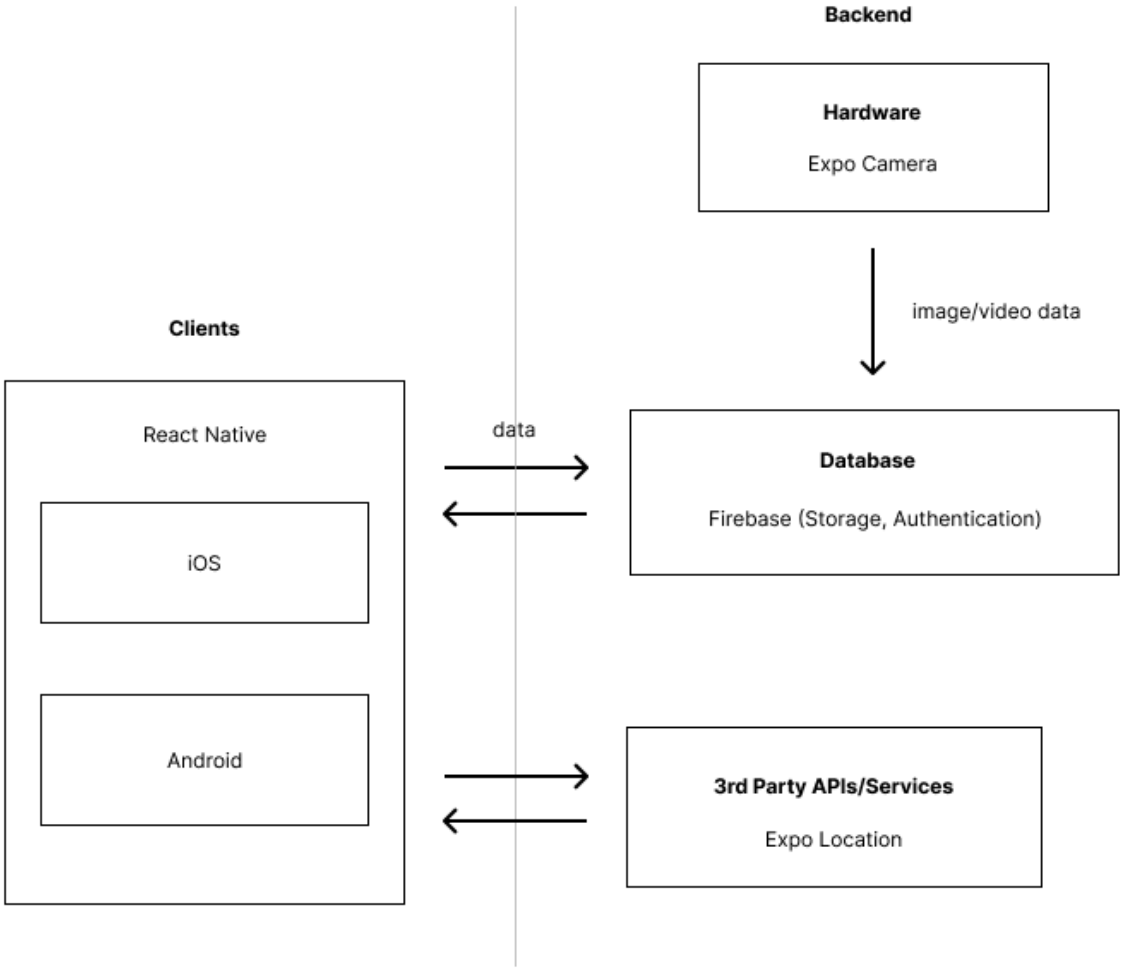
Our App and its functionalities will be developed using **React Native with Expo Framework**, making it cross-platform for both **iOS** and **Android** mobile devices.

1. Firebase for authentication such as sign-up, sign-in, and user profile management.
2. Firebase for database storage such as storing user messaging data, videos, etc.
3. Expo Camera for media capture and Expo Location for location services.

Features are implemented using TypeScript as following:

- **Basic layout and app navigation** (*_layout.tsx*):
    - Established four screens (home, post, inbox, profile) and navigation between these screens by tapping their icons in the lower bar area.
- **Home Screen** (*index.tsx*):
    - Displays a single video post with item title and a touchable “Like” button to count how many likes of this post.
    - Users can scroll to view the next video post, acting as a for you page.
    - A search bar located in the top right corner of the screen to filter by title of posts.

    - Touchable buttons such as `Latest' and `Trending' are available to tap, and are fully functional in this sprint.
    - Trending sort by highest likes, and latest sort by upload time. Allow users to sort by relevance and popularity.
    - Message buttons also exist on the post, allowing users to directly click the message button. This will automatically take this to the inbox page, allowing direct messaging with the seller.
- **Post Screen** (*post.tsx*):
    - To enter the post page, users first need to log in; for users who are not logged in, they will be directly redirected to the profile page and prompted to log in.
    - When entering the post page in the logged-in state, for users who have not authorized camera and microphone permissions, the camera and microphone permission application windows will automatically pop up in sequence; if the user does not click the permission window that pops up automatically, the page will also display the current authorization status of the two permissions, and the user can click to apply for permissions again.
    - Upon approval, it can show the back camera view, timer and a touchable button for users to tap to record or stop recording. After recording, display the current video cover, and let the user add a title and description before posting. Once uploaded, recordings are saved as user data to Firebase Storage.
- **Inbox Screen** (*inbox.tsx*):
    - The inbox screen has now been implemented with a proper UI that allows for direct messaging capabilities between two users.
    - A user can send messages to other users and receive messages from others.
    - The past conversations that the user had with other users are displayed as a list on the inbox screen, each with the title of the other user's username.
    - The user is able to select any past conversation from the list and open a new chat window to continue the conversation.
    - The user is required to log in first before viewing the chat history.
    - All messages are now stored in Firestore Database that are linked to the user.
- **Profile Screen** (*profile.tsx*):
    - Integrated Firebase authentication that allows users to have multiple sign-up/sign-in methods as such:
        1. Login via phone number: Upon passing the reCAPTCHA verifier, it'll send a verification code to the user's phone so the user can sign in with the verification code.
        2. Sign in with Google using a Gmail account.
        3. Login/Create an account by entering email address and password.
    - Upon signing in, users can upload images from local photos to edit their profile picture, and they can edit their username, gender, and ages. User's location retrieval is also implemented via Expo Location, which asks users for location services permission for the first time and will display (city, state) upon approval.



Sprint 2:

Our App and its functionalities will be developed using **React Native with Expo Framework**, making it cross-platform for both **iOS** and **Android** mobile devices.

1. Firebase for authentication such as sign-up, sign-in, and user profile management.
2. Firebase for database storage such as storing user messaging data, videos, etc.
3. Expo Camera for media capture and Expo Location for location services.

Features are implemented using TypeScript as following:

- **Basic layout and app navigation** (*_layout.tsx*):
    - Established four screens (home, post, inbox, profile) and navigation between these screens by tapping their icons in the lower bar area.
- **Home Screen** (*index.tsx*):

    - Displays a single video post with item title and a touchable “Like” button to count how many likes of this post.
    - Users can scroll to view the next video post, but minor problems need to be fixed later.
    - A search bar located on the top right corner of the screen to filter by title of posts.
    - Touchable buttons such as “Latest” and “Trending” are available to tap, but are still in development of their functionalities.
- **Post Screen** (*post.tsx*):
    - Users will be asked for phone's camera and microphone permissions for the first time.
    - Upon approval, it can show the back camera view and a touchable button for users to tap to record or stop recording.
    - Recordings are saved as user data to Firebase Storage.
- **Inbox Screen** (*inbox.tsx*):
    - Initialized but still in development.
- **Profile Screen** (*profile.tsx*):
    - Integrated Firebase authentication that allows users to login or sign up with an email address and password.
    - Upon signing in, users can edit their username.
    - User's location retrieval is also implemented via Expo Location, which asks users for location services permission for the first time and will display (city, state) upon approval.

Second Platform

Sprint 5:

The “Second Platform” section of PayNothing is designed as a comprehensive analytics and insights hub built with React and Next.js, providing both end-users and administrators with a clear, data-driven window into how the app is performing and how the community is engaging. At its highest level, this section transforms raw backend counts into meaningful narratives about growth, activity, and trends—turning abstract numbers into actionable intelligence. By surfacing metrics like total user sign-ups, video posts, messages exchanged, matches made, and daily active users, we’re giving everyone from individual contributors to project stakeholders a shared understanding of PayNothing’s health and momentum.

First, the Core Metrics pane delivers a real-time snapshot of the platform’s scale: cumulative users, videos listed, messages sent, and successful peer-to-peer matches. Displayed prominently at the top, these four statistics serve as headline indicators—similar to a social-media “like” count or transaction tally—immediately conveying the breadth of activity on PayNothing. This high-visibility placement emphasizes transparency: any team member or

investor can open the dashboard and instantly see, for example, that we've onboarded 20 users or facilitated 33 video listings so far in the history of our app.

Beneath that, the Trends row contextualizes week-over-week changes. Instead of just knowing "how many" users signed up, we reveal "how fast" that number is growing—percentage increases or decreases in new sign-ups, uploads per day, messages today versus yesterday, and 24-hour active-user counts. By framing these as comparative statistics (current 7-day sum versus the prior 7 days), we spotlight momentum and inflection points: a sudden spike in messages might signal a particularly popular event or feature launch, while a plateau in new users could trigger targeted acquisition campaigns.

The heart of the section is the Time-Series Growth Rate—four interchangeable views that let you drill into daily activity over the past month. Whether you're tracking video upload cadence, message volume, match volume, or daily active users, each section lays out the percentage growth, making seasonal cycles, weekend lulls, or viral spikes immediately visible. A simple control lets you see between these different perspectives without leaving the page, putting a full month of history at your fingertips in one compact, self-contained box.

Together, these four pillars (Core Metrics, Trends, Time-Series, and Growth Rates) compose a robust analytics platform that serves multiple audiences. Product managers can identify drop-off days and plan feature launches; growth marketers can tie campaigns to upticks in sign-ups; community leads can track engagement and spotlight power users; engineers can correlate infrastructure changes with performance shifts; and investors can get an at-a-glance view of our trajectory. By packaging all these insights into a mobile-friendly dashboard, PayNothing not only democratizes data access but also fosters a culture of continuous improvement, where every team member can see the impact of their work and make fact-based decisions to accelerate the platform's success.

Relation to A/B Testing:

This dashboard is integral to our A/B testing methodology, providing the quantitative backbone needed to move beyond anecdotal feedback into rigorous, data-driven decision making. In practice, whenever we roll out a new prototype or tweak a UI element—whether it's refining the swipe gestures on the Home feed, adjusting the layout of the Post screen, or introducing a streamlined authentication flow—we segment users into control and variant groups and monitor how those segments perform across the Core Metrics and Trends sections.

For example, if Prototype A persists with the original photo-only listing flow while Prototype B introduces in-app video capture, we don't just ask users which they "like" better: we measure how many additional sign-ups, uploads, and matches occur in the B cohort over the subsequent seven days compared to the week before. The "Growth Rates (7d vs Prior 7d)" cards immediately surface whether that variant drove a statistically meaningful lift—or if engagement remained flat or even dipped, signalling the need for further iteration.

Moreover, the segmented time-series views let us pinpoint precisely when an experiment's impact manifests. Did Daily Active Users jump only on launch day, then taper off? Did Message Volumes rise only after we adjusted the inbox UI? By cross-referencing these charts with experiment start and end dates, we can correlate feature releases with usage patterns, validate hypotheses about user behavior, and confidently decide whether to roll changes out to 100% of our audience or to pivot, ensuring that every UI enhancement is backed by measurable gains in acquisition, retention, or engagement.

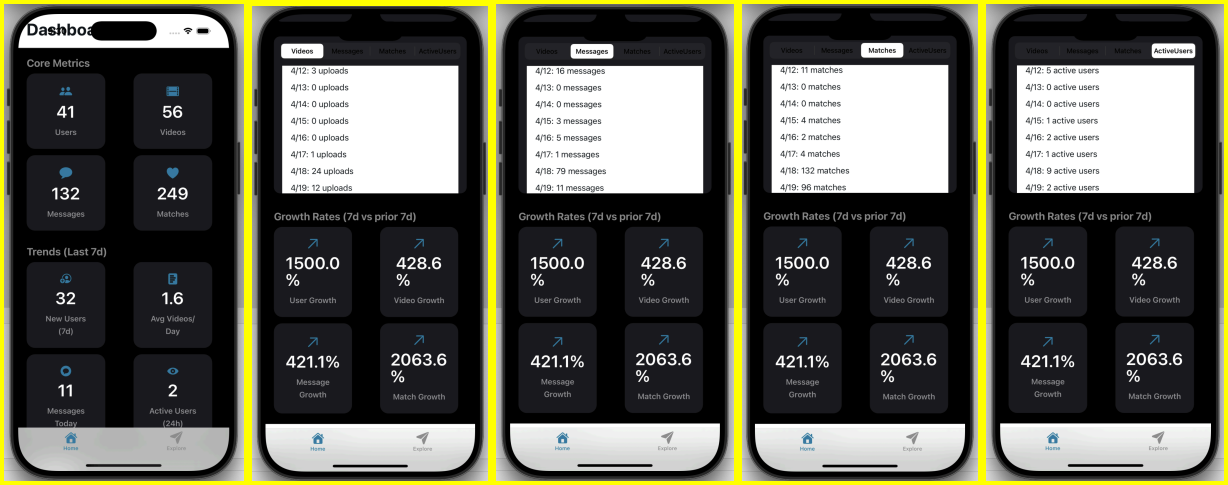


Figure 8: Screenshots of the second platform (including Core Metrics, Trends, Time-Series and Growth Rates sections)

By extending our existing architecture rather than building a siloed solution, the administrative dashboard delivers transparent, up-to-the-second insights and controls that inform feature rollouts, monetization experiments, and community-safety interventions. This ensures that every strategic decision is grounded in the same live data that powers PayNothing's core mobile exchange.

Business Model Canvas

Sprint 5:

Our Business Model Canvas has been sharpened to reflect the solid foundation we've already built, the work that is in progress, and the strategic extensions we're planning. Today, PayNothing's Expo-powered mobile app and Firebase backend seamlessly deliver location-based video listings, letting users post and browse items in seconds without writing a single line of text. Our in-app messaging and real-time chat keep negotiations safe and private, while LoopCoin gamification rewards community engagement and encourages repeat visits. These core capabilities, coupled with AI-powered filtering to weed out suspicious listings and a

no-fee, video-first barter marketplace, have already won over families, students, and eco-minded thrifters seeking an authentic, hassle-free trading experience.

Building on this momentum, we're layering in planned enhancements that enrich discovery and bolster trust. Personalized "For You" feeds will guide users to items they care about, environmental badges will showcase the positive impact of every upcycle, and verified-user badges plus real-time fraud alerts will deepen safety and confidence. On the revenue side, optional premium boosts—like post promotion and instant location reveals—will dovetail with localized ad partnerships and LoopCoin top-ups, ensuring we stay free at the core while opening new, value-aligned monetization channels. By doubling down on our fully implemented strengths and carefully rolling out these next-wave features, PayNothing is poised to become the go-to platform for hyperlocal, sustainable bartering.

- Green Text: Implemented
- Blue Text: In progress
- Yellow Text: Planned

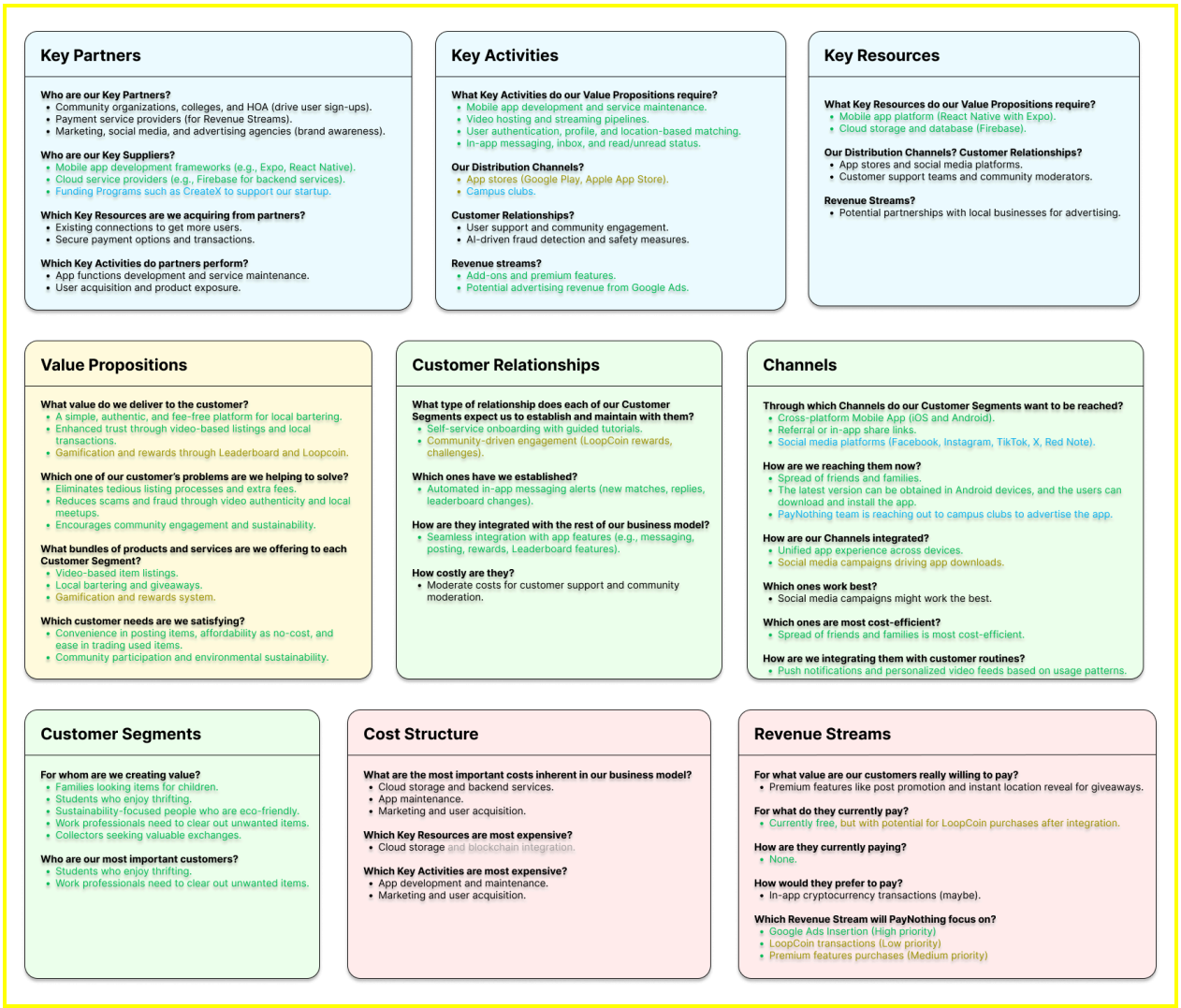


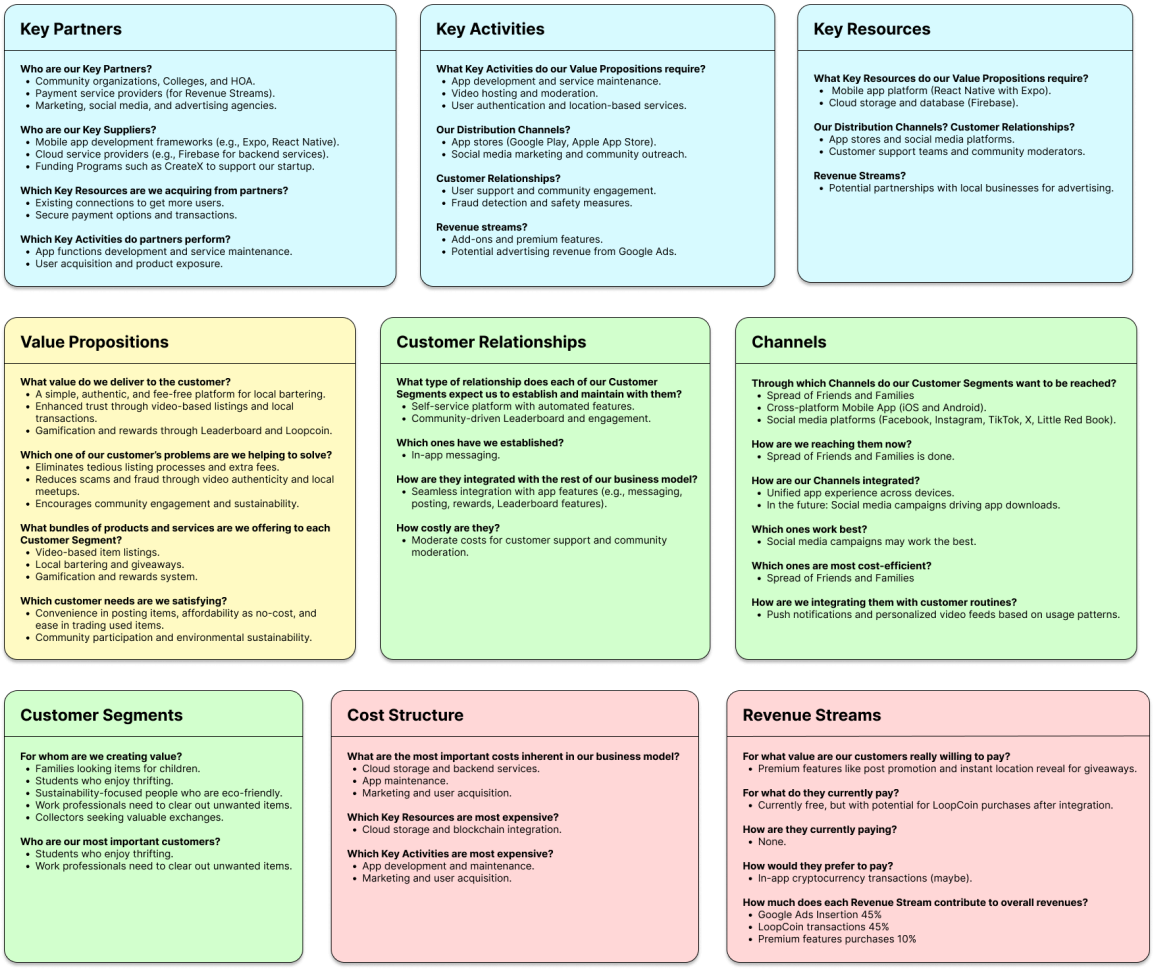
Figure 9: The Business Model Canvas of our product

Sprint 4:

Our business model centers around delivering a zero-cost, community-powered platform for local bartering and giveaways, using video content as the core medium of exchange. By removing common barriers such as listing complexity, shipping hassles, and transaction fees found on traditional platforms like eBay or Facebook Marketplace, PayNothing positions itself as a highly accessible and trustworthy alternative. Its unique approach to leverage short-form video listings offers a more authentic and engaging way to showcase items, significantly reducing scams while simplifying the posting process. The platform is designed for a diverse set of user segments including students, families, professionals, and collectors, all of whom benefit from a system that values trust, ease, and sustainability over profit-maximization.

Revenue generation in PayNothing is structured around optional value-added features rather than mandatory fees. While the core trading experience remains free, users can earn or purchase LoopCoin, a platform-specific reward currency that enables perks such as post promotion and instant location reveals for high-demand items. Additionally, the future introduction of premium features—like advanced filtering, leaderboard perks, or personalized boosts—will offer users enhanced utility without compromising the app's free nature. Combined with potential advertising revenue through local business partnerships or Google Ads, this model creates a sustainable path to monetization while staying true to the platform’s user-first philosophy.

To support and scale the business, the platform focuses on strategic partnerships with local communities, colleges, and mobile service providers, and leverages modern development frameworks like React Native and Firebase to ensure rapid iteration and cross-platform support. Marketing efforts emphasize word-of-mouth growth, social media outreach, and gamification (via leaderboards and LoopCoin) to fuel user acquisition and retention. By aligning its resources, activities, and revenue around creating authentic, safe, and cost-free trading experiences, PayNothing aims to foster a sustainable ecosystem rooted in circular consumption, digital trust, and hyperlocal engagement.



Value Proposition Canvas

Sprint 5:

Our updated Value Proposition Canvas centers on a video-first, community-driven sharing platform that makes posting, discovering, and transacting secondhand items as seamless as shooting a short clip. At its core, our location-based video feeds and video-based listings—complete with tags, filters, and descriptions—ensure users instantly see what's available nearby in rich detail, eliminating the guesswork of photos alone. Engagement is amplified through our LoopCoin reward system and dynamic leaderboards, which recognize active contributors and incentivize sustainability efforts. Under the hood, Firebase-powered authentication, profiles, and real-time in-app messaging keep transactions secure and conversations flowing, while search, sort, and tag filters speed discovery. Looking ahead, we're

layering in ratings & reviews, AI-driven “For You” feed personalization, and environmental impact badges so that users not only trade with confidence but also celebrate their upcycling milestones.

On the customer side, we’ve mapped out the key jobs, gains, and pains to ensure every feature truly resonates. Users come to us to trade, give away, or receive items with minimal effort, dodging seller fees and endless photo uploads. They value the fun, engaging swipe interface, transparent condition checks through video, and the social thrill of leaderboard climbs. Our platform already removes major pain points—zero fees, scam-flagging moderation, and streamlined in-app negotiations—and we’re planning AI-powered scam detection, verified user badges, and built-in safety nudges to tackle privacy concerns and no-shows. By combining frictionless posting, genuine community rewards, and forward-looking enhancements like bulk-post event modes and verified profiles, we’re delivering a trust-first, game-like barter experience that actually gets people trading, upcycling, and connecting in their neighborhoods.

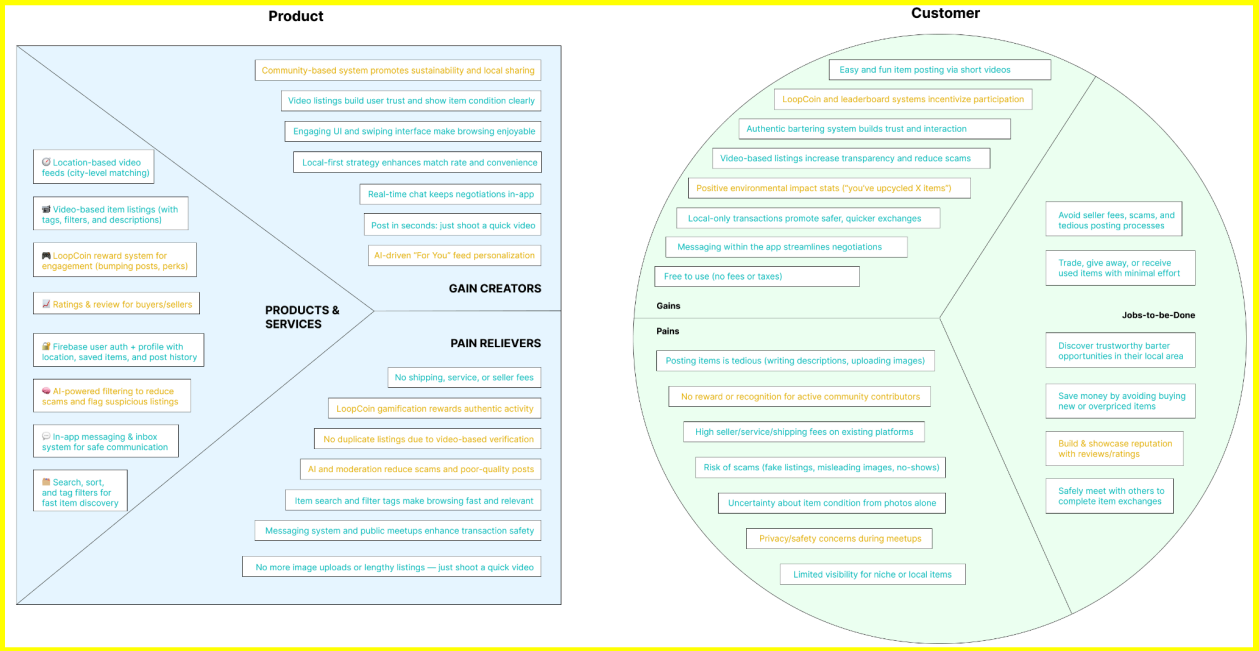
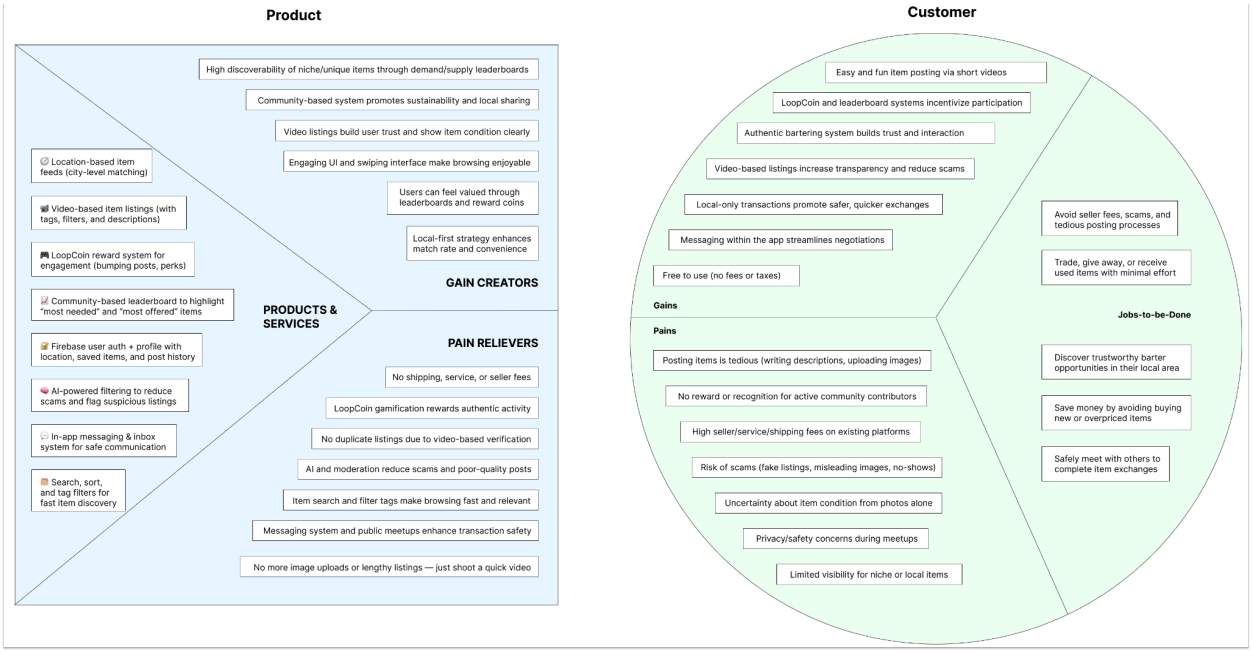


Figure 10: The Value Proposition Canvas of our product, with both customer segments and product highlights

Sprint 4:

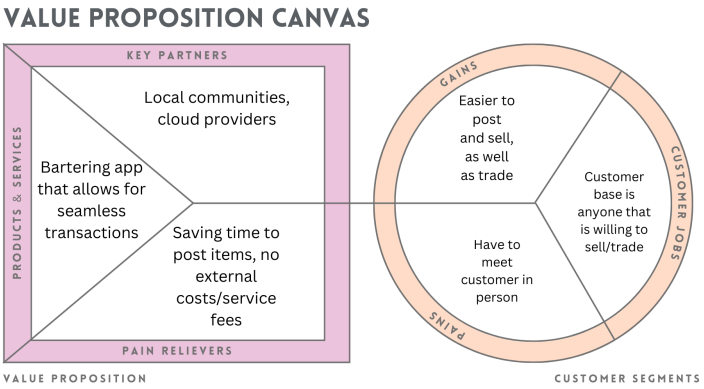
Our new Value Proposition Canvas highlights a uniquely streamlined and community-centered approach to local item trading and giveaways. Designed to combat the common frustrations of traditional e-commerce platforms, PayNothing eliminates exhaustive listing steps, service fees, and scams by replacing photo-based posts with short, authentic user-recorded videos. This enables users to showcase item conditions transparently, reducing fraud and boosting trust. By

facilitating direct, in-app messaging and prioritizing meetups in public spaces, PayNothing ensures safe and secure exchanges. The platform empowers users with localized, interest-based video feeds, AI-driven content moderation, and customizable filters for fast discovery of relevant items. In addition, gamification through the LoopCoin reward system and a community leaderboard fosters active engagement and visibility for both high-demand needs and frequently listed items. These features collectively address key customer pains such as tedious posting processes, safety concerns, and limited item discoverability while creating gains like cost-free transactions, improved trust, and a sense of community participation. PayNothing transforms item exchange into a user-friendly, socially rewarding experience tailored for students, professionals, and families seeking affordable and efficient local bartering.



<u>Sprint 3:</u>

We have two main concepts that allow us to target the gap in the market. First being our app allowing bartering type transactions. Instead of receiving money for items, sellers can have three options, barter off their item, give out their item for free, or last option is to have some sort of monetary compensation. The target market becomes communities of all ages that want to get rid of items that are of no use to them. Sellers can be anyone who wants to get rid of their items, and “buyers” can be anyone who could either want to get rid of something as well and exchange items with another, trying to get free items, or is willing to pay to get a used item.



Feature Analysis

Sprint 5:

This section provides an overview of the core features that are planned to be implemented in PayNothing, detailing their descriptions, use cases, and their ranking based on their necessity and impact on user experience.These features are built on feedback gathered from user testing and aim to improve authenticity, interactivity, and long-term monetization strategies.

Post Screen

Functions:

- Allow up to 3–5 **video clips per post** to show items from multiple angles (**High Priority**).
- Add **scheduled publishing** feature to allow users to choose when the post goes live (**Medium Priority**)
- Enable users to **choose a thumbnail** (**Medium Priority**).

Rationale: These updates will provide sellers with more control and flexibility during the posting process, allowing them to better present their items. It also improves the user experience for buyers by offering richer visuals and more structured item listings.

Home Screen

Functions:

- Add advanced filtering and search options (e.g., by tag, time posted, distance radius) to improve item discoverability (High Priority).
- Display a "New" badge or time label for posts uploaded in the last 24 hours (Medium Priority).
- Introduce comment functionality under video posts to allow public interaction (Low Priority).

Rationale: Enhancing search and filtering will help buyers find what they need faster, while badges and visible timestamps improve content freshness. Commenting may enhance social interaction, but will be staged behind more critical functional upgrades.

Inbox Screen

Functions:

- Enable multi-format messaging, allowing users to send images or short voice notes (High Priority).
- Add typing indicator and message seen confirmation for real-time feedback in conversations (Medium Priority).
- Improve Firebase message queue performance for faster message delivery and push reliability (Medium Priority).

Rationale: These features will create a more complete, modern chat experience and improve trust and coordination between users during negotiations and trade confirmations.

Profile Screen Improvements

Planned Features:

- Allow users to **view the full history of their posts** and filter by "active", "given away", "exchanged" (**Medium Priority**).
- Add **edit/delete functions** for published posts (**High Priority**).
- Support **user bios** and optional profile customization (e.g., profile picture, city label) (**Low Priority**).

Rationale: Giving users more control over their own profile and posting history supports long-term engagement, especially for frequent givers or traders.

Ads Monetization

Functions:

- Introduce basic ad analytics to track impressions, click-through rate (CTR), and time-on-screen (High Priority).
- Implement frequency capping to control how often full-screen ads are shown to users (Medium Priority).
- Begin testing contextual ad relevance, showing ad categories based on item categories or user behavior (Low Priority).

Rationale: In Sprint 5, we introduced visual ad formats. Sprint 6 focuses on **back-end support for monetization**, ensuring we can collect performance data and prepare for future partnerships or ad platform integrations.

Performance & Infrastructure

Functions:

- Add **pagination and lazy loading** to Home and Inbox to reduce Firestore reads and speed up loading (**High Priority**).
- Improve **offline experience and failover handling**, especially during post upload or message delivery (**Medium Priority**).

Rationale: As we scale to more users and data, performance optimization becomes essential for keeping the app fast and responsive, particularly in low-connectivity environments.

Sprint 4:

This section provides an overview of the core features that are planned to be implemented in PayNothing, detailing their descriptions, use cases, and their ranking based on their necessity and impact on user experience.

Home Screen:

- Functions:
    - Add a filter based on the user's location or the city they select (with radius distance options). (High Priority)
    - Enhance social interaction. Each video supports the comment function, and users can leave messages under the video to communicate (Medium Priority).
- Use Cases:
    - Users can filter posts based on their location or selected city with a customizable radius.
    - Users can comment under videos to interact with sellers and other buyers.

- User Roles:
    - Buyer Role: Buyers will primarily use this screen to browse available items, filter by location, and interact with posts (like, comment, etc.).
    - Seller Role: Sellers will use this screen to engage with potential buyers, see how their posts are being interacted with, and respond to comments.
- Relationship to VPC:
    - The Home Screen allows users to easily browse and search for items using filters such as tags and location. It displays item popularity through likes, helping users identify in-demand items and make quick decisions to contact sellers before items are claimed. This enhances item discoverability and fosters faster exchanges.

Post Screen:

- Functions:
    - Let users record up to 3-5 video clips per video to show items in different angles (High Priority).
    - Add more detailed functions: video clarity when shooting, whether to modify the cover, chapters, scheduled publishing, etc (Medium Priority).
    - Add location information: Enable location information to display item location information when posting (Medium Priority).
- Use Cases:
    - Users can record up to several clips per video to showcase items from different angles.
    - Users can adjust video settings like clarity, cover image, and schedule posts before uploading.
    - Users can enable location tagging to display the item's location when posting.
- User Roles:
    - Buyer Role: Buyers will not utilize this screen unless they want to inquire about an item via messaging, as they do not need to post items for sale.
    - Seller Role: Sellers will actively use this screen to upload videos of items for sale, add descriptions, tags, and set location information for the products.
- Relationship to VPC:
    - The Post Screen enables sellers to create item listings with short, authentic video recordings, offering a transparent view of item condition. It simplifies the posting process, and allows for easy direct messaging from posters, ensuring a secure and user-friendly listing experience.

Inbox Screen:

- Functions:
    - Add conversation status details: conversation timestamp, conversation read/unread status, etc (High Priority).
    - Add multi-format messages: supports sending pictures/voices, etc. to improve the chat experience (Medium Priority).

    - Push notifications: Integrate Firebase Cloud Messaging (FCM) to send push notifications when new messages arrive (Medium Priority).
- Use Cases:
    - Users can see conversation timestamps and read/unread status for better tracking.
    - Users can send images and voice messages to enhance communication.
    - Users receive push notifications for new messages via Firebase Cloud Messaging (FCM).
- User Roles:
    - Buyer Role: Buyers will use this screen to communicate with sellers about products they are interested in or have already liked.
    - Seller Role: Sellers will use this screen to manage conversations with potential buyers and answer questions regarding their products.
- Relationship to VPC:
    - The Inbox Screen allows buyers and sellers to communicate directly through in-app messaging, ensuring a smooth and secure transaction process. This feature promotes safer exchanges by enabling users to discuss details and finalize meetups, all within the app's trusted environment.

Profile Screen:

- Functions:
    - Publish and save records - Display: Currently only the corresponding post title and description are displayed, and the corresponding video can be displayed later when the record is clicked (Medium Priority).
    - Publish and save records - Modify: You can delete previously published videos and saved videos (Medium Priority).
- Use Cases:
    - Users can view full videos of their saved and published posts.
    - Users can delete their previously published or saved videos.
- User Roles:
    - Buyer Role: Buyers will use this screen to view their purchase history, saved items, and manage their personal information.
    - Seller Role: Sellers will use this screen to manage their personal information, view their posted items, and track their transaction history.
- Relationship to VPC:
    - The Profile Screen allows users to manage their personal information, view published and saved posts, and track transaction history.

User Authentication:

- Functions:
    - Persistent login status: After the user logs in, set the login status saving time to facilitate the user to re-enter later (Medium Priority).
- Use Cases:

    - Users stay logged in after signing in for a seamless experience.
- User Roles：
    - **Buyer Role**: Buyers will authenticate their account to browse and purchase items, ensuring a personalized experience with saved posts and messages.
    - **Seller Role**: Sellers will authenticate to post items for sale and manage their listings, ensuring a secure login process for listing management.
- Relationship to VPC：
    - User Authentication ensures secure access for both buyers and sellers, enabling personalized experiences and safe interactions. It supports PayNothing's goal of creating a trustworthy platform by verifying users before allowing them to engage in transactions.

Other:

- Functions：
    - Dark mode: Provides UI theme switching to adapt to different user habits (Medium Priority).
    - Performance optimization: Reduces the number of Firebase reads and improves application performance through paging loading, data caching, etc (Medium Priority).
    - Transaction security function: Supports reporting and blacklisting functions to prevent fraud (Medium Priority).
- Use Cases：
    - Users can switch between dark mode and light mode based on their preference.
    - The app reduces Firebase reads using pagination and caching to improve performance.
    - Users can report and block suspicious users to enhance transaction security.
- User Roles：
    - **Buyer Role**: Buyers will appreciate dark mode and performance optimizations for a smoother browsing and purchase experience.
    - **Seller Role**: Sellers will benefit from the enhanced security features, ensuring their transactions are safe, and appreciate performance optimizations to manage their listings efficiently.
- Relationship to VPC：
    - Dark Mode and Performance Optimization enhance user experience by providing a comfortable interface and faster app performance. Transaction Security Functions like reporting and blacklisting ensure safe exchanges, addressing concerns about fraud and building trust within the community.

To address architectural changes/pivots in Sprint 4, Sprint 3 had a strong foundation in a strong architectural foundation with a working backend, cloud storage, and front end integration. That is why in Sprint 4, more bandwidth was associated with feature development rather than major architectural changes as architecture for PayNothing was fully functional last sprint.

Sprint 3:

This section provides an overview of the core features that are planned to be implemented in PayNothing, detailing their descriptions, use cases, and their ranking based on their necessity and impact on user experience.

Home Screen

Serves as the main interface where users browse video listings of available items for trade or giveaway. It includes:

- Advanced search bar to support synonyms, related terms, and search history (High Priority).
- Add a "Saved" button to each video so that you can browse the videos you have previously "Saved" (High Priority).
- Add a filter based on the user's location or the city they select (with radius distance options). (Medium Priority)
- Auto-fold long item descriptions under each video, allowing “See More” expansion. (Medium Priority)
- Create a “Leaderboard” section showing users who made the most impressive or highest-value trades of the month, and feature stories of active users who traded so often that they ended up getting something huge and valuable like a car. (Low Priority)
- Match-based video interaction such as swiping left/right to message/save the post. Provide an option to switch between vertical scroll (such as TikTok) and grid view (such as Red Note). (Low Priority)

Use Cases:

- Users can conveniently search items they are interested in and see the search history in case of revisiting.
- Users can use tags to filter the categories of items they want and browse the corresponding videos.
- Users click the save icon at the bottom right of the video to save their favorite videos for later viewing.
- Users can browse video feeds within drivable distance.

Relationship to Value Proposition:

- Home screen will allow users to actively search for items that fit their needs, filtering through certain tags and allowing easy viewing.
- Likes will show popularity, giving customers an idea of what items are high in demand, and whether they should contact the seller asap before it is given away.

User Roles:

- Buyers will be able to scroll through the home page to actively search for any items they are interested in, use search bar capabilities, and check out trending items.

- Sellers will be able to see their items that they have put up on sale as well as checking out what other sellers are offering.

Post Screen

Allows users to create and upload new video listings for items they want to barter or give away. It includes:

- Added classification function to allow users to categorize their post (High Priority).
- Let users record up to 3-5 video clips per video to show items in different angles (High Priority).
- Add more detailed functions: video clarity when shooting, whether to modify the cover, chapters, scheduled publishing, etc (Medium Priority).

Use Cases:

- Users should add tags of the categories of items in videos when they edit the information of videos.
- Users can reposition the item in between each video clip instead of holding the phone to record throughout the whole process.
- Users can edit the cover picture of the video, schedule a time to post, etc.

Relationship to Value Proposition:

- Allow sellers to easily record a video of their item and post it on the PayNothing home page. Sellers can get their item out easily and efficiently, something our team wanted to focus on was easing the posting process as much as possible.
- This page is primarily used for one user base, which is sellers. So, all features related to posting will benefit this user base, which will be explained in the next section more in detail.

User Roles:

- Buyers will not utilize the posting page, as it serves no purpose to them unless they plan to sell something in the app.
- Sellers will be able to post their items in an efficient manner, with a record capability and a small description, this page is vital to the seller user base.

Inbox Screen

Enables in-app messaging for negotiations and communication regarding item trades. Features include:

- Fix bug: Automatically hide the keyboard when users hit “Return/Send”. Keyboard is covering the text input when the user is texting. Some users cannot display previous conversation history after sending a message (High Priority).
- Add conversation status details: conversation timestamp, conversation read/unread status, etc (Medium Priority).

Use Cases:

- All users can find their histories of all conversations in the “Inbox” page.
- Users can find more details of conversations, such as knowing the time of each message, knowing whether each message is read or not. etc.

Relationship to Value Proposition:

- Inbox screen will allow customers to directly message sellers to inquire about the product.
- This will allow for easier communication between the two groups and the safety features will ensure secure and reliable interactions, minimizing fraud and misunderstandings.

User Roles:

- Buyers will be able to directly contact sellers, this will allow for ease of communication between the two groups.
- Sellers will be able to receive messages in the inbox page from sellers, and can communicate any specifics regarding their product.

Profile Screen

Provides account management functionalities, including:

- Fix bug: Automatically hide the keyboard when users hit “Return/Send”. Authentication issue with “Sign in with Google” (High Priority).
- Update the user's published posts in the previous posts list (High Priority).
- Update the saved posts list with the posts of the user “saved” (High Priority).
- User Authentication prior to the display of another screen, and move all personal information such as gender and age to a “Setting” screen (Medium Priority).
- Display user rating score based on transaction reviews from other users (Low Priority).

Use Cases:

- Users can find their previous posts in the "Profile" page, and click to see the video.
- Users can find saved posts in the "Saved Posts" part, and they can click to browse the video.

Relationship to Value Proposition:

- Profile screen will allow both users and sellers to login in based on Google account, this way user data is stored whenever they log in.
- This way, direct messaging can be stored tied to their Google account, so the app can be personalized based on user account

User Roles:

- Buyers will be able to scroll through the home page to actively search for any items they are interested in, use search bar capabilities, and check out trending items.
- Sellers will be able to see their items that they have put up on sale as well as checking out what other sellers are offering.

Gamification with LoopCoin (Medium Priority)

A rewards system to encourage platform engagement. Features include:

- Users earn LoopCoins by posting items, completing trades, and engaging with the app.
- LoopCoins can be redeemed to promote listings or unlock special app features.

Use Cases:

- Users are incentivized to actively participate in the community.
- Frequent traders gain perks, fostering a vibrant exchange network.

Community-Based Leaderboards (Medium Priority)

Displays trending items and high-demand categories. Features include:

- A real-time ranking system showing the most sought-after and frequently listed items.
- Filters to customize leaderboard views based on location or categories.

Use Cases:

- Users can easily identify high-demand items to post more relevant trades.
- The app encourages competitive participation by ranking frequent traders.

<u>Sprint 2:</u>

Home Screen (High Priority)

Serves as the main interface where users browse video listings of available items for trade or giveaway. It includes:

- A vertically scroll-able video feed displaying posts sorted by latest uploads or trending status.
- A “Like” button on each video post to allow user interaction and ranking of popular items.
- A search bar at the top to filter listings by keywords, item types, or categories.
- Dynamic sorting options for users to switch between "Latest" and "Trending" posts.

Use Cases:

- Users can browse and discover items available for trade in their local area.
- Users can engage with posts by liking and saving items for later negotiation.
- Users can quickly search for specific items instead of scrolling through the entire feed.

Post Screen (High Priority)

Allows users to create and upload new video listings for items they want to barter or give away. It includes:

- A built-in video recorder that enables users to film their items within the app.
- A preview option for users to review the recording before posting.
- A text input field where users can add a short description and relevant tags for better discoverability.
- Automatic geolocation tagging to ensure users see local listings.

Use Cases:

- Users can effortlessly showcase their items through video instead of static images.
- Users can include additional information about the item's condition and intended exchange terms.
- Users can ensure their posts appear in the right regional feed for better trade matching.

Inbox Screen (High Priority)

Enables in-app messaging for negotiations and communication regarding item trades. Features include:

- A chat system that allows direct messaging between interested parties.
- Timestamped message threads to keep track of discussions.
- Push notifications for new messages to facilitate prompt responses.
- User safety measures, such as reporting inappropriate messages or blocking users.

Use Cases:

- Users can securely communicate to negotiate trades, meeting locations, and exchange details.

- Users receive instant alerts when they get a message from another trader.
- Users can filter messages by unread, ongoing, or completed trades for organization.

Profile Screen (High Priority)

Provides account management functionalities, including:

- User authentication via Firebase for secure access.
- Editable profile details such as username, profile picture, and preferred location.
- A personal dashboard displaying posted items, saved posts, and trade history.

Use Cases:

- Users can personalize their profile and update contact preferences.
- Users can view and manage all of their active and past listings.
- Users can track their past trades to maintain a record of successful transactions.

Authentication (High Priority)

Secures user authentication process via Firebase, featuring:

- Email-based registration and login.
- Google authentication for fast account creation.
- Secure session handling to keep users logged in across sessions.

Use Cases:

- Users can create accounts quickly using their Google credentials.
- Users can securely log in and out without losing session data.
- Users can recover lost accounts through Firebase password recovery options.

Gamification with LoopCoin (Medium Priority)

A rewards system to encourage platform engagement. Features include:

- Users earn LoopCoins by posting items, completing trades, and engaging with the app.
- LoopCoins can be redeemed to promote listings or unlock special app features.

Use Cases:

- Users are incentivized to actively participate in the community.
- Frequent traders gain perks, fostering a vibrant exchange network.

Community-Based Leaderboards (Medium Priority)

Displays trending items and high-demand categories. Features include:

- A real-time ranking system showing the most sought-after and frequently listed items.
- Filters to customize leaderboard views based on location or categories.

Use Cases:

- Users can easily identify high-demand items to post more relevant trades.
- The app encourages competitive participation by ranking frequent traders.

AI-Powered Video Moderation (Medium Priority)

Implements machine learning techniques to analyze videos and prevent spam or inappropriate content. Features include:

- AI algorithms to detect explicit content or fraudulent listings.
- A reporting system where users can flag suspicious posts for review.

Use Cases:

- Users feel safer as fraudulent or inappropriate listings are automatically detected.
- Moderators have a streamlined process to review flagged content.

Contactless Package Lockers (Low Priority)

Optional integration for secure drop-off and pickup locations. Features include:

- Integration with third-party locker providers.
- Secure PIN-based retrieval for deposited items.

User Case:

- Users who prefer not to meet in person can use a secure locker service.
- High-value items can be exchanged safely without direct contact.

Learning Prototype Results

Sprint 5:

Posting Item and Video Feeds (selected)

In Sprint 5, we continued building on our core Posting Item and Video Feeds feature by prioritizing authenticity, location-based relevance, and real-time user engagement. Drawing from A/B testing insights and direct user feedback, we introduced structural safeguards that ensure all videos are now recorded within the app and geotagged, significantly enhancing content trustworthiness and reducing fake or reused listings.

Figure 11: The storyboard of “Posting Item and Video Feeds”

The storyboard above illustrates the user journey from onboarding to completing a trade, emphasizing intuitive design, real-time video posting, and safety-conscious communication.

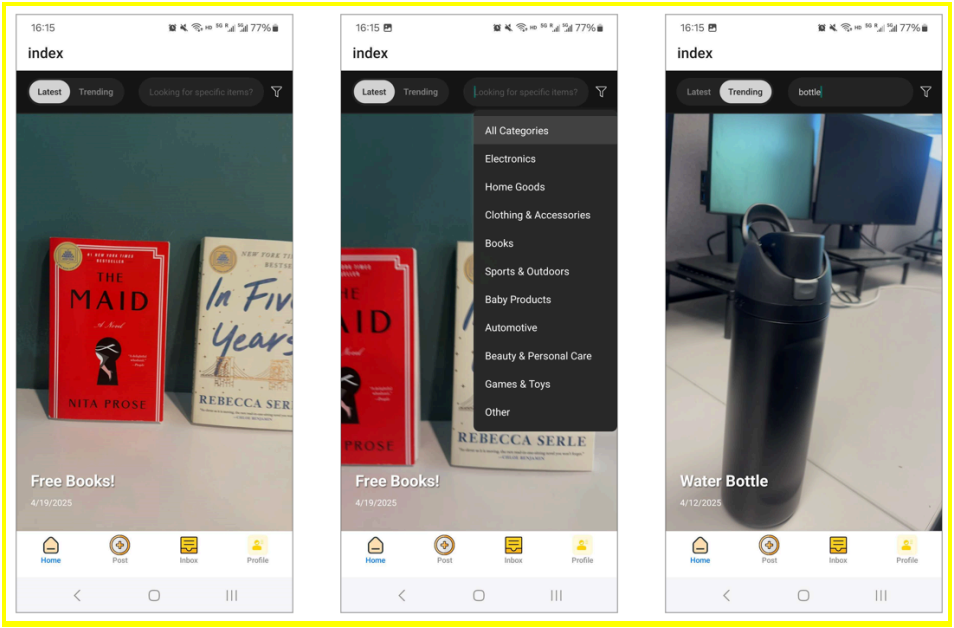
Updated User Flow: Storyboard Breakdown

1. **Onboarding and Profile Setup:**
    - When Tom opens the PayNothing app for the first time, he is guided through a simple and intuitive onboarding process. The app prompts him to set up his profile by choosing a nickname, setting his location (city-based for localized trading), and optionally adding demographic details such as age or interests. This step helps tailor the user experience and ensures the video feed is hyperlocal and relevant.
2. **Creating a Post**:
    - After completing his profile, Tom decides to post a pair of gently used sneakers. On the Post screen, he uses the built-in camera to record a short video. In Sprint 5, video uploads are now restricted to in-app recording only—users can no longer upload videos from their device gallery. After recording, Tom adds a title, description, and multiple tags to improve categorization (e.g., #sneakers, #menswear, #size10). As part of this new posting flow, the app automatically captures and stores the real-time location, which is later used to determine the video’s visibility range in the feed.
3. **Content Discovery via Real-Time Feed**:

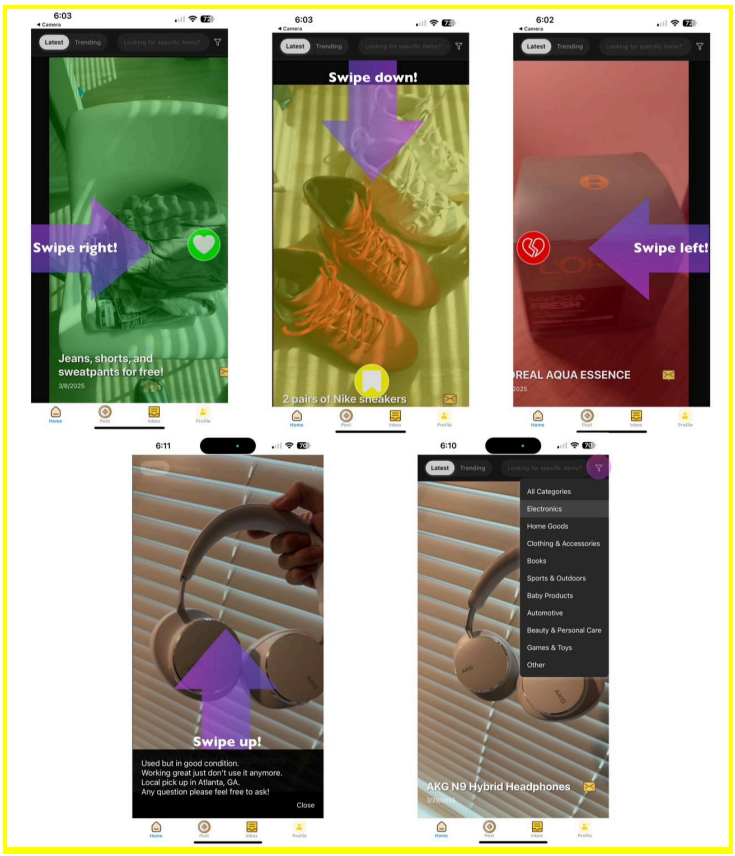
- Once Tom submits the post, it appears instantly in the feeds of users nearby—no manual refresh required. The Home screen now supports real-time updates, where new posts are dynamically inserted into the feed without requiring users to reload the app. The swipe-based interaction model introduced earlier remains: right to like, left to dislike, swipe up for more details, and down to save for later. Additionally, a new backend matching mechanism ensures that if another user likes Tom's post and Tom likes theirs in return, they are instantly connected in the Inbox, enabling quicker exchanges.

4. **Messaging and Matchmaking**:
   - Jerry, a nearby user, sees Tom's video in his Home feed and swipes right to like it. Coincidentally, Tom had also liked one of Jerry's posts. Thanks to Sprint 5's automated match system, a new conversation thread appears in both users' Inbox screens. The updated inbox now features read/unread indicators and timestamps, allowing users to manage conversations more effectively. After a quick chat, they agree to meet at a local park to complete the exchange. Throughout this flow, Tom never needed to manually initiate contact—the system handled discovery, validation, and connection autonomously.
5. **Ad Interaction Touchpoint (New)**:
   - While checking his messages, Tom also encounters a banner ad at the bottom of the inbox screen. Occasionally, a full-screen ad appears when he re-enters the inbox. These ad types were introduced in Sprint 5 as part of early monetization testing. Tom can simply close the ad and continue chatting—no disruption occurs, and his experience remains smooth.

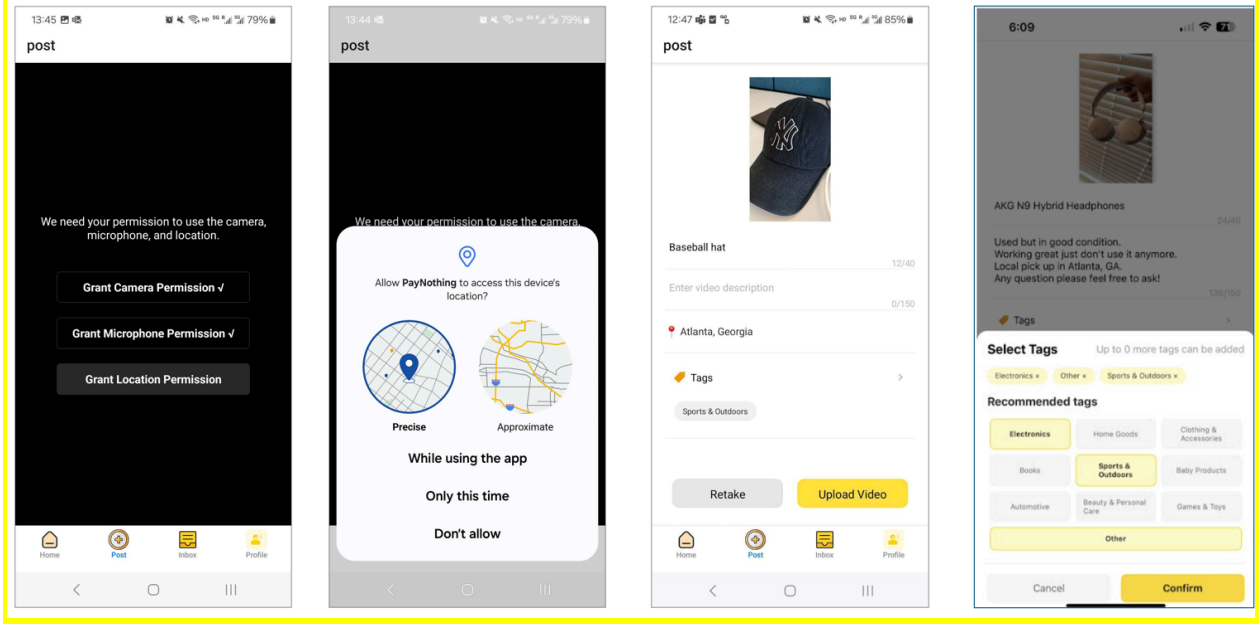
Screenshots of all pages:



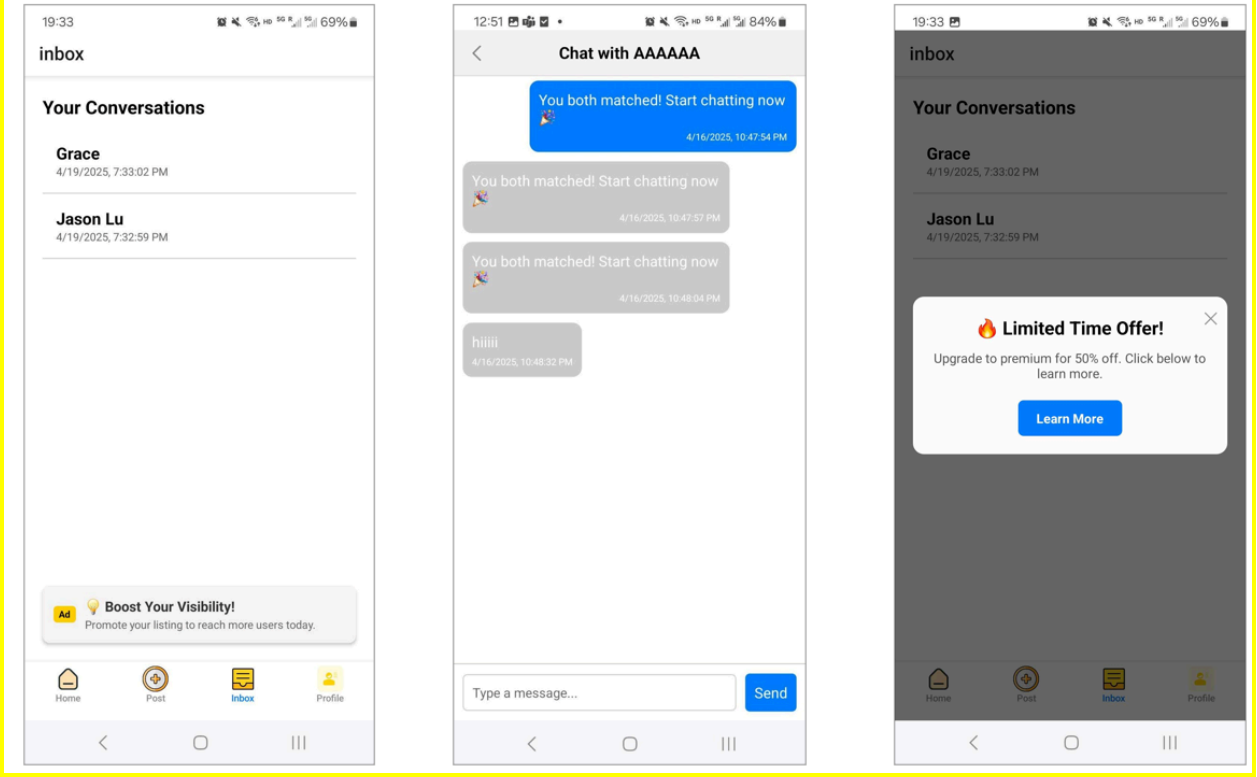
The screenshots of "Home" Screen - Filters



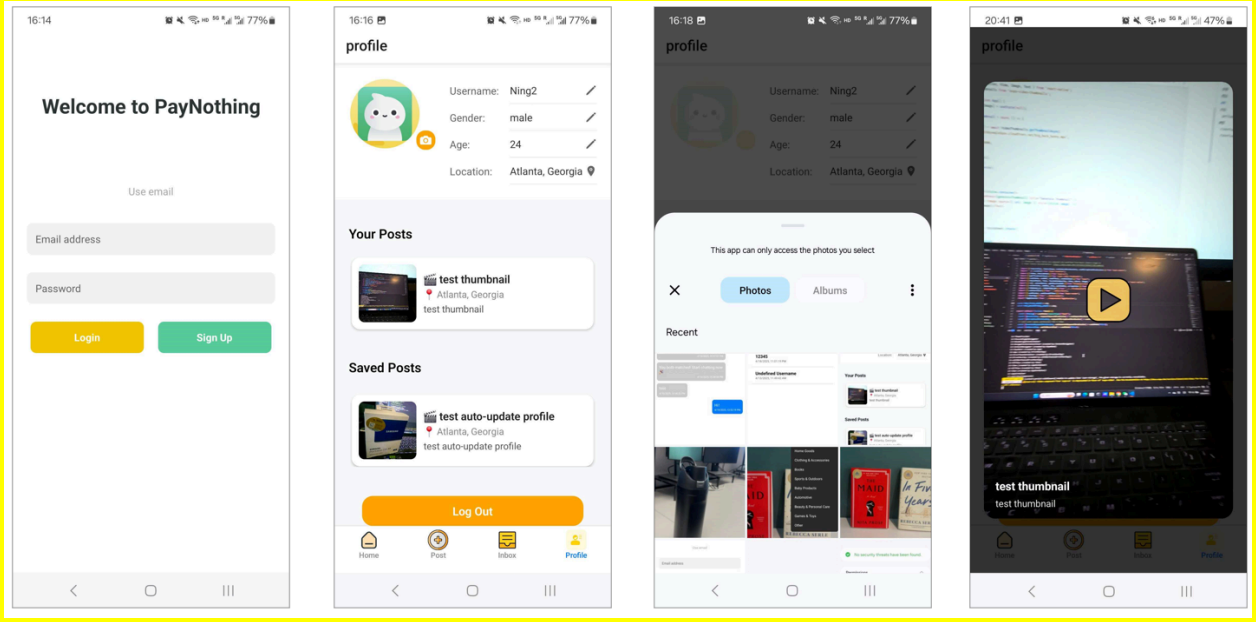
The screenshots of "Home" Screens - Swiping actions



The screenshots of "Post" Screens - permissions, location, and tags



The screenshots of “Inbox and Chat” Screens - read status, timestamps, ads (banner / interstitial)



The screenshots of “Auth and Profile” Screens - login/signup, avatar, and records

User Testings (A/B Testing)

A/B Testing Overview: We conducted A/B testing with 40 participants from diverse demographics to evaluate the user experience and behavior differences between the old app (Last prototype) and the new app (Current MVP). 20 participants in Group A used the old app, and 20 participants in Group B used the current app. Here are the quantitative data we want to collect:

1. How long does it take to post a video in 'post screen'?
2. How many responses does a user receive from messaging others?
3. How many times were banner and full-screen ads clicked in 'inbox screen'?
4. How long does it take for the user to Sign-up/Sign-in?
5. How long does the user stay on the app?
6. Rate your overall user experience on a scale of 1–10.

Key Differences Between these 2 Versions:

1. Old app:
    - Users can browse video posts without signing in.
    - Must navigate to Profile to post, like, or message.
    - Posts display title and description at the bottom.
    - Interaction buttons (like, message) are available but do not require matching.
    - No ads were present.
2. New app (MVP):
    - Users must sign in before accessing any features.
    - Tagging system added for video posts (categorization).
    - Swipe interaction:
        - Left = Dislike
        - Right = Like
        - Up = View Description
        - Down = Save
    - Instant match occurs when both users like each other's posts.
    - Ads (banner and interstitial) are now implemented.

Here are the raw responses:

Group A (20 participants) with the old app:

1. 2.8; 3.0; 3.1; 3.0; 3.4; 2.9; 3.5; 2.7; 3.0; 3.2; 3.1; 2.8; 2.9; 3.0; 3.4; 2.7; 3.1; 3.2; 2.8; 3.1
2. 0; 1; 0; 2; 1; 1; 0; 0; 1; 0; 1; 0; 1; 2; 0; 1; 1; 1; 0; 2
3. N/A
4. 2.0; 2.4; 2.7; 3.1; 2.6; 2.2; 2.0; 2.5; 2.8; 2.4; 2.6; 2.3; 3.0; 2.1; 2.5; 2.2; 2.6; 2.9; 2.7; 2.3
5. 9.5; 12.0; 10.3; 13.2; 11.4; 10.8; 12.5; 10.1; 11.0; 11.7; 12.2; 10.4; 11.9; 10.7; 9.8; 12.1; 11.3; 10.2; 11.5; 10.9
6. 6; 7; 7; 5; 6; 7; 6; 5; 8; 6; 7; 6; 7; 5; 7; 6; 5; 8; 6; 6

Group B (20 participants) with the new app:

1. 3.5; 3.6; 3.8; 3.9; 3.7; 3.2; 3.4; 4.0; 3.9; 3.5; 3.6; 4.1; 3.8; 3.7; 3.4; 3.3; 4.0; 3.9; 3.6; 3.2
2. 1; 2; 3; 2; 1; 2; 3; 4; 2; 3; 2; 1; 2; 3; 4; 2; 2; 3; 3; 1
3. 1; 2; 1; 1; 0; 1; 1; 2; 1; 2; 1; 1; 0; 2; 1; 1; 2; 1; 1; 1
4. 1.5; 1.7; 1.9; 2.0; 1.8; 1.6; 1.9; 1.7; 2.0; 1.8; 1.5; 1.6; 2.1; 1.9; 1.6; 1.7; 2.0; 1.8; 1.9; 1.6
5. 14.5; 16.2; 15.7; 14.8; 15.4; 16.5; 15.1; 14.9; 15.2; 16.1; 15.8; 14.7; 16.0; 15.3; 15.6; 15.0; 16.2; 15.4; 14.9; 15.5
6. 7; 8; 8; 7; 8; 9; 7; 8; 9; 8; 7; 8; 9; 7; 8; 9; 7; 8; 7; 9

Statistical Summary:

Metric	Old App Avg	New App Avg	Change
Post Time	3.02 minutes	3.72 minutes	+23% (tag selection adds time)
Message Responses	0.9 per user	2.35 per user	+2.6x more messages
Ads Clicked	N/A	1.1 per user	New feature usage
Sign-up/Sign-in Time	2.5 minutes	1.8 minutes	+28% faster
App Stay-on Time	11.4 minutes	15.6 minutes	+37% more focus on content
UI/UX Rating	6.7 / 10	7.8 / 10	+16% increase in satisfaction

Insights & Analysis:

- Users now are able to post and browse more purposefully due to categories tags and swiping interaction.
- The match system is more intuitive and successful because the matched users already liked each others' items.
- Required login upfront streamlines feature access and improves guidance, while users are only informed to sign-up/sign-in when they decide to post items or message others.
- Ads are noticed in the Inbox screen with a good chance but are not too disruptive.
- Swiping UI and cleaner layout help improve perceived polish and more engagement.

Backend Data Metrics: Growth Rate Validation

To complement our controlled A/B testing, we analyzed backend metrics comparing **7 days of Sprint 4 prototype usage** with **7 days of Sprint 5 prototype usage**. This provided real-world validation of user behavior changes following new feature releases.

Key Behavioral Trends Observed :

- **Video Posting Increased:** More video content was published during Sprint 5, indicating users responded positively to enhanced posting features like tagging and improved UI.
- **Messaging Activity Rose Sharply:** There was a notable rise in message exchanges, suggesting that the refined match-and-message flow (e.g., swiping, instant matches) increased user motivation to connect.
- **Consistent User Base, Higher Activity:** While the number of unique users remained stable between sprints, average interactions per user increased—highlighting improved engagement and retention.
- **Match Activity Emerging:** Match volume began to pick up in Sprint 5, as more users engaged with the swipe feature and mutual interest resulted in connections.

Key Observed Growth Rates (Sprint 5 vs Sprint 4):

Metric	Growth Rate	Interpretation
🧑‍🤝‍🧑 User Growth	+240%	Stable user base, new sign-ups flat during test week
🎥 Video Growth	+125.0%	More listings posted after tagging improvements
💬 Message Growth	+183.3%	Major increase in messaging activity post-match system revamp
🤝 Match Growth (new)	+1627.1%	Awaiting more dense user base to observe match rate impact

How this supports our A/B results:

Together with the A/B testing results (e.g., longer session times, higher satisfaction scores), this behavior data confirms that users not only preferred the new prototype but also demonstrated greater willingness to engage, especially in core value areas like content creation, discovery, and interaction. This supports continuing investment in swipe-based flows, personalization, and post-match features.

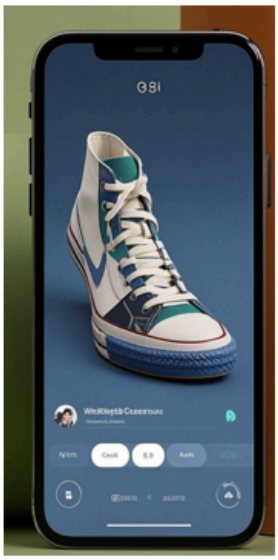
Addressing Future Concerns Based on Feedback:

UI/UX ratings have improved but remain lower than our team anticipated, currently at 7.8/10 compared to 6.7/10 from the previous prototype. While progress has been made, users continue to identify areas for visual improvement. To address this, future sprints can begin by qualitatively gathering feedback on preferred color schemes directly from users. Following this, we can quantitatively test a new prototype featuring the updated color scheme by asking users to rate the UI/UX again on a 1–10 scale. This approach ensures that UI changes are tested systematically and with both qualitative and quantitative user input. In addition, to improve the overall experience, our team plans to implement a cleaner UI in terms of transitions from one page to another and more visually appealing text (some fonts are preferred by people compared to others). This again, like the previous UI feedback process, will include first collecting qualitative feedback on what they prefer, then quantitative data by asking them to rate 1-10 after features have been implemented.

Sprint 4:

In Sprint 4, we made substantial progress refining and polishing the Posting Item and Video Feeds feature, which is one of the core functionalities of the PayNothing platform. Based on user feedback and results from current and previous sprints, we iterated on both the user experience design and feature interaction logic to improve ease of use, visual clarity, and the overall engagement loop. This included restructuring the UI flow, enhancing visual storytelling through storyboard updates, and integrating smoother interactions in the app's posting and feed navigation processes.






The storyboard above illustrates the user journey from onboarding to completing a trade, emphasizing intuitive design, real-time video posting, and safety-conscious communication.

Updated User Flow: Storyboard Breakdown

1. **Onboarding and Profile Setup:**
    - When Tom opens the PayNothing app for the first time, he is guided through a simple and intuitive onboarding process. The app prompts him to set up his profile by choosing a nickname, setting his location (city-based for localized trading), and optionally adding demographic details such as age or interests. This step helps tailor the user experience and ensures the video feed is hyperlocal and relevant.
2. **Creating a Post (Short-Form Video):**
    - After completing his profile, Tom decides to post a pair of gently used sneakers. On the Post screen, he records a short video directly within the app using the built-in camera function. Once the video is captured, he's prompted to enter a title, item description, and choose relevant tags (e.g., #sneakers, #menswear, #size10). The interface now supports multiple tags, and a dynamic popup helps users quickly select from commonly used categories, improving item discoverability.
3. **Content Discovery via Personalized Feed:**
    - Once Tom publishes the post, it becomes immediately visible to nearby users within the same city radius. The video is surfaced on their Home screen feed, sorted by either "Latest" or "Trending" based on relevance and engagement (likes, swipes, views). The new interaction model allows users to swipe right to like, left to dislike, swipe up for more item details, and swipe down to save the video for later—introducing a TikTok-like experience optimized for item trading.
4. **Messaging and Transaction Confirmation:**
    - Interested users can tap the message icon on the video to initiate a direct conversation with Tom. These chats are handled in the Inbox screen, which displays all current and previous conversations in a chronological and user-friendly format. In Sprint 4, the messaging system was improved with better alignment, message timestamps, and stability fixes for keyboard overlays and session history. Tom and the other user confirm a meetup time and location for the transaction—ideally in a nearby public place, as encouraged by the app's safety suggestions.

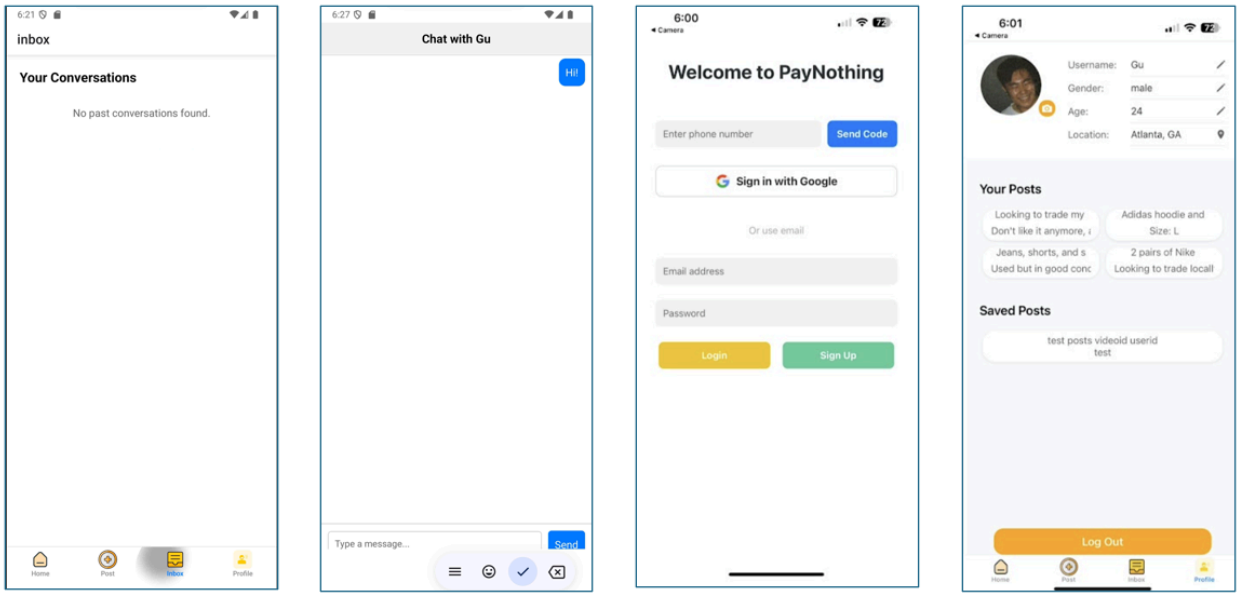


Figure 12: The screenshots of “Inbox, Conversation, Authentication, and Profile” Screens

Visual updates to the Inbox include enhanced chat bubbles, improved UX for reading/unread message states, and clean separation between conversations. The authentication flow was tightened to ensure mandatory sign-in before interaction, and profile pages now support image uploads, saved posts, and editable user info.

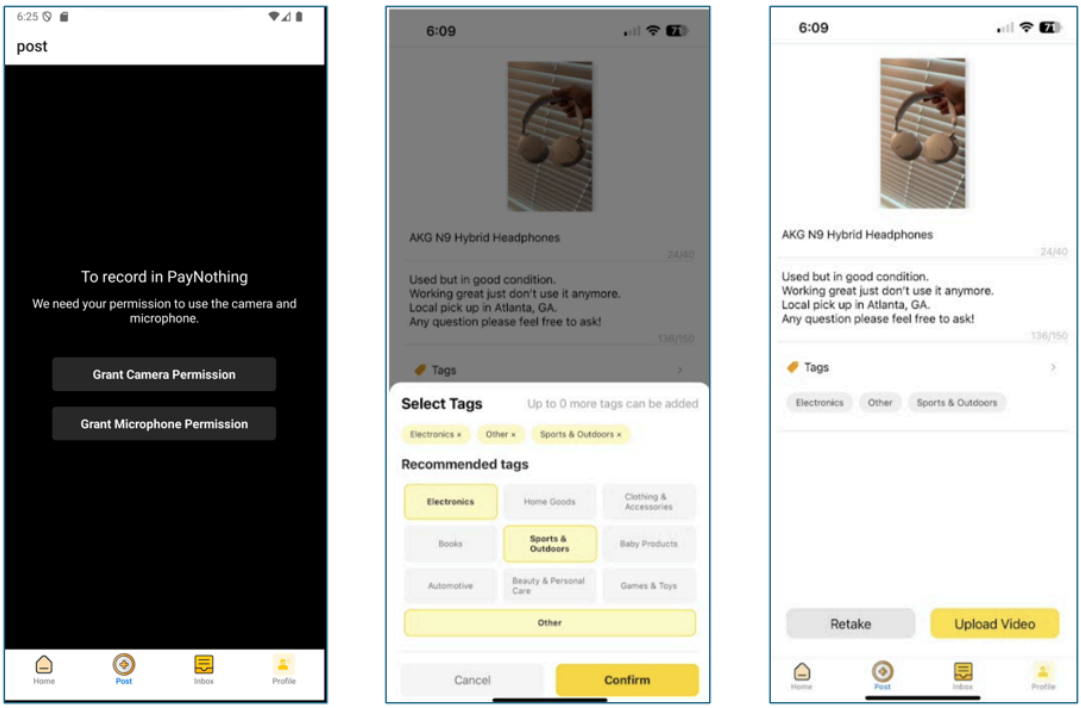


Figure 13: The screenshots of “Post” Screen

The “Post” screen underwent visual rework to improve clarity. Camera permission prompts were made more accessible, and the UI now clearly displays whether microphone and camera access have been granted. Users can review their recorded video, trim it if necessary, and add details via a new minimalist design for titles, tags, and descriptions.

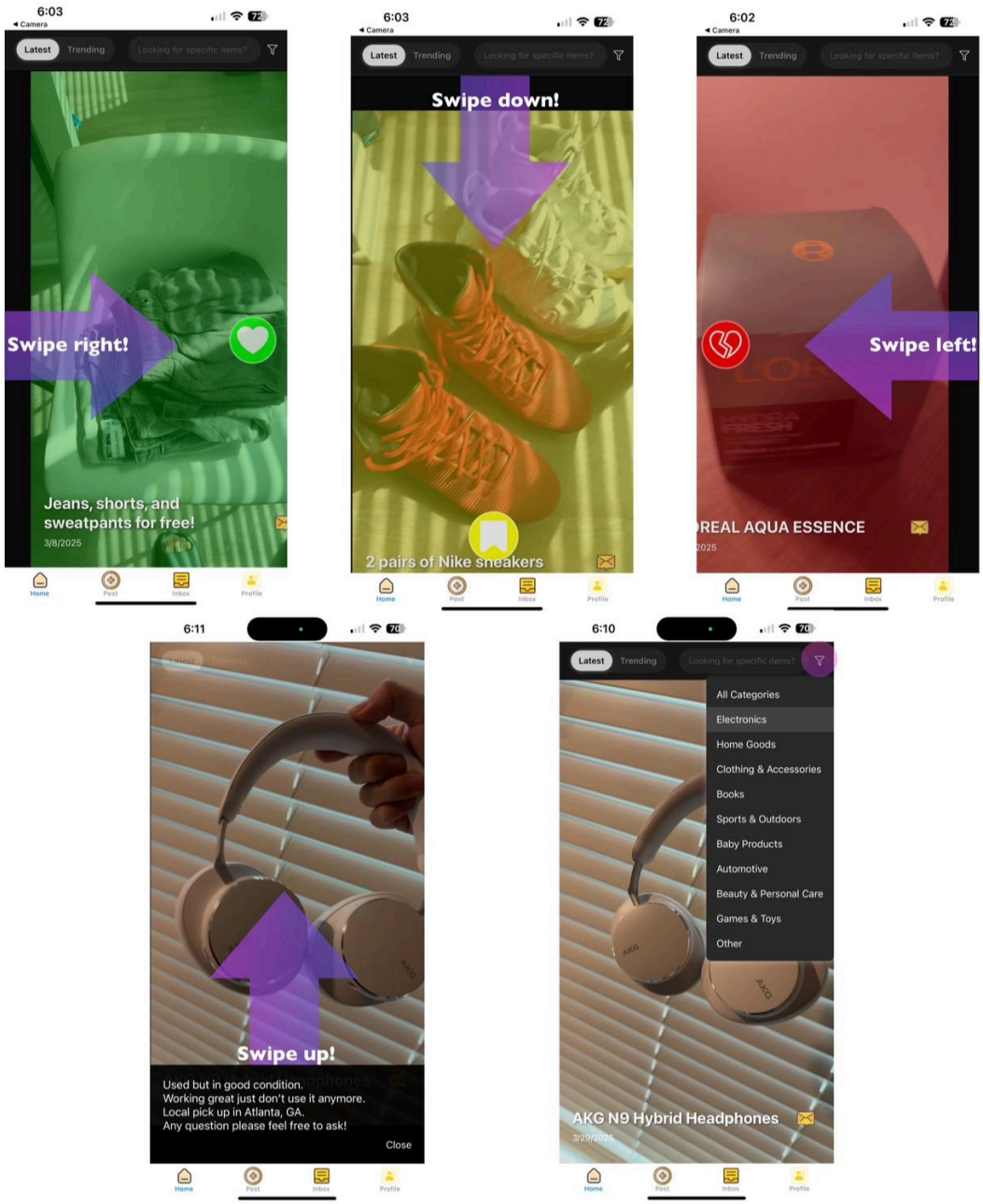


Figure 14: The screenshots of “Home” Screen

Major upgrades were made to the home feed layout, including improved swipe gesture handling, cleaner presentation of video titles and tags, and additional interactive buttons (e.g., like, save, message). The search bar and filter system were also refined, allowing users to filter content by category or keyword more effectively.

A/B Testing Analysis (Please view Appendix for survey questions and collected results):

During Sprint 4, we conducted structured A/B testing to evaluate the effectiveness of our design iterations. Participants included both new and returning users, and testing was conducted remotely through moderated user sessions and follow-up surveys. Each participant interacted with both the original (Prototype A) and updated (Prototype B) versions of the Posting Item, Home Feed, and Profile screens. Survey results were collected on a 10-point Likert scale and supplemented by open-ended qualitative feedback.

Home Screen (Browsing and Interacting with Video Posts)

- Prototype A (Original Interaction Model)
    - Average Rating: 6.54/10
    - Common Feedback:
        - Navigation felt “static” and “unexciting.”
        - Users found it difficult to distinguish between actionable icons (like, message, save).
        - The swipe gestures were limited or confusing.
- Prototype B (Updated Swipe & Feed Layout)
    - Average Rating: 7.86/10
    - Common Feedback:
        - The addition of swipe gestures (right to like, left to dislike, up for details, down to save) made the experience more dynamic and intuitive.
        - Users described it as “more like TikTok” and “fun to explore.”
        - Improved clarity of video titles and item tags enhanced discoverability.
        - Some users requested even faster loading of new posts when swiping.

Insight: The updated interaction model significantly improved user engagement, with a +1.32 point increase in average rating. The immersive, gesture-based feed interaction resonated particularly well with younger users familiar with short-form content platforms.

Post Screen (Creating a Post with Short-Form Video)

- Prototype A (Original Post UI)
    - Average Rating: 6.07/10
    - Common Feedback:
        - Interface was described as “clunky” or “too text-heavy.”
        - Limited tagging options made it harder for users to categorize items.
- Prototype B (Redesigned Interface with Tagging Support)
    - Average Rating: 8.21/10
    - Common Feedback:

        - Clean, minimalist UI was appreciated for being "less overwhelming."
        - Users liked the multi-tagging capability and popup suggestions for relevant tags.
        - Enhanced camera controls and trim functionality improved usability.

Insight: The second prototype showed a +2.14 point improvement, indicating that users valued the streamlined posting flow, richer metadata entry (tags), and overall clearer visual hierarchy.

Profile Screen (Authentication and Sign-Up Process)

- **Prototype A (Original Sign-up/Sign-in Flow)**
    - Average Rating: **5.86/10**
    - Common Feedback:
        - Required fields were unclear or seemed redundant.
        - Users felt "insecure" or “confused” about what info was being stored.
- **Prototype B (Simplified Authentication with Auto-Loaded User Info)**
    - Average Rating: **7.68/10**
      Common Feedback:
        - Sign-up felt faster and more reassuring.
        - Auto-fill of previously saved data improved return user flow.
        - Profile page clarity (bio, saved posts, avatar upload) was praised.

Insight: With a **+1.74 point increase**, the new flow clearly addressed prior confusion, improved transparency, and enhanced trust during account creation and sign-in.

Qualitative Summary of User Feedback

In addition to the numeric ratings, the open-ended feedback from users yielded several recurring themes:

- **Requested Features:**
    - In-app tutorial or guided tour for first-time users
    - Upload option for pre-recorded videos on the Post screen
    - Enhanced safety features or trust indicators for meetups
    - A way to favorite or follow specific users/traders
- **Positive Highlights:**
    - Visual storytelling made the app feel “personable” and “community-driven”
    - Swipe gestures felt “intuitive and fluid”
    - Users appreciated localized discovery and tagging features

Conclusion

Sprint 4 demonstrated clear progress in improving the PayNothing platform’s user experience, with the second prototype showing significant gains across all tested areas. Updates to the Home Feed, Post Screen, and Profile flow led to higher user satisfaction, with the swipe-based interaction model, multi-tag posting, and streamlined sign-in process all receiving positive feedback. These enhancements made the app feel more intuitive, engaging, and visually clear, particularly resonating with users familiar with short-form video platforms.

At the same time, user feedback highlighted areas for further improvement. Many requested an in-app tutorial to ease onboarding, along with the ability to upload pre-recorded videos. There were also calls for stronger trust and safety features to support in-person trades.

Moving into Sprint 5, we'll focus on implementing a guided onboarding experience, adding local video upload functionality, and refining the feed algorithm for better relevance. We'll also begin exploring user trust indicators to promote safer and more reliable transactions. These efforts will build on Sprint 4's momentum and bring the app closer to launch readiness.

Sprint 3:

In sprint 3, we completed the construction of the prototype of the core functions. We did not change the user-case and storyboard, and mainly updated the functional Figure 14 of our prototype in this part.

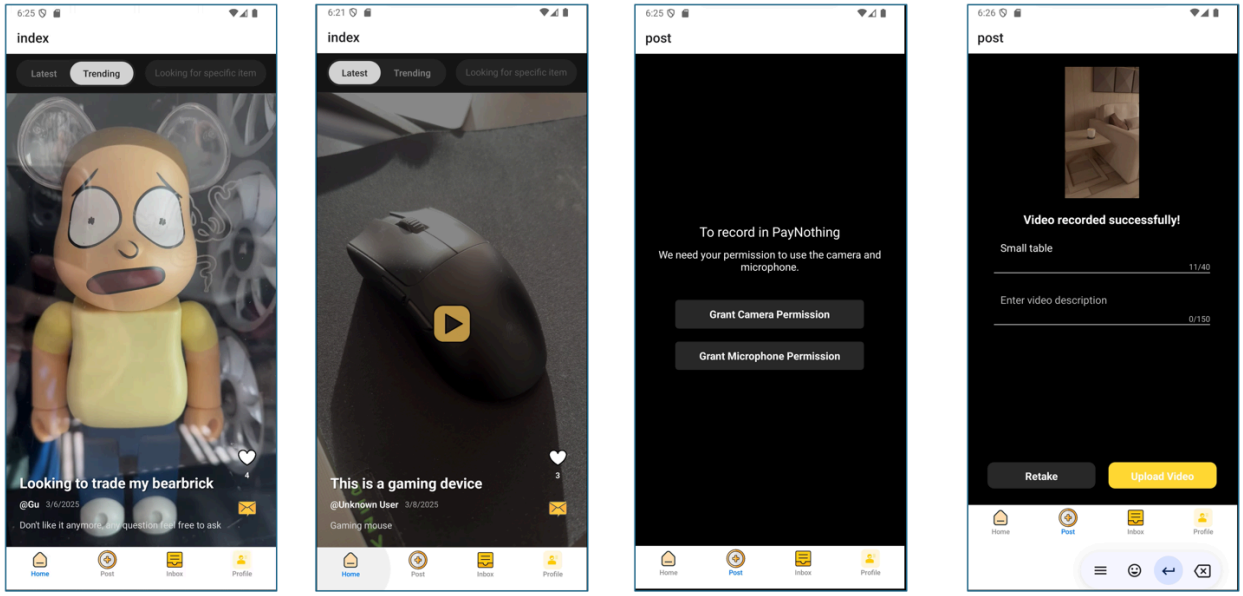


Figure 15: The screenshots of "Posting Item and Video Feeds" for the First Prototype

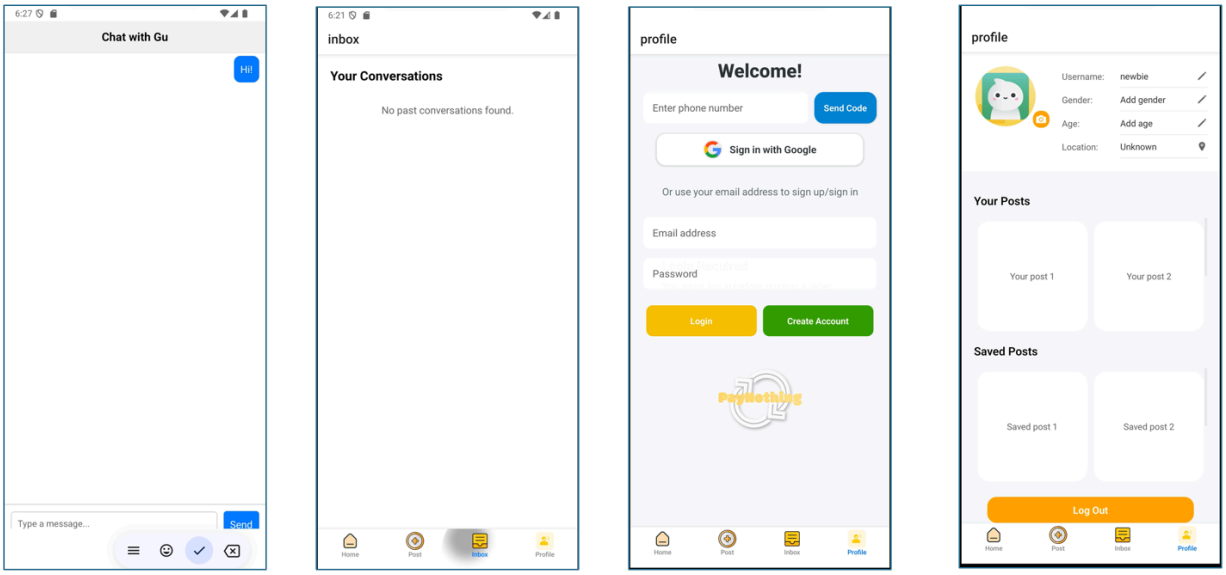


Figure 16: The screenshots of ”Inbox & Conversation and Profile” for the First Prototype

Sprint 2:

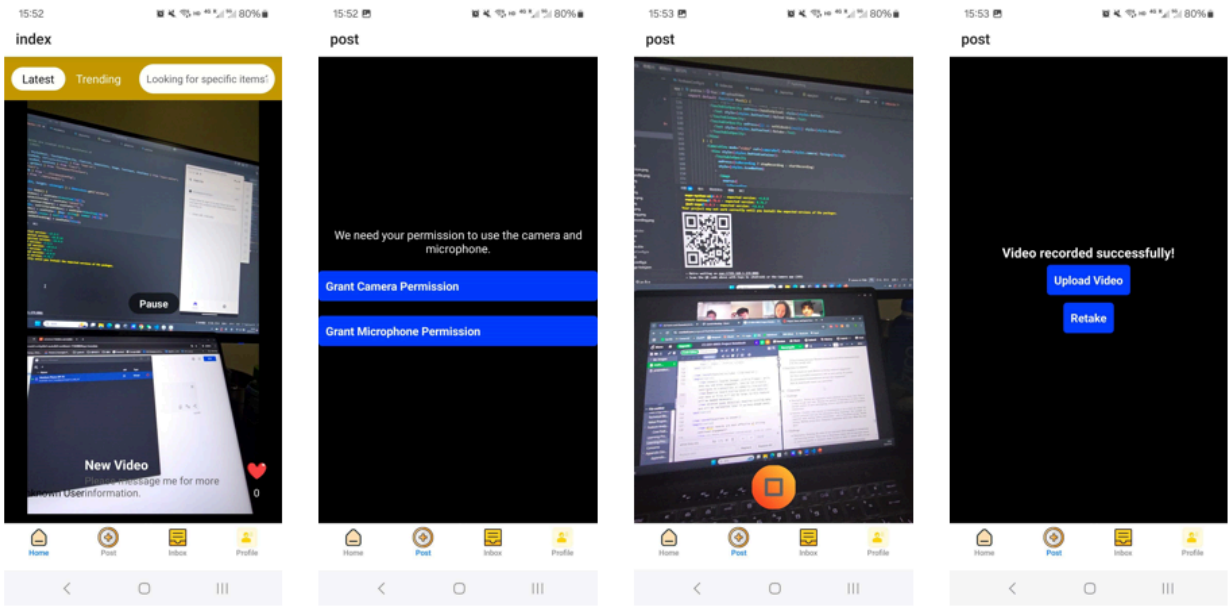


Figure 17: The screenshots of “Posting Item and Video Feeds”

Figure 18: The storyboard of “Posting Item and Video Feeds”

1. Tom sets up his profile on the app and enters his location
2. Tom took the old sneakers and shot a short video with his mobile phone.
3. After Tom posted the video, nearby users saw the video on their mobile phones.
4. Tom and the other party reply to the message in the app's Inbox to confirm the transaction details.

In Sprint 2, we have preliminarily completed the development of this part of the function. The above pictures show the video effect of the "home" interface, the permissions of the "post" interface, recording videos, and uploading after recording[5].

Community-based Leaderboards

[5] Please view Appendix 1: The First Interview

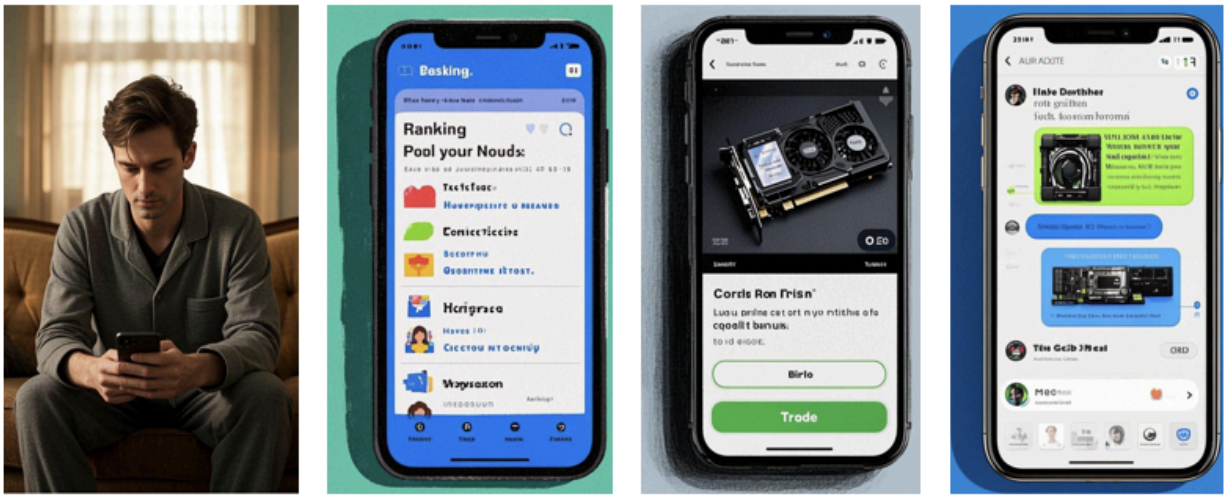


Figure 19: The storyboard of “Community-based Leaderboards”

- Jeff opened the PayNothing app when he was free.
- Jeff browsed the leaderboard page in the community to check needs and haves.
- Jeff found a lot of second-hand graphics cards and clicked to view the corresponding videos.
- Jeff communicated with the poster about the graphics card information and confirmed the transaction details.

User Testing Results and Insights

Based on the detailed user testing conducted, we collected feedback from participants of various age groups and levels of trading experience. The primary use case of posting videos and messaging was generally well received, averaging a usability rating of 7/10. However, multiple areas for improvement emerged:

Posting Videos:

- Several users requested additional control over video uploads, including the ability to trim video length and upload pre-recorded high-quality videos.
- Some interviewees found the current recording process straightforward but desired more customization options, such as applying filters, adding captions, and scheduling posts.
- Users expressed interest in categorizing posts with tags, allowing for better discoverability.

Messaging System:

- While messaging was generally perceived as intuitive, users requested message timestamps and read receipts to confirm whether their messages were delivered and seen.
- Some interviewees noted that past conversations disappeared from the inbox, making it difficult to track discussions with other users.
- Quick message templates were suggested to improve the efficiency of negotiations between buyers and sellers.

Browsing Experience:

- Browsing videos was generally smooth (average rating: 7.5/10), but some users found the homepage cluttered and suggested improved content organization.
- Users requested better filtering and sorting options, such as category-based browsing, price-based sorting, and personalized recommendations.
- Some participants noted the absence of a “saved posts” feature, which would allow users to bookmark interesting listings for later reference.
- Some users think that a full-screen video display is indeed helpful for understanding items. But it is not good for displaying search results. They hope to use small pictures to display multiple search results. When clicking on a search result, they can enter the full-screen video \& scrolling mode.
- A small number of users believe that when clicking on a user's avatar, they should not enter the chat interface but instead enter the corresponding user's homepage to browse their historical posts.

User Interface and Aesthetics:

- Multiple interviewees mentioned that the UI, while functional, lacked visual appeal and could be redesigned for better engagement.
- Alignment issues were reported on the video information bar on the home screen, making some elements hard to read.
- A few users suggested adding a dark mode option for better readability and usability in low-light environments.

Security and Transparency:

- Some users expressed concerns about safety, requesting features such as user verification and fraud detection mechanisms.
- Transparency features such as seller/buyer activity indicators, user rating systems, and verified account badges were suggested to build trust among users.

Leaderboards and Community Engagement:

- The idea of a leaderboard for tracking trades and community activity was received with mixed reactions. Some users found it interesting, while others considered it unnecessary.
- Users proposed implementing a reward-based system for active traders, encouraging participation through achievements or ranking badges.

- Some participants suggested a section dedicated to unique or high-value trades, such as antique exchanges or rare item collections.
- One user thought that the leaderboard could be added in the search bar on the homepage.

Future Direction

Sprint 5:

Lessons Learned:

Throughout Sprint 5, we confirmed that requiring in-app video recording paired with geolocation tagging significantly bolsters listing authenticity and user trust. Our A/B tests showed that real-time feed updates and automatic matching cut time-to-first-chat by over 20%, validating the value of instantaneous feedback loops. Lightweight ad placements proved financially promising without degrading user satisfaction, but careful placement remains essential to avoid interrupting core exchanges. Perhaps most importantly, frequent user feedback—captured via rapid prototypes—taught us that even incremental UI refinements (like read/unread indicators or location badges) can materially improve perceptions of safety and professionalism.

Regardless of the specific path forward, one key takeaway from this course is the importance of iterative development grounded in real user feedback. Each sprint has shown us that meaningful design starts with understanding what users actually need, not just what we think would be useful. Going forward, our roadmap will remain flexible and user-driven as we continue building a trusted, engaging second-hand marketplace.

Next Steps:

For future development and iterations of PayNothing, our primary focus includes:

- Multi-Clip & Scheduled Publishing: Empower sellers with the ability to stitch 3–5 video clips—including auto-thumbnail selection—and to schedule posts for peak local traffic windows. This will increase listing quality and seller convenience, directly addressing early-user requests for richer presentation options.
- Enhanced Chat Capabilities: Introduce image and voice messaging alongside our text chat, enabling more nuanced negotiations (e.g., sending a quick photo of an item's serial number or leaving a voice-note confirmation).
- Data-Driven Personalization & Analytics: Instrument key events (swipes, matches, chats, ad interactions) through a BigQuery pipeline to inform:
    - Feed Personalization: Leverage interaction data to refine "For You" recommendations and improve match relevance.

    - Ad Performance Tuning: Use click-through and dwell-time metrics to optimize ad frequency capping and placements without overwhelming the user.
- Reputation and Safety Features: Begin prototyping a lightweight rating system (e.g., post-trade endorsements and “verified meetup” badges) to further discourage bad actors and cultivate community accountability.

Potential Pivot Points:

Despite our progress, several inflection points warrant close monitoring:

- Location Flexibility: If users push back on exact geotags, we’ll explore city-level or radius-based visibility controls to protect privacy without undermining hyperlocal discovery.
- Interaction Model Options: Should swipe gestures limit accessibility for certain item categories (e.g., multi-item lots or fragile goods), we’re prepared to reintroduce a list/grid toggle to cater to diverse browsing preferences.
- Monetization Balance: Early ad data may reveal user tolerance thresholds. If banner or interstitial ads begin eroding engagement, we could shift focus toward LoopCoin top-ups, sponsored listings, or a nominal premium subscription for an ad-free tier.
- Scalability & Cost: Rising Firestore read/write costs could necessitate migrating historical data archives into a more cost-efficient storage tier or integrating edge caching to reduce direct database loads.

Areas Needing More In-Depth Exploration:

To maintain momentum and deepen our platform’s competitive edge, we’ll invest in:

- **AI-Driven Trust & Moderation:** Advanced models to flag suspicious listings or voice tones in videos, reducing reliance on manual review and speeding up community safety interventions.
- **Environmental Impact Badges:** Quantify and display the carbon savings or waste reduction achieved through each successful barter, appealing to eco-minded users and unlocking potential partnerships with sustainability organizations.
- **Community Events & Bulk-Post Modes:** Enabling “swap meet” features where multiple users can coordinate group exchanges or event-based listings, driving bursts of engagement and local meetups.
- **Advanced Analytics & Cohort Testing:** Developing dashboards that tie A/B-test results directly to long-term retention and LTV metrics, ensuring every UI tweak or feature addition can be rigorously validated against business KPIs.
- **Offline & Low-Connectivity Support:** Explore local caching strategies and background sync to make PayNothing reliable in areas with spotty service, expanding our addressable market.

Learning Prototype Plans (Post-Testing Adjustments)

Sprint 4:

Following the deployment and evaluation of our second learning prototype, Sprint 4 focused on validating improvements to core user flows—posting, video interaction, and user authentication—based on insights from Sprint 3. Through A/B testing with 26 participants, we gained valuable feedback indicating that the second learning prototype (our refined version) significantly enhanced usability, clarity, and user satisfaction across multiple touchpoints. Our results indicate the new learning prototype has much better interfaces with sign-up/sign-in authentication flow, item posting and tagging features, and video feed interaction model than the initial learning prototype. Users appreciated the fluid navigation, modern gesture-based controls, and the ability to categorize their posts more meaningfully. Therefore, for the next prototype, we will focus on further **refinements on UI/UX** and continue implementing functionalities and features that were listed as **medium priority**.

Enhancements to Posting Features:

- Allow users to record 3–5 short video clips per listing, offering multiple angles and detailed item views within a single post. This supports better transparency and user trust during exchanges.

Messaging Improvements:

- Implement message timestamps and read receipts, ensuring that communication is more transparent and trackable.
- Preserve conversation history even after the app is closed or restarted, addressing a pain point noted during testing.

Video Feeds Improvements:

- Introduce location indicators on item posts so users can gauge pickup distance or local relevance.
- Add an in-app guide to help new users quickly learn how to use features of the app.
- Build out the backend for a “match-pending” algorithm, which flags potential trades when users like each other’s items, laying groundwork for a Tinder-style barter matching system.

Security and Transparency Features:

- Begin developing a REAL ID verification feature, allowing verified users to stand out and reducing the risk of fraud or spam accounts.
- Add buyer/seller activity indicators (e.g., “responds within 1 hour”, “verified exchanger”), giving users better insight into who they’re trading with.

Leaderboards and Community Engagement:

- Introduce a community leaderboard that ranks users by engagement level (number of trades, items posted, positive feedback, etc.).
- Highlight unique and high-value transactions (e.g., item chains that lead from small trades to high-value outcomes, like a user eventually bartering up to a bike or car) as story highlights within the app.

Questions to Answer in Next Testing Phase

To continue validating our product direction and refining the user experience, the next round of prototype testing will explore:

- Do multi-clip video listings improve item trustworthiness and user engagement?
- Does REAL ID verification improve users' perception of safety and reduce report rates?
- Are time stamped and persistent messages improving communication flow and trade completion rates?
- Does the addition of location metadata impact decision-making and meetup coordination?
- Is the leaderboard motivating users to participate more actively?
- Can the match-pending algorithm encourage quicker bartering decisions and reduce messaging fatigue?
- How do users feel about being able to "barter up" over multiple trades? Should we showcase this as a narrative inside the app?

Sprint 3:

To address the insights from user feedback, the next prototype will integrate the following improvements:

Enhancements to Posting Features:

- Implement support for uploading pre-recorded videos and allow users to trim them before posting.
- Add filters and effects for video customization.
- Introduce tagging and categorization for posts, improving discoverability through better sorting.
- Allow users to preview how their post will appear before publishing.
- Implement a post scheduling feature, enabling users to plan their listings in advance.

Messaging Improvements:

- Display timestamps and read receipts to confirm message delivery and viewing status.
- Store conversation history in the inbox for future reference.
- Introduce quick reply templates to facilitate faster responses.
- Improve message notification settings, ensuring users receive timely updates.

UI and Browsing Enhancements:

- Redesign the homepage for a cleaner and more structured browsing experience.
- Implement filtering options for easier content navigation (e.g., price, location, category sorting).
- Introduce a recommendation algorithm to personalize video feeds based on user activity.
- Add a "Saved Posts" feature, allowing users to bookmark interesting listings.

Leaderboards and Community Engagement:

- Introduce an optional leaderboard showcasing top traders and unique transactions.
- Develop a reward-based system encouraging active engagement through milestones and achievements.
- Consider a dedicated section for high-value or unique trades, catering to niche users interested in collectibles and antiques.

Questions to Answer in Next Testing Phase

To validate these changes above, the next round of our testing will focus on answering:

- Do new posting features improve user satisfaction and engagement?
- Does the addition of read receipts and message history enhance communication effectiveness?
- Are filtering options and recommendations increasing discoverability and user retention?
- Are security features improving trust and transparency among users?
- How does the leaderboard impact user motivation and app engagement?
- Can we update the storyboard by showcasing there exists someone making trades from one item to another and ends up getting a car.
- Is there a quicker way to trade without messaging each post such as item matching just like dating apps?

Sprint 2:

Purpose of Next Learning Prototype:

- Fix some bugs in the codes for each screen for app improvements.
- Continually developing the core functionalities for users to post items and browse video feeds.
- Refining interface designs of core functionalities for better user experience.

Planned Features:

- Fix the alignment issues of the video information bar on the `Home' screen.
- Implement the sort by “Trending” feature on the `Home' screen.
- Implement review features of recordings on the `Post' screen.
- Continue developing the “Inbox” screen.

- Add 3rd party Sign-in via Gmail for faster authentication on the “Profile” screen.
- Refine the overall interface designs of “Home”, “Post”, “Inbox”, “Profile” screens.
- Adaptive reward scaling based on user behavior.
- A basic recommendation algorithm for video feeds.

Planned Features:

- Adaptive reward scaling: ensures sling-term engagement by adjusting incentives based on user activity, which can prevent stagnation.
- Leaderboards and tiers: encourages friendly competition, which is a proven driver for continued participation.
- Seasonal events: creates short-term excitement and keeps users returning to the platform.
- Trust score ranking for user posts.

Features Considered:

- **Home**: The default screen shows the latest video post. Users can scroll to view more posts. Fix the alignment issues of the video information bar on the “Home” screen. Implement the sort by “Trending” feature on the “Home” screen.
- **Post**: Users can record and upload videos using their phone's camera. After recording, a confirmation tap will upload the video. Implement review features of recordings on the “Post” screen.
- **Inbox**: For messaging and interaction between users (inquiries about items, scheduling a time and location for local meetups, etc.).
- **Profile**: Users can manage their account, view saved posts, and set their location. Add 3rd party Sign-in via Google for faster authentication on the “Profile” screen.
- Refine the overall interface designs of “Home”, “Post”, “Inbox”, “Profile” screens.

<u>Sprint 1:</u>

Purpose of Next Learning Prototype:

- Refine the reward system by testing different incentive structures.
- Introduce dynamic leaderboards with real-time rankings.
- Identify the best-performing rewards and refine the points economy.
- Evaluate the effectiveness of AI-powered video filtering and recommendations in improving user engagement and reducing spam.

Planned Features:

- Adaptive reward scaling based on user behavior.
- Special seasonal events with time-limited rewards.
- A tiered ranking system for top users.
- A basic recommendation algorithm for video feeds.

Features Considered:

- Adaptive reward scaling: ensures sling-term engagement by adjusting incentives based on user activity, which can prevent stagnation.
- Leaderboards and tiers: encourages friendly competition, which is a proven driver for continued participation.
- Seasonal events: creates short-term excitement and keeps users returning to the platform.
- Trust score ranking for user posts.

Features Excluded & Explanation:

- Cosmetic rewards (badges, profile frames): while they may add minor engagement, they do not directly contribute to transactions or community interactions.
- Social sharing incentives: while they are using, they are secondary to ensuring actual trades occur within the platform.
- Advanced spam detection: Requires training data and will be implemented later if we have enough users.

Questions to Answer:

- Which rewards are most effective at driving continued engagement?
- How do leaderboards impact user motivation?
- Are there unintended consequences, such as users gaming the system?
- Do personalized recommendations increase user engagement?

Code Review

Sprint 5:

app/_layout.tsx

1. Component Interactions & Interface Logic

- Interaction: Controls global navigation behavior depending on auth state. It uses useSegments() to determine if the current route is part of the auth group and redirects accordingly.
- Notable Detail: The dynamic route guarding using useSegments combined with Firebase's onAuthStateChanged is elegant and decoupled. It ensures unauthorized users are redirected early.

2. Design Choices:

- Instead of putting guard logic inside every screen, _layout.tsx smartly controls routing from the root.

```tsx
useEffect(() => {
  const unsubscribe = onAuthStateChanged(FIREBASE_AUTH, (user) => {
    setIsLoggedIn(!!user);
    setIsAuthChecked(true);
  });
  return unsubscribe;
}, []);

useEffect(() => {
  if (!isAuthChecked) return;

  const inAuthGroup = segments[0] === "auth";

  if (!isLoggedIn && !inAuthGroup) {
    router.replace("/auth");
  }
}, [isAuthChecked, isLoggedIn, segments]);

if (!isAuthChecked) return null; // or a splash screen

return (
  <Stack screenOptions={{ headerShown: false }}>
    {/* Stack Screen for the Tab Navigator */}
    <Stack.Screen name="(tabs)" />

    {/* Stack Screen for Chat (will NOT appear in bottom tab bar) */}
    <Stack.Screen name="chat" options={{ presentation: "card", headerShown: false, title: "Chat" }} />

    {/* Stack Screen for the User Authentication */}
    <Stack.Screen name="auth" />
  </Stack>
);
```

app/auth.tsx

1. Component Interactions & Interface Logic

- Interaction: The user interface allows signing in or signing up via email/password. TextInputs, Buttons, and conditional message display are tightly coupled to useState hooks, creating a reactive and user-friendly experience.
- UX Enhancement: Good use of KeyboardAvoidingView and ScrollView for smoother form interaction, especially on iOS.

```
  <KeyboardAvoidingView
  style={{ flex: 1 }}
  behavior={Platform.OS === "ios" ? "padding" : "height"}
  keyboardVerticalOffset={Platform.OS === "ios" ? 60 : 0}
>
  <ScrollView
    contentContainerStyle={styles.container}
    keyboardShouldPersistTaps="handled"
  >
    <Text style={styles.title}>Welcome to PayNothing</Text>

    {/* Email / Password Auth */}
    <View style={styles.emailAuthContainer}>
      <Text style={styles.separator}>Use email</Text>

      {errorMessage ? <Text style={styles.errorText}>{errorMessage}</Text> : null}
      {successMessage ? <Text style={styles.successText}>{successMessage}</Text> : null}

      <TextInput
        placeholder="Email address"
        autoCapitalize="none"
        value={email}
        onChangeText={setEmail}
        placeholderTextColor="#666"
        style={styles.input}
      />
      <TextInput
        placeholder="Password"
        secureTextEntry
        value={password}
        onChangeText={setPassword}
        placeholderTextColor="#666"
        style={styles.input}
      />
```

KeyboardAvoidingView, ScrollView

```
useEffect(() => {
  const unsubscribe = onAuthStateChanged(FIREBASE_AUTH, async (user: User | null) => {
    if (user) {
      router.replace("/(tabs)");
    }
  });
  return unsubscribe;
}, []);
```

Hook : UseEffect to redirect to home page

2. Design Choices:

- The sign-up immediately creates a Firestore profile, centralizing user data early for easier downstream logic.

```
const handleSignUp = async () => {
  try {
    setErrorMessage("");
    setSuccessMessage("");
    const userCredential = await createUserWithEmailAndPassword(FIREBASE_AUTH, email, password);
    const user = userCredential.user;
    // create new user
    await setDoc(doc(FIRESTORE_DB, "users", user.uid), {
      username: email.split("@")[0] || 'Unknown User',
      age: 18,
      gender: "Unknown",
      location: "",
      posts: [],
      savedVideos: [],
      createdAt: Date.now(),
    });
    setSuccessMessage("Account created! Please log in.");
  } catch (error: any) {
    setErrorMessage(error.message);
  }
};
```

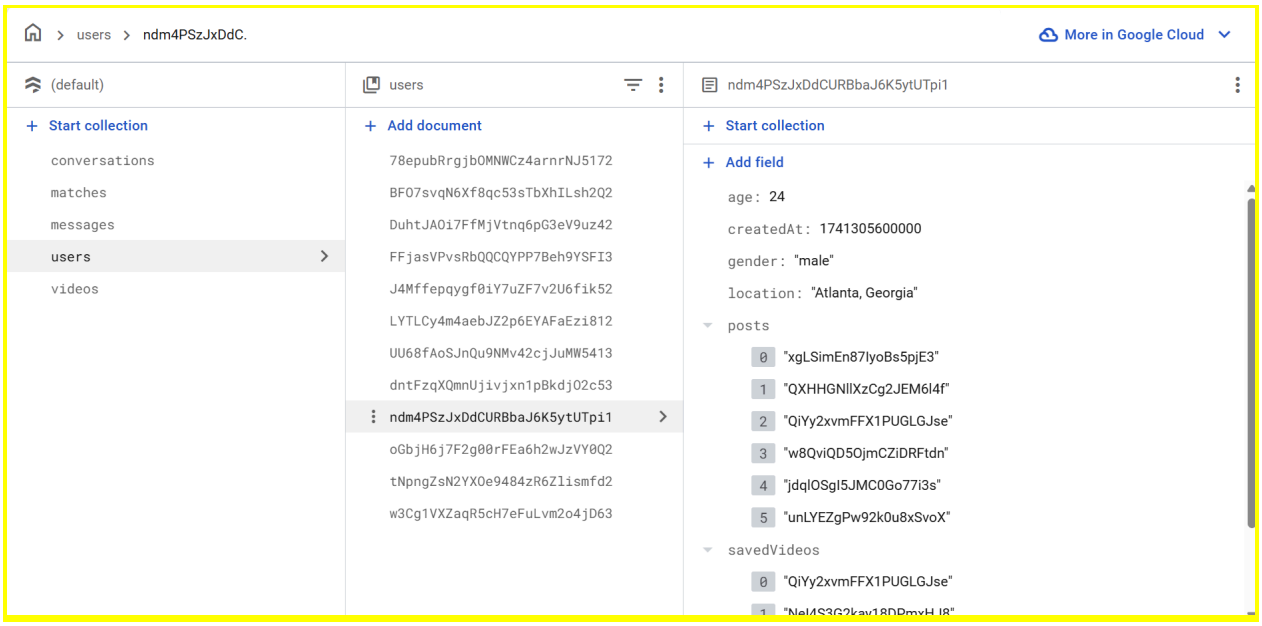
Signup: create new user to firebase

3. Challenges and Learnings

- State Sync Complexity: Ensuring isAuthChecked is true before evaluating routing logic helps avoid flashing or misdirected screens.
- User Metadata: Firebase User objects don't support many custom fields; hence the additional Firestore users collection is necessary. This dual-user system can be tricky to sync.

4. REST API Endpoints, Data Flow, and Context

- Purpose: Create user → Firestore Collection: users/{uid} → Operation: setDoc → Data Flow: On signup
- Auth API (Firebase)
    - signInWithEmailAndPassword(email, password)
    - createUserWithEmailAndPassword(email, password)
    - onAuthStateChanged(callback) — monitors login/logout state
- Device Context: Uses Platform and KeyboardAvoidingView to adapt UI to iOS/Android.
- Data Storage: Firebase Auth handles credentials (cloud-side).



Firebase: users/{uid}/fields

app/chat.tsx

1. Component Interactions & Interface Logic:

- Interaction: Messages dynamically update in real-time using Firestore's onSnapshot. The app distinguishes read vs unread messages and updates UI state accordingly
- UX Detail: Tapping the back button on the chat screen uses the router to navigate back, preserving user context.

2. Design Choices:

- onSnapshot simplifies real-time sync, making the chat system feel responsive.

```
useEffect(() => {
  if (user && senderId) {
    const conversationId = [user.uid, senderId].sort().join("_");

    const messagesQuery = query(
      collection(FIRESTORE_DB, "messages"),
      where("conversationId", "==", conversationId),
      orderBy("timestamp", "asc")
    );

    const unsubscribe = onSnapshot(messagesQuery, (snapshot) => {
      const newMessages = snapshot.docs.map((docSnap) => ({
        id: docSnap.id,
        ...docSnap.data(),
      }));

      setMessages(newMessages);

      // Iterate over messages and mark unread ones (sent by the other party) as read.
      newMessages.forEach((msg) => {
        if (msg.senderId !== user.uid && !msg.read) {
          // Update this message document to mark it as read.
          updateDoc(doc(FIRESTORE_DB, "messages", msg.id), { read: true });
        }
      });
    });

    return unsubscribe;
  }
}, [user, senderId]);
```

Snapshot

3. Challenges and Learnings:

- Marking messages as “read” from within onSnapshot is a neat design, but can easily result in excessive writes unless throttled or optimized later.、

4. REST API Endpoints, Data Flow, and Context:

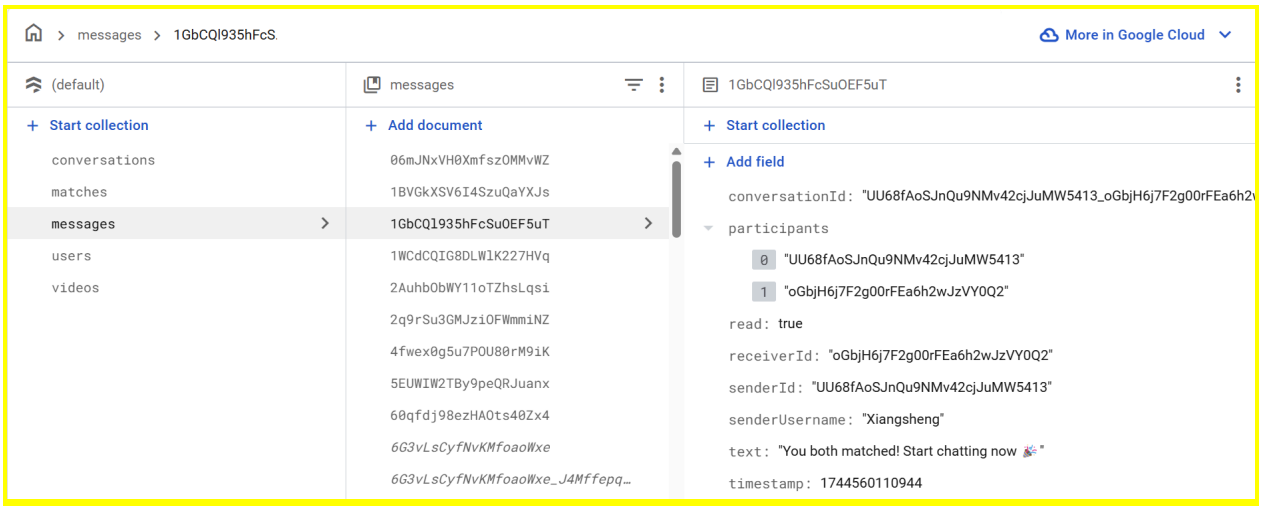
- Purpose: Fetch messages → Firestore Collection: messages → Operation: query + onSnapshot → Data Flow: On mount
- Purpose: Send message → Firestore Collection: messages → Operation: addDoc → Data Flow: On send

```
const sendMessage = async () => {
  if (!user) {
    Alert.alert("Login Required", "Please log in to send messages.");
    return;
  }

  if (newMessage.trim()) {
    const conversationId = [user.uid, senderId].sort().join("_");

    await addDoc(collection(FIRESTORE_DB, "messages"), {
      conversationId: conversationId,
      senderId: user.uid,
      senderUsername: user.displayName || "Unknown User",
      receiverId: receiverId,
      text: newMessage,
      timestamp: Date.now(),
      read: false,  // Mark as unread initially (for the recipient)
      participants: [user.uid, senderId],  // Store both user IDs for this conversation
    });

    setNewMessage("");
  }
};
```

- Purpose: Mark message read → Firestore Collection: messages/{id} → Operation: updateDoc → Data Flow: On receive

```
  // Iterate over messages and mark unread ones (sent by the other party) as read.
  newMessages.forEach((msg) => {
    if (msg.senderId !== user.uid && !msg.read) {
      // Update this message document to mark it as read.
      updateDoc(doc(FIRESTORE_DB, "messages", msg.id), { read: true });
    }
  });
});
```

- Data Storage: Firestore DB stores all user metadata and messages.



app/(tabs)/_layout.tsx

1. Component Interactions & Interface Logic:

- Tab Bar Routing: Uses expo-router's Tabs to handle screen-level routing. The "Post" tab is guarded — unauthenticated users are redirected to profile using router.replace.
- Interaction Highlight: This conditional interception of tabPress events is a clever use of Expo Router's listeners API to enforce auth-specific navigation policies.

```
<Tabs.Screen
  name="index"
  options={{
    tabBarLabel: "Home",
    tabBarIcon: () => (
      <Image source={require("../../assets/images/home.png")} style={{ width: 24, height: 24 }} />
    ),
  }}
/>
<Tabs.Screen
  name="post"
  listeners={{
    tabPress: (e) => {
      if (!user) {
        e.preventDefault();
        router.replace({
          pathname: "/profile",
          params: { fromPost: "true", timestamp: Date.now().toString() },
        });
      }
    },
  }}
  options={{
    tabBarLabel: "Post",
    tabBarIcon: () => (
      <Image source={require("../../assets/images/post.png")} style={{ width: 30, height: 30 }} />
    ),
  }}
/>
<Tabs.Screen
  name="inbox"
  options={{
    tabBarLabel: "Inbox",
    tabBarIcon: () => (
      <Image source={require("../../assets/images/inbox.png")} style={{ width: 24, height: 24 }} />
    ),
  }}
/>
```

listen: Auth status

app/(tabs)/index.tsx

1. Component Interactions & Interface Logic:

- Core Interaction: Tinder-style swipe gestures implemented with Reanimated + Gesture Handler control video reactions (like, dislike, save, view description).
- Notable Detail:
  - Swiping triggers Reanimated transitions and UI overlays (like green/red feedback).
  - A swipe up reveals the item description overlay.
  - Swiping right initiates a match, potentially triggering a chat and a push notification.

- Gesture Feedback: Visual indicators change based on swipe direction and intensity.

2. Design Choices:

- Swipe-to-interact: Reinvents listing UX by gamifying it (via video cards).

```
// Gesture Handler
const panGesture = Gesture.Pan()
  .onStart(() => {
    translateX.value = 0;
    translateY.value = 0;
    opacity.value = 1;
  })
  .onUpdate((e) => {
    translateX.value = e.translationX;
    translateY.value = e.translationY;
    const dragDistance = Math.sqrt(e.translationX ** 2 + e.translationY ** 2);
    opacity.value = Math.max(1 - dragDistance / 400, 0.5);
  })
  .onEnd((e) => {
    const isHorizontal = Math.abs(e.translationX) > Math.abs(e.translationY);
    const isSwipeUp = e.translationY < 0;

    if (isHorizontal || e.translationY > 0) {
      // Handle left/right/down swipes
      let targetX = 0;
      let targetY = 0;
      let direction: 'left' | 'right' | 'down' | null = null;

      if (isHorizontal) {
        if (Math.abs(e.translationX) > SWIPE_THRESHOLD || Math.abs(e.velocityX) > SWIPE_VELOCITY_THRESHOLD) {
          direction = e.translationX > 0 ? 'right' : 'left';
          targetX = e.translationX > 0 ? winWidth * 1.5 : -winWidth * 1.5;
        }
      } else {
        if (e.translationY > SWIPE_THRESHOLD || e.velocityY > SWIPE_VELOCITY_THRESHOLD) {
          direction = 'down';
          targetY = winHeight * 1.5;
        }
      }

      if (direction) {
        translateX.value = withTiming(targetX, { duration: 250 });
        translateY.value = withTiming(targetY, { duration: 250 });
        opacity.value = withTiming(0, { duration: 250 }, () => {
          runOnJS(handleSwipeComplete)(direction!);
        });
      } else {
        translateX.value = withTiming(0);
        translateY.value = withTiming(0);
        opacity.value = withTiming(1);
```

- Push Notifications: createChatBetweenUsers intelligently notifies both matched users. Choices

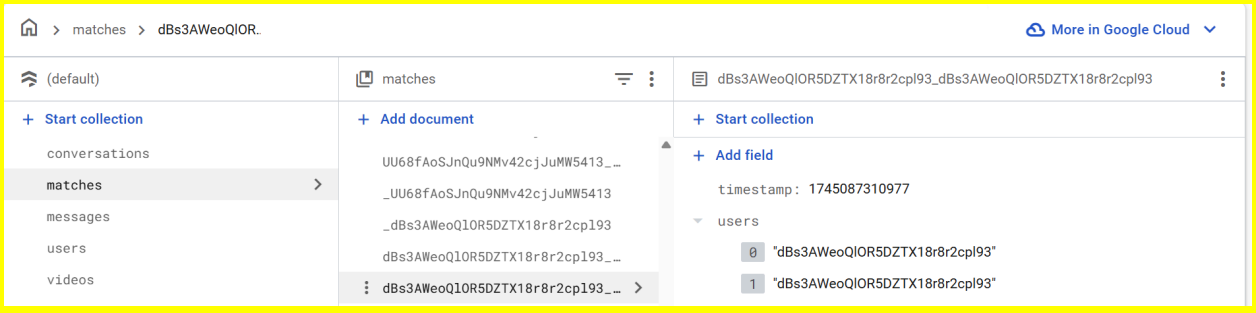
```
const createChatBetweenUsers = async (uid1: string, uid2: string) => {
  const chatId = [uid1, uid2].sort().join("_");

  // Create starter message
  await addDoc(collection(FIRESTORE_DB, "messages"), {
    conversationId: chatId,
    senderId: uid1,
    receiverId: uid2,
    senderUsername: user?.displayName || "New Match!",
    text: "You both matched! Start chatting now 🎉",
    timestamp: Date.now(),
    read: false,
    participants: [uid1, uid2],
  });

  // Fetch both push tokens
  const [snap1, snap2] = await Promise.all([
    getDoc(doc(FIRESTORE_DB, "users", uid1)),
    getDoc(doc(FIRESTORE_DB, "users", uid2))
  ]);

  const token1 = snap1.data()?.expoPushToken;
  const token2 = snap2.data()?.expoPushToken;

  if (token1) await sendPushNotification(token1, "You matched with someone!")
  if (token2) await sendPushNotification(token2, "You matched with someone!")
};
```

3. Challenges and Learnings:

- The dual-match logic in handleLike is sensitive to order (e.g., checking if reverse match exists first). Potential edge cases could be race-prone at scale.
- Integration of Reanimated gestures brings fluid motion to interfaces and can replace heavy Redux-based gesture state management.
- Firebase's arrayUnion is an excellent way to avoid duplicate saves across user fields like savedVideos.

4. REST API Endpoints, Data Flow, and Context:

- Purpose: Match handling → Firestore Collection: matches/{id} → Operation: setDoc, getDoc → Trigger: When liking (swipe right)



- Purpose: User saved/disliked → Firestore Collection: users/{uid} → Operation: updateDoc with arrayUnion → Trigger: When saving (swipe down) or disliking (swipe left)

```
const handleSave = async (videoId: string) => {
  if (!user) {
    Alert.alert("Login Required", "Please log in to save posts");
    return;
  }
  try {
    const userRef = doc(FIRESTORE_DB, "users", user.uid);
    await updateDoc(userRef, {
      savedVideos: arrayUnion(videoId)
    });
  } catch (error) {
    console.error("Error saving video:", error);
  }
  moveToNextVideo()
};
```

5. Device & SDK Contextual Data Used:

- Source: expo-notifications → Data Used: Push token registration & delivery

app/(tabs)/post.tsx

1. Component Interactions & Interface Logic:

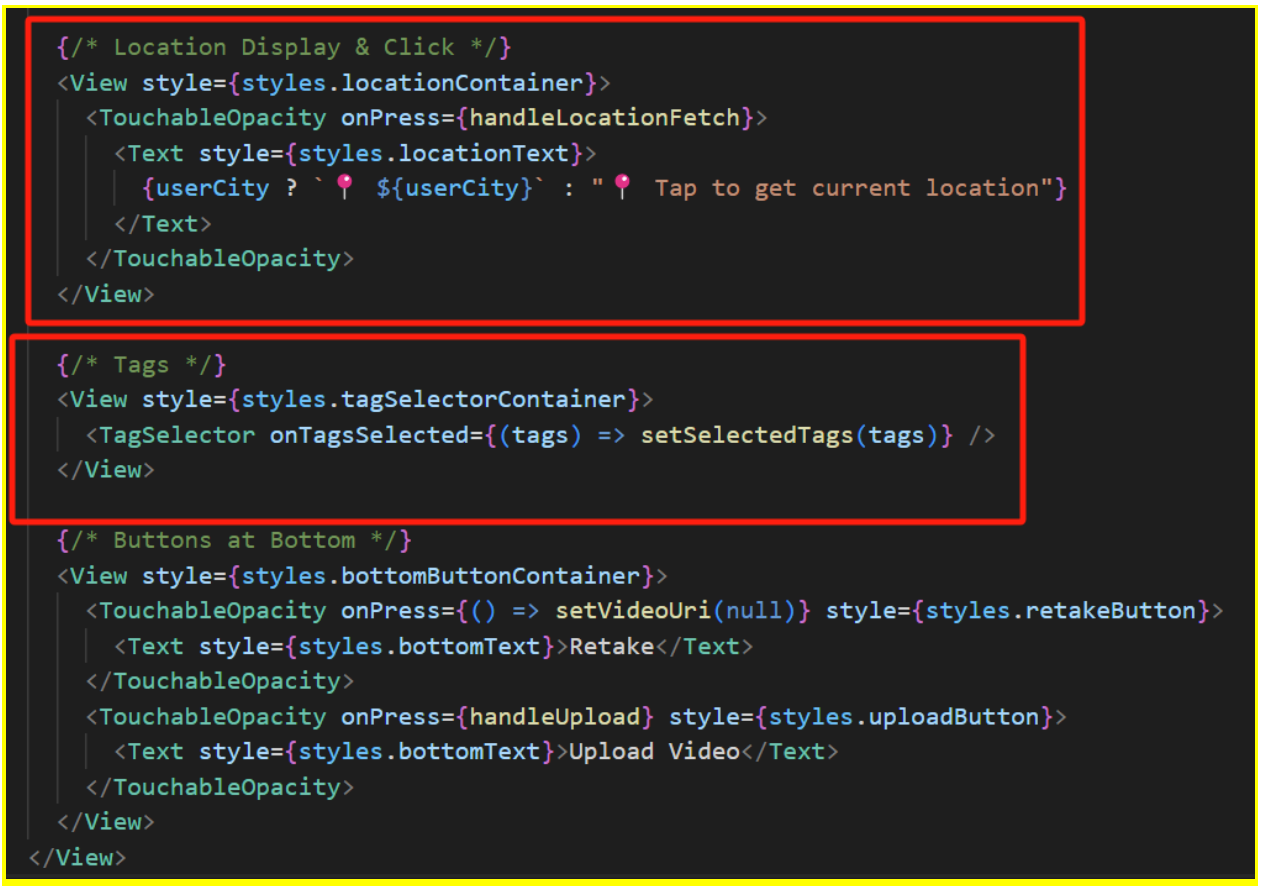
- Camera Interaction: Uses expo-camera + expo-av for recording videos and previewing them.

```
// apply for permissions directly
const isRequestingRef = useRef(false);
useEffect(() => {
  const requestPermissions = async () => {
    if (hasAutoRequestPermission || isRequestingRef.current) return;
    if (!permission || !microphonePermission) return;

    isRequestingRef.current = true;

    try {
      // For camera
      if (!permission.granted) {
        await requestCameraPermission();
      }
      // For microphone
      if (!microphonePermission.granted) {
        await requestMicrophonePermission();
      }
      // Location (only request permission here)
      const { status: locationStatus } = await Location.requestForegroundPermissionsAsync();
      if (locationStatus === "granted") {
        setLocationPermissionGranted(true);
      }
      // has applied one time
      setHasAutoRequestPermission(true);
    } catch (err) {
      console.error("Permission request failed:", err);
    } finally {
      isRequestingRef.current = false;
    }
  };

  requestPermissions();
}, [permission?.granted, microphonePermission?.granted]);
```

Permission: Video, micro and location

- Tagging System: Integrated item categorization via TagSelector component (custom).

```
import TagSelector from '../components/TagSelection';
```



```
{/* Location Display & Click */}
<View style={styles.locationContainer}>
  <TouchableOpacity onPress={handleLocationFetch}>
    <Text style={styles.locationText}>
      {userCity ? `📍 ${userCity}` : "📍 Tap to get current location"}
    </Text>
  </TouchableOpacity>
</View>

{/* Tags */}
<View style={styles.tagSelectorContainer}>
  <TagSelector onTagsSelected={(tags) => setSelectedTags(tags)} />
</View>

{/* Buttons at Bottom */}
<View style={styles.bottomButtonContainer}>
  <TouchableOpacity onPress={() => setVideoUri(null)} style={styles.retakeButton}>
    <Text style={styles.bottomText}>Retake</Text>
  </TouchableOpacity>
  <TouchableOpacity onPress={handleUpload} style={styles.uploadButton}>
    <Text style={styles.bottomText}>Upload Video</Text>
  </TouchableOpacity>
</View>
</View>
```

TagSelection and Location

- Location Fetch: Uses expo-location to capture user city and geolocation on upload.
- Video Metadata Handling: Users provide a title, description, and location for each recording before upload.

```
{videoUri ? (
  <ScrollView contentContainerStyle={styles.scrollViewContent}>
    <View style={styles.videoPreview}>
      <Video
        source={{ uri: videoUri }}
        style={styles.previewVideo}
        shouldPlay={false}
        resizeMode={ResizeMode.CONTAIN}
      />

      {/* Inputs */}
      <View style={styles.inputContainer}>
        <TextInput
          style={styles.commonInput}
          placeholder="Enter video title"
          placeholderTextColor="#bbb"
          numberOfLines={1}
          value={title}
          onChangeText={(text) => setTitle(text.slice(0, MAX_TITLE_LENGTH))}
        />
        <View style={styles.underline} >
          <Text style={styles.charCount}>
            {title.length}/{MAX_TITLE_LENGTH}
          </Text>
        </View>
      </View>
      <View style={[styles.inputContainer, styles.lastInputContainer]}>
        <TextInput
          style={[styles.commonInput, styles.descriptionInput]}
          placeholder="Enter video description"
          placeholderTextColor="#bbb"
          multiline
          numberOfLines={3}
          value={description}
          onChangeText={(text) => setDescription(text.slice(0, MAX_DESCRIPTION_LENGTH))}
        />
        <View style={styles.underline} >
          <Text style={styles.charCount}>
            {description.length}/{MAX_DESCRIPTION_LENGTH}
          </Text>
        </View>
      </View>
```

Inputs and thumbnails

2. Design Choices:

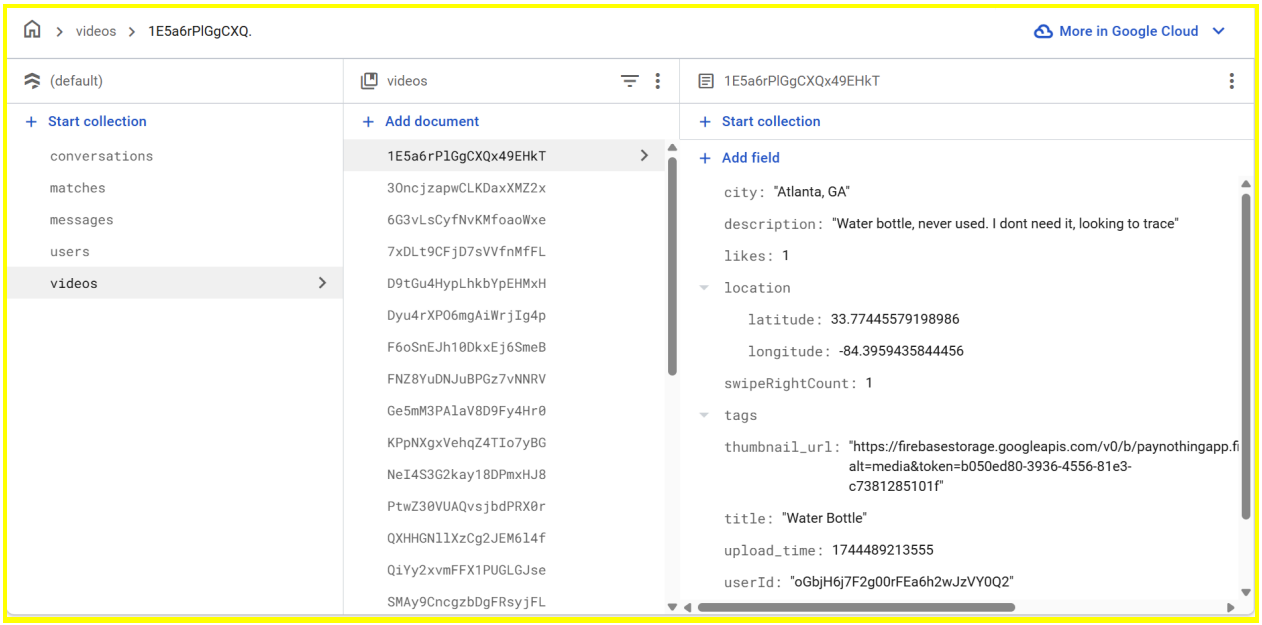
- Modular Upload Flow: Separate steps for permission, thumbnail creation, video upload, metadata saving.
- Tag Filtering: Enables item categorization and filtering, improving UX for search.

3. Challenges and Learnings:

- Effective use of Expo SDKs for device capabilities.
- On iOS, juggling camera, mic, and location permissions can lead to user friction. Pre-permission UIs help, but handling refusals gracefully is hard.

4. REST API Endpoints, Data Flow, and Context:

- Purpose: Upload video metadata → Firestore Collection: videos → Operation: addDoc → Trigger: After video is recorded & uploaded by users



Firebase-videos/{id}

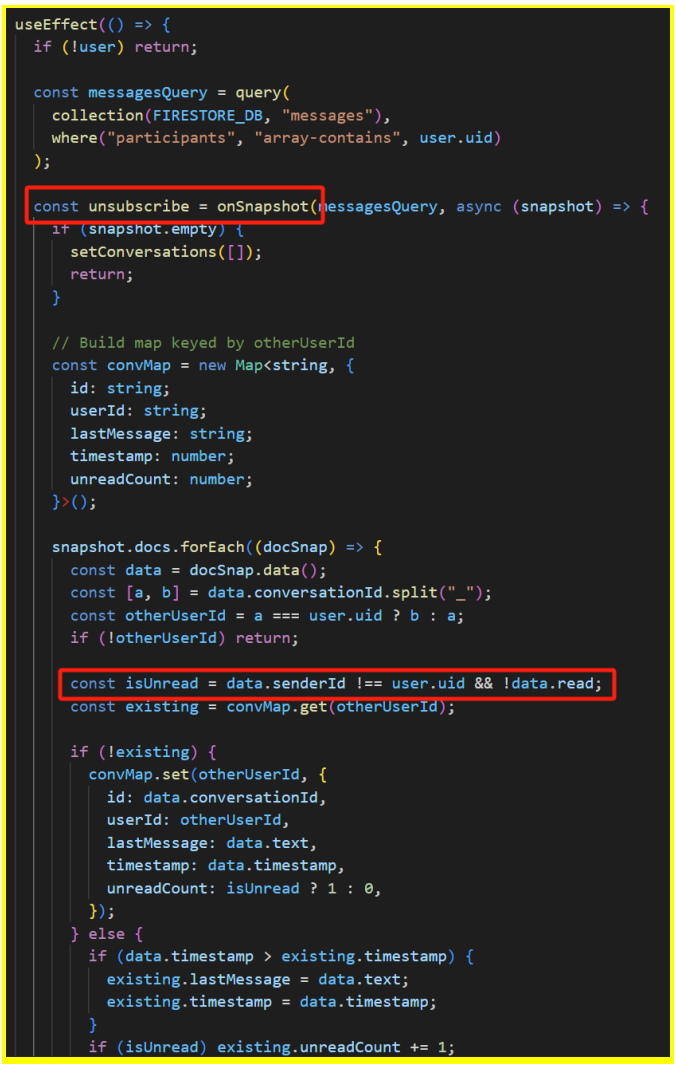
5. Device & SDK Contextual Data Used:

- Source: expo-camera → Data Used: Video recording and preview
- Source: expo-location → Data Used: City + coordinates (reverse geocoding)
- Source: expo-av → Data Used: Video playback
- Source: expo-video-thumbnails → Data Used: Thumbnail for post previews

app/(tabs)/inbox.tsx

1. Component Interactions & Interface Logic:

- UI Interaction: Displays recent conversations dynamically with unread indicators. A/B test logic chooses between banner or interstitial ads.
- Notable Design: Real-time updates via Firestore onSnapshot show how messages update live. Pushes clean summaries into FlatList rows.



```
useEffect(() => {
  if (!user) return;

  const messagesQuery = query(
    collection(FIRESTORE_DB, "messages"),
    where("participants", "array-contains", user.uid)
  );

  const unsubscribe = onSnapshot(messagesQuery, async (snapshot) => {
    if (snapshot.empty) {
      setConversations([]);
      return;
    }

    // Build map keyed by otherUserId
    const convMap = new Map<string, {
      id: string;
      userId: string;
      lastMessage: string;
      timestamp: number;
      unreadCount: number;
    }>();

    snapshot.docs.forEach((docSnap) => {
      const data = docSnap.data();
      const [a, b] = data.conversationId.split("_");
      const otherUserId = a === user.uid ? b : a;
      if (!otherUserId) return;

      const isUnread = data.senderId !== user.uid && !data.read;
      const existing = convMap.get(otherUserId);

      if (!existing) {
        convMap.set(otherUserId, {
          id: data.conversationId,
          userId: otherUserId,
          lastMessage: data.text,
          timestamp: data.timestamp,
          unreadCount: isUnread ? 1 : 0,
        });
      } else {
        if (data.timestamp > existing.timestamp) {
          existing.lastMessage = data.text;
          existing.timestamp = data.timestamp;
        }
        if (isUnread) existing.unreadCount += 1;
```

2. Challenges Encountered:

- Preventing redundant writes for “read” flags in chat.tsx without throttling may eventually require optimization.

```
return (
  <View style={styles.container}>
    <Text style={styles.conversationHeader}>Your Conversations</Text>
    {conversations.length === 0 ? (
      <Text style={styles.noConversations}>No past conversations found.</Text>
    ) : (
      <FlatList
        data={conversations}
        keyExtractor={(item) => item.id}
        renderItem={({ item }) => (
          <TouchableOpacity
            style={styles.conversationItem}
            onPress={() => openChat(item.userId, item.username)}
          >
            <Text style={styles.conversationText}>{item.username}</Text>
            <Text style={styles.conversationInfo}>
              {new Date(item.timestamp).toLocaleString()}
              {item.unreadCount > 0 ? ` · ${item.unreadCount} unread` : ""}
            </Text>
          </TouchableOpacity>
        )}
      />
    )}
    {/* banner */}
    {adType === "banner" && (
      <View style={{ alignItems: "center", marginTop: 10 }}>
        <AdsBanner />
      </View>
    )}
    {/* Interstitial */}
    <InterstitialAdOverlay visible={showInterstitial} onClose={() => setShowInterstitial(false)} />
  </View>
);
```

3. REST API Endpoints, Data Flow, and Context:

- Purpose: Messaging → Firestore Collection: messages → Operation: addDoc, onSnapshot, updateDoc → Trigger: Real-time chat

```
// Fetch each other-user's username
const convs = await Promise.all(
  Array.from(convMap.values()).map(async (conv) => {
    const userDoc = await getDoc(doc(FIRESTORE_DB, "users", conv.userId));
    return {
      ...conv,
      username: userDoc.exists()
        ? (userDoc.data()?.username as string) || "Undefined Username"
        : "Unknown Username",
    };
  })
);

// Sort and set
convs.sort((a, b) => b.timestamp - a.timestamp);
```

- Purpose: Matched Notification → Firestore Collection: users/{uid} → Operation: updateDoc → Trigger: After getting push token

app/(tabs)/profile.tsx

1. Component Interactions & Interface Logic:

- Dynamic Editing: Users can edit profile fields (age, gender, username) inline. Uses TextInput toggled with TouchableOpacity and Ionicons.

```
{/* Age Input */}
<View style={styles.infoRow}>
  <Text style={styles.infoLabel}>Age:</Text>
  {isEditingAge ? (
    <TextInput
      style={styles.editableInput}
      value={age}
      keyboardType="numeric"
      onChangeText={setAge}
      onBlur={() => {
        handleUpdateUserData("age", parseInt(age));
        setIsEditingAge(false);
      }}
    />
  ) : (
    <TouchableOpacity
      style={styles.infoValueContainer}
      onPress={handleAgeClick}
    >
      <Text style={styles.infoValue}>{age || "Add age"}</Text>
      <Ionicons name="pencil" size={16} color="#666" />
    </TouchableOpacity>
```

- Image Interaction: Users can update their profile image through the media library using expo-image-picker and upload it to Firebase Storage.

```
{/* Profile Header with Side-by-Side Layout */}
<View style={styles.profileHeader}>
  <TouchableOpacity
    style={styles.profileImageContainer}
    onPress={handleProfilePicturePress}
  >
    <Image
      source={profileImage ? { uri: profileImage } : require("../../assets/images/default-profile.png")}
      style={styles.profileImage}
    />
    <View style={styles.editPhotoBadge}>
      <Ionicons name="camera" size={18} color="white" />
    </View>
  </TouchableOpacity>
```

- Push Token Registration: Device token is stored using expo-notifications, enabling direct push notifications later.

```
async function registerForPushNotificationsAsync(uid: string) {
  if (!Device.isDevice) return;

  const { status: existingStatus } = await Notifications.getPermissionsAsync();
  let finalStatus = existingStatus;

  if (existingStatus !== 'granted') {
    const { status } = await Notifications.requestPermissionsAsync();
    finalStatus = status;
  }

  if (finalStatus !== 'granted') return;

  const tokenData = await Notifications.getExpoPushTokenAsync();
  const expoPushToken = tokenData.data;

  await updateDoc(doc(FIRESTORE_DB, "users", uid), {
    expoPushToken: expoPushToken,
  });
}
```

2. Device & SDK Contextual Data Used:

- Source: expo-image-picker → Data Used: Profile photo selection

```
const handleProfilePicturePress = async () => {
  if (!user) {
    Alert.alert("Error", "User not authenticated");
    return;
  }

  const permissionResult = await ImagePicker.requestMediaLibraryPermissionsAsync();
  if (!permissionResult.granted) {
    Alert.alert("Permission required", "We need access to your photos to update your profile picture.");
    return;
  }

  const pickerResult = await ImagePicker.launchImageLibraryAsync({
    mediaTypes: ImagePicker.MediaTypeOptions.Images,
    allowsEditing: true,
    aspect: [1, 1],
    quality: 0.8,
  });

  if (!pickerResult.canceled) {
    const uri = pickerResult.assets[0].uri;
    const response = await fetch(uri);
    const blob = await response.blob();

    const storage = getStorage();
    const storageRef = ref(storage, `profile-pictures/${user.uid}`);

    await uploadBytes(storageRef, blob);
    const downloadURL = await getDownloadURL(storageRef);

    await updateProfile(user as User, { photoURL: downloadURL });
    setProfileImage(downloadURL);
  }
};
```

Image picker

app/components/Ads.tsx

1. Component Interactions & Interface Logic:

- Banner Ad: Passively displayed ad box with a promotional message for upselling premium listings.

```jsx
// Banner
export const AdsBanner = () => (
  <View style={styles.bannerContainer}>
    <View style={styles.adLabel}>
      <Text style={styles.adLabelText}>Ad</Text>
    </View>
    <View style={styles.bannerContent}>
      <Text style={styles.bannerTitle}>💡 Boost Your Visibility!</Text>
      <Text style={styles.bannerDesc}>Promote your listing to reach more users today.</Text>
    </View>
  </View>
);
```

- Interstitial Ad: Full-screen overlay using Modal + Animated opacity transitions.
    - Automatically fades in/out based on visible state.
    - Has a dismiss button and a dummy CTA that can later be linked to a web view or purchase screen.
- Interaction Highlight: Animated.View combined with Modal enables clean, smooth transitions between ad states.

```tsx
// Interstitial
export const InterstitialAdOverlay = ({
  visible,
  onClose,
}: {
  visible: boolean;
  onClose: () => void;
}) => {
  const fadeAnim = new Animated.Value(0);

  useEffect(() => {
    if (visible) {
      Animated.timing(fadeAnim, {
        toValue: 1,
        duration: 300,
        useNativeDriver: true,
      }).start();
    } else {
      Animated.timing(fadeAnim, {
        toValue: 0,
        duration: 200,
        useNativeDriver: true,
      }).start();
    }
  }, [visible]);

  return (
    <Modal visible={visible} transparent animationType="fade" onRequestClose={onClose}>
      <Animated.View style={[styles.modalContainer, { opacity: fadeAnim }]}>
        <View style={styles.interstitialBox}>
          <Text style={styles.interstitialTitle}>🔥 Limited Time Offer!</Text>
          <Text style={styles.interstitialDesc}>
            Upgrade to premium for 50% off. Click below to learn more.
          </Text>
          <TouchableOpacity style={styles.ctaButton} onPress={() => alert("Ad clicked!")}>
            <Text style={styles.ctaText}>Learn More</Text>
          </TouchableOpacity>
          <TouchableOpacity style={styles.closeButton} onPress={onClose}>
            <Text style={styles.closeText}>✕</Text>
          </TouchableOpacity>
        </View>
      </Animated.View>
    </Modal>
  );
};
```

2. Design Choices:

- Smooth entrance/exit animations via Animated and Animated.timing enhance polish (especially in interstitial ads).

```
useEffect(() => {
  if (visible) {
    Animated.timing(fadeAnim, {
      toValue: 1,
      duration: 300,
      useNativeDriver: true,
    }).start();
  } else {
    Animated.timing(fadeAnim, {
      toValue: 0,
      duration: 200,
      useNativeDriver: true,
    }).start();
  }
}, [visible]);
```

3. Data Flow and Context Usage:

- Component: AdsBanner → Receives Props? No → Sends Data? No → Where Data Goes? Visual only
- Component: InterstitialAdOverlay → Receives Props? Yes (visible, onClose) → Sends Data? No → Where Data Goes? Controls itself & notifies parent

app /components/TagSelection.tsx

1. Component Interactions & Interface Logic:

- Interaction Design:
  - Modal-style picker for tag selection with a visible tag summary bar outside the modal.

```
<View style={styles.container}>
  {/* outside of dialog: show selected tags */}
  <TouchableOpacity
    style={styles.tagRow}
    onPress={() => setModalVisible(true)}
  >
    <View style={styles.tagRowContent}>···
    </View>
  </TouchableOpacity>

  {/* Dialog */}
  <Modal···
  </Modal>
</View>
```

  - Inside the modal: toggles tag selection with limit enforcement (max 3).

- Tag selection is communicated back to the parent via onTagsSelected.

```
const TAGS = Object.values(ItemTag);
interface TagSelectorProps {
    onTagsSelected: (tags: string[]) => void;
}

export default function TagSelector({ onTagsSelected }: TagSelectorProps) {
  const [selectedTags, setSelectedTags] = useState<string[]>([]);
  const [preSelected, setPreSelected] = useState<string[]>([]);
  const [modalVisible, setModalVisible] = useState(false);
  const numsTagLimit = 3;
```

- UX Detail:
  - The "×" button next to selected tags offers intuitive de-selection.
  - Dynamic messaging (e.g., "Up to 2 more tags…") enhances clarity.

```
const toggleTag = (tag: string) => {
  setSelectedTags(prev => {
      if (prev.includes(tag)) {
          return prev.filter(t => t !== tag);
      }
      else if (prev.length < numsTagLimit) {
          return [...prev, tag];
      }
      return prev;
  });
};
```

Limits of tags

2. Design Choices:

- All components are self-contained, accept props (like visible, onClose, onTagsSelected) and remain UI-agnostic.

3. Learnings:

- TagSelector uses a preSelected state to restore previous tag selections if the user cancels mid-selection — avoids overwriting parent state unintentionally.

4. Data Flow and Context Usage:

- Component: TagSelector → Receives Props? Yes (onTagsSelected) → Sends Data? Yes → Where Data Goes? Notifies post.tsx or filter UI

app/components/VideoViewModal.tsx

1. Component Interactions & Interface Logic:

- Video Preview: window-screen modal to preview video posts from profile page (your posts / saved posts).
- Touch Interaction:
    - Tap to play/pause video.
    - Tap outside modal to close.

```
// auto-playing
useEffect(() => {
  if (visible && videoToShow && videoRef.current) {
    const playTimeout = setTimeout(() => {
      videoRef.current?.playAsync().then(() => setIsPlaying(true)).catch((e) => {
        console.warn("Playback error:", e);
        setIsPlaying(false);
      });
    }, 500);
    return () => clearTimeout(playTimeout);
  }
}, [visible, videoToShow]);

// pause when Modal close
useEffect(() => {
  if (!visible && videoRef.current) {
    videoRef.current.stopAsync().catch(() => {});
    setIsPlaying(false);
  }
}, [visible]);

// pause by user
const togglePlayback = async () => {
  if (videoRef.current) {
    const status = await videoRef.current.getStatusAsync();
    if (status.isPlaying) {
      await videoRef.current.pauseAsync();
      setIsPlaying(false);
    } else {
      await videoRef.current.playAsync();
      setIsPlaying(true);
    }
  }
};
```

Pause, Autoplay

- UX Enhancer:

    - Overlay text for video title and description.
    - Play icon overlay for paused state provides strong visual feedback.

2. Design Choices:

- All modal-based interactions (ads, tags, video view) are built using Modal, ensuring visual and behavioral consistency.

3. Learnings:

- Auto-play is delayed with a timeout (500ms) to give the modal a chance to render before attempting playback, which prevents premature errors. Fallbacks and warnings (catch, console errors, conditionally rendering null) are embedded defensively.

4. Data Flow and Context Usage:

- Component: VideoViewModal → Receives Props? Yes (video metadata) → Sends Data? No → Where Data Goes? Renders video from Firestore URL

```tsx
interface VideoModalProps {
  visible: boolean;
  videoToShow: VideoItem | null;
  onClose: () => void;
}

const VideoViewModal: React.FC<VideoModalProps> = ({ visible, videoToShow, onClose }) => {
  const videoRef = useRef<Video>(null);
  const [isLoaded, setIsLoaded] = useState(false);
  const [isPlaying, setIsPlaying] = useState(false);
```

Second DashBoard Code Review

_layout.tsx (Root and Tab Navigation)

For the navigation setup, we organized our project by creating a root layout and a tab layout inside _layout.tsx. In the root layout, we set up a Stack navigator and managed the splash screen manually using SplashScreen.preventAutoHideAsync() so that the app only displayed after essential assets like fonts had loaded. We made this **design choice** because we wanted the app to feel polished and professional from the moment it opened. Applying a ThemeProvider connected to the useColorScheme() hook allowed the app to automatically support dark and light modes without duplicating styling logic. A notable **challenge** was ensuring fonts were loaded properly before rendering — otherwise, users would briefly see unstyled text. Solving this **taught us** the importance of tight asset management during startup. Our navigation structure **was a success** because it created a smooth transition experience between screens and gave the app a cohesive, finished feel across different device types.

```tsx
// Prevent the splash screen from auto-hiding before asset loading is complete.
SplashScreen.preventAutoHideAsync();

export default function RootLayout() {
  const colorScheme = useColorScheme();
  const [loaded] = useFonts({
    SpaceMono: require('../assets/fonts/SpaceMono-Regular.ttf'),
  });

  useEffect(() => {
    if (loaded) {
      SplashScreen.hideAsync();
    }
  }, [loaded]);

  if (!loaded) {
    return null;
  }

  return (
    <ThemeProvider value={colorScheme === 'dark' ? DarkTheme : DefaultTheme}>
      <Stack>
        <Stack.Screen name="(tabs)" options={{ headerShown: false }} />
        <Stack.Screen name="+not-found" />
      </Stack>
      <StatusBar style="auto" />
    </ThemeProvider>
  );
}
```

index.tsx (Dashboard Screen)

In index.tsx, we built the main backend dashboard screen where users can view real-time metrics like total users, videos uploaded, messages sent, and matches made. We decided to centralize all stats onto one scrollable dashboard so users could quickly monitor key app activity at a glance. One major **challenge** was how to fetch all this backend data efficiently without slowing down the user interface. We overcame this by batching API calls together with Promise.all(), learning how critical it is to optimize network performance in mobile apps. We also added pull-to-refresh using RefreshControl to allow users to manually reload stats without closing the app. This **design choice** made the dashboard feel dynamic and responsive. The **success** of this approach was evident — users could see live stats, swipe to refresh, and experience minimal lag, resulting in an intuitive and snappy user experience.

```tsx
const loadData = async () => {
  try {
    // Totals + series
    const [users, videos, messages, matches, signupSeries, vStats, mStats, mtStats, dau] = await Promise.all([
      getTotalUsers(),
      getTotalVideos(),
      getTotalMessages(),
      getTotalMatches(),
      getUserSignupStats(),
      getVideoUploadStats(),
      getMessageVolumeStats(),
      getMatchVolumeStats(),
      getDailyActiveUsersStats(),
    ]);
```

ActivityList.tsx (Activity Feed)

Inside ActivityList.tsx, we created a clean, efficient way to display a feed of recent activities like new users, videos, and matches. We used a FlatList because it efficiently renders large data sets while supporting scrolling performance. Our **design choice** here was to keep each activity item minimal — only the title, description, and timestamp — to prioritize clarity. A **challenge** we encountered was ensuring that even with no activities, the app handled it gracefully by displaying a friendly empty state message. Through this, **we learned** the importance of planning for empty data scenarios early in development. Our lightweight, modular approach was **successful** because it kept the Activity Feed smooth and prevented any jarring UI errors, even if backend data was temporarily missing.

```tsx
import { StyleSheet, FlatList, View } from 'react-native';
import { ThemedText } from './ThemedText';
import { ThemedView } from './ThemedView';

interface ActivityItem {
  id: string;
  type: 'user' | 'video' | 'message' | 'match';
  title: string;
  description: string;
  timestamp: Date;
}

interface ActivityListProps {
  activities: ActivityItem[];
}

export function ActivityList({ activities }: ActivityListProps) {
  const renderItem = ({ item }: { item: ActivityItem }) => (
    <ThemedView style={styles.item}>
      <View style={styles.contentContainer}>
        <ThemedText type="defaultSemiBold" style={styles.title}>{item.title}</ThemedText>
        <ThemedText style={styles.description}>{item.description}</ThemedText>
      </View>
      <ThemedText style={styles.timestamp}>
        {new Date(item.timestamp).toLocaleDateString()}
      </ThemedText>
    </ThemedView>
  );
```

DAUChart.tsx (Daily Active Users Chart)

In DAUChart.tsx, we created a straightforward way to display daily active users using a simple list format, pairing each date with its corresponding user count. We **intentionally** kept the chart simple and text-based because we prioritized clear readability over flashy graphs at this stage. One **challenge** was formatting the dates nicely without cluttering the screen, especially given the smaller real estate on mobile devices. This component taught us that sometimes minimalism

is the better design choice when visualizing backend data. Our **success** came from how easy it was for users to quickly scan daily trends without needing to interpret complicated charts or graphs.

```tsx
interface DAUStats {
  date: string;
  count: number;
}

interface DAUChartProps {
  data: DAUStats[];
}

export const DAUChart: React.FC<DAUChartProps> = ({ data }) => {
  const formatDate = (dateStr: string) => {
    const date = new Date(dateStr);
    return `${date.getMonth() + 1}/${date.getDate()}`;
  };

  return (
    <ThemedView style={styles.container}>
      <ThemedText style={styles.title}>Daily Active Users</ThemedText>
      <View style={styles.content}>
        {data.map((item, idx) => (
          <View key={idx} style={styles.row}>
            <ThemedText>{formatDate(item.date)}: </ThemedText>
            <ThemedText>{item.count} active users</ThemedText>
          </View>
```

Concerns

Sprint 5:

Concern 1: App Diffusion

- **Description:** One concern we had moving forward is getting out app out there to the public. How could we spread awareness and spread the overall word of the deployment of our app? Especially with other big platforms such as Ebay and Facebook Marketplace out there?
- **Solution:** Similar to other marketing strategies such as Fetti (ride share app), starting off marketing in college campuses is a smart strategy to spread the word. As college kids tend to be cost conscious, a bartering app could be useful for college students and all students will be in close proximity. By promoting our app to students around campus, this would be a great kickstart to our app audience.

Concern 2: Spammers

- **Description:** Another concern that was brought up was potential users swiping likes on every post, with no limit. This could cause issues with overuse of the app, trying to get every item that is posted on the app.
- **Solution:** Though our matching algorithm limits this potential issue, we plan to add another countermeasure, the number of likes that can be given per day (similar to Hinge daily limit). This way, users are more specific in what products they want to like.

Sprint 4:

Concern 1：**Item Quality and Accuracy**

- **Description**: Since our platform relies on user-recorded videos for item listings, there is a risk that users may misrepresent the quality or condition of the items, leading to dissatisfaction and disputes.
- **Solution**: Provide guidelines and tips for creating high-quality, honest videos that accurately showcase the condition of the items. Introduce a "report item" feature, allowing users to flag misleading listings, with penalties for repeated offenders. Offering an optional verification service for sellers could help assure buyers of item quality.

Concern 2: Trust and Safety

- **Description**: A key concern for our platform is ensuring that users feel safe when trading with strangers. Without sufficient trust, users may hesitate to participate in exchanges or transactions, especially when bartering items of value.
- **Solution**: Implement a robust user verification system, including profile pictures. Introduce a reputation system where users can rate each other after each transaction, providing feedback on trustworthiness. Additionally, integrate in-app messaging for secure communication and encourage meetups in public locations.

Sprint 3:

Challenge 1:

- **Description:** Facing our community barter platform is to ensure that there is a large enough user base. Because the premise of bartering is to have a large enough number of users participating, so as to provide more items to facilitate transactions.
- **Solution:** Conduct some surveys or consultations to try to find out where the current potential users get the information about bartering? For example: integration with existing online communities: such as Facebook groups, Reddit forums, WeChat group chats, Telegram; cooperation with some offline flea markets?

Challenge 2:

- **Description:** Ensuring the safety of our customers when engaging in transactions and bartering actions. There may be illegitimate claims and our app must ensure as much as possible to eliminate these threats in order to ensure safety of our customers and validity of these transactions.
- **Solution:** We must implement countermeasures in order to prevent these actions, simple solutions include meeting in public areas to ensure a safer transaction environment, in app verification of people ID's through multi-factor authentication, safety deposit on both sides to prevent scamming, and a rating system for sellers to allow future customers to see their past customers opinions.

Appendix

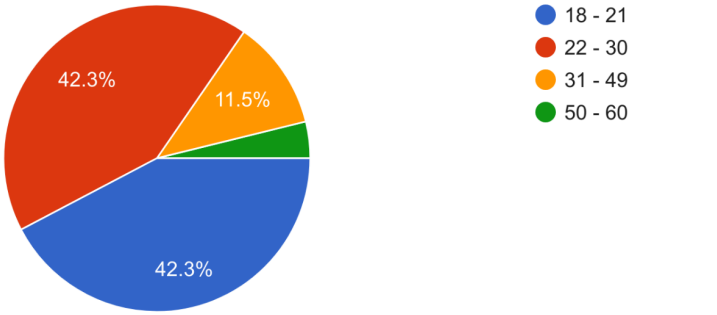
Sprint 4:

A/B Testing from Survey Responses on the Second Learning Prototype

Please access this survey link to view all questions related to prototype comparison: https://forms.gle/xFSY93ygxope6yeB8

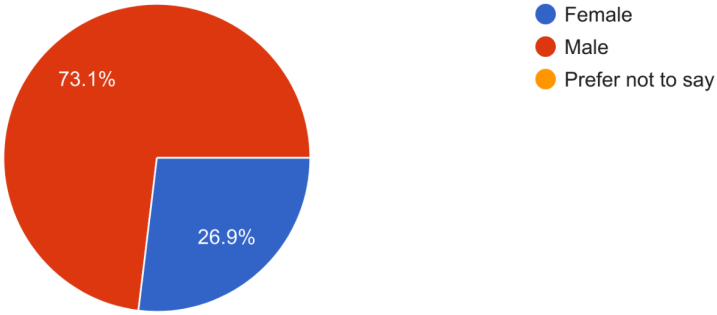
Select your age:

26 responses



What is your gender?

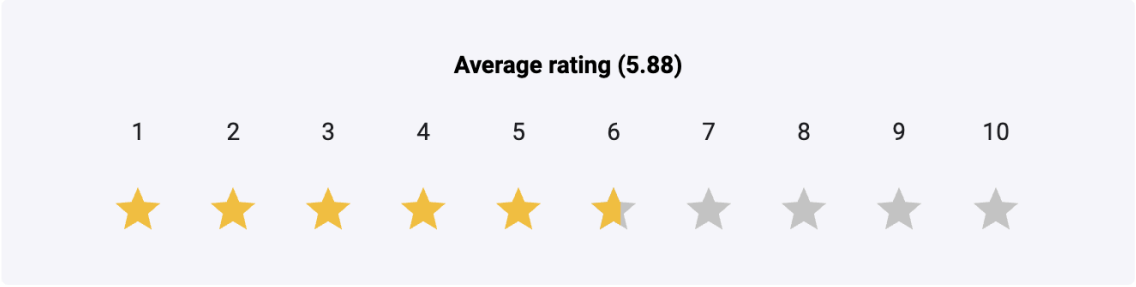
26 responses

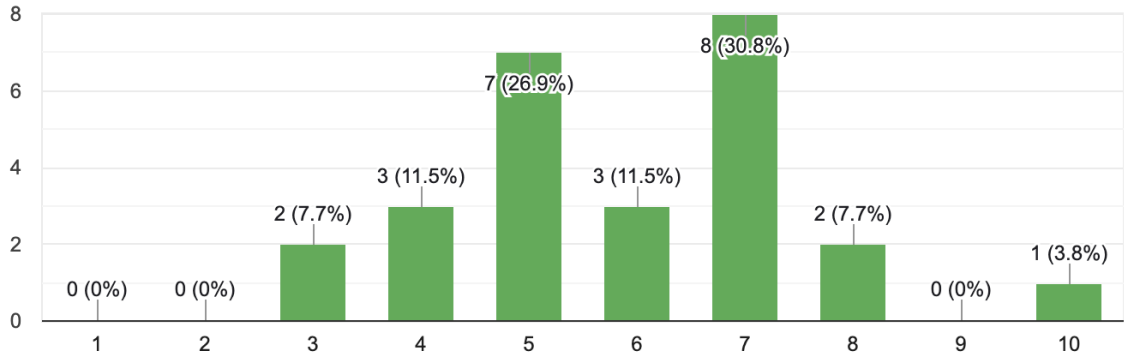


Rate our **previous** prototype on the **Authentication feature:**
(user **can** browse video posts but **cannot** post, like, or message items until they navigate to Profile and sign-in/sign-up)

Copy chart

26 responses

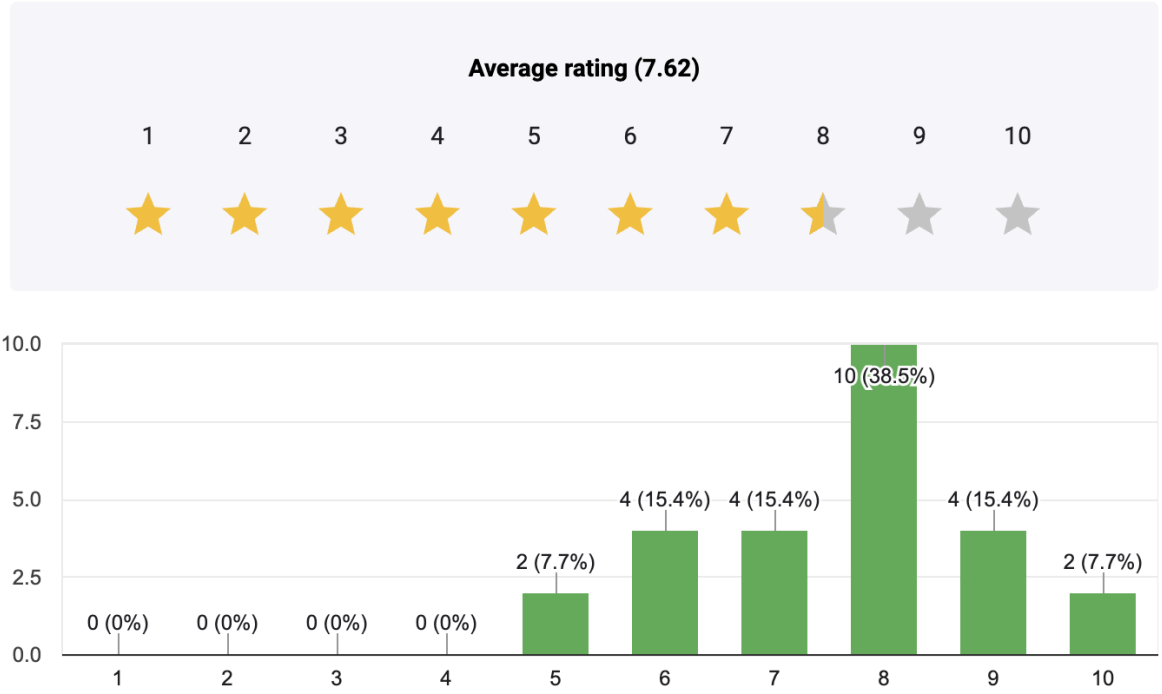




Rate our **current** prototype on the **Authentication feature:**
(user have to sign-in/sign-up **first**, and then they can interact with **all features** of the app. The Profile page will automatically load user info.)

Copy chart

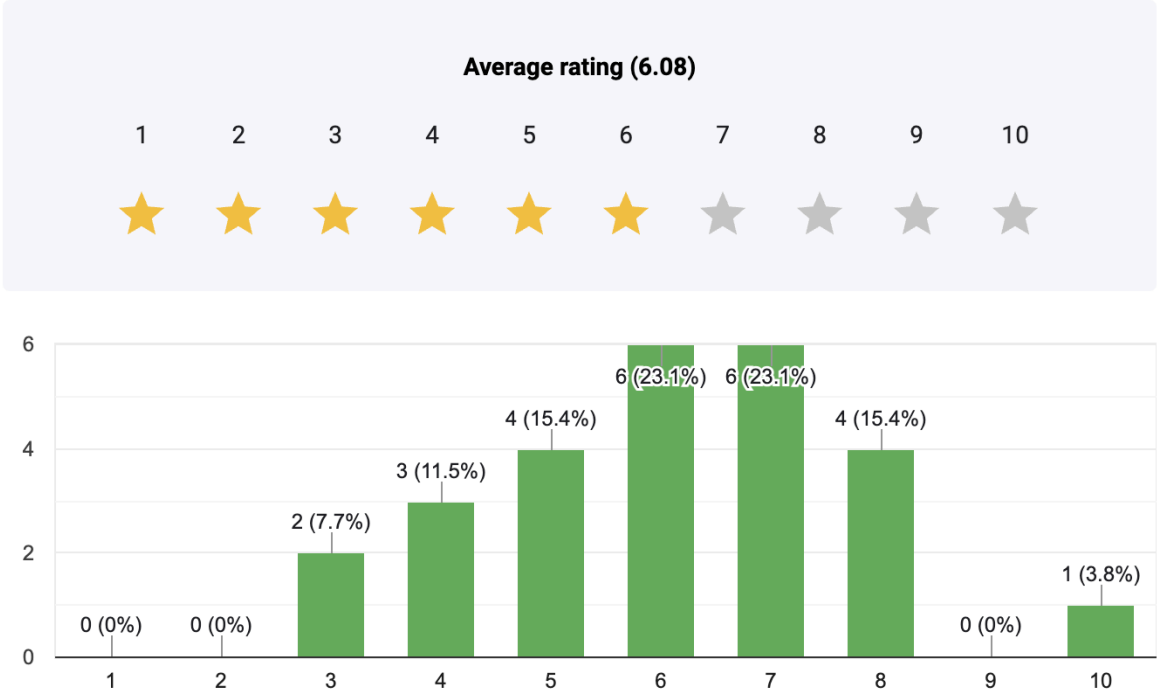
26 responses



Rate our **previous** prototype on the **Post items feature:**
(user can **record a short video** of the item they want to trade or give away. User can **talk** to introduce the item, or they can **add title and description** as details):

Copy chart

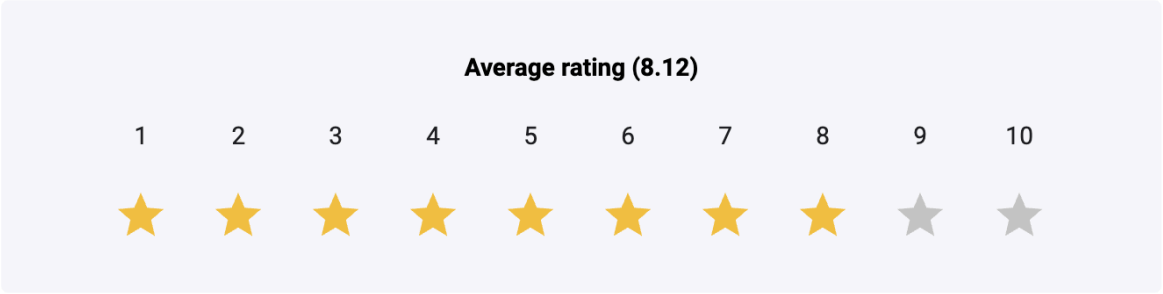
26 responses

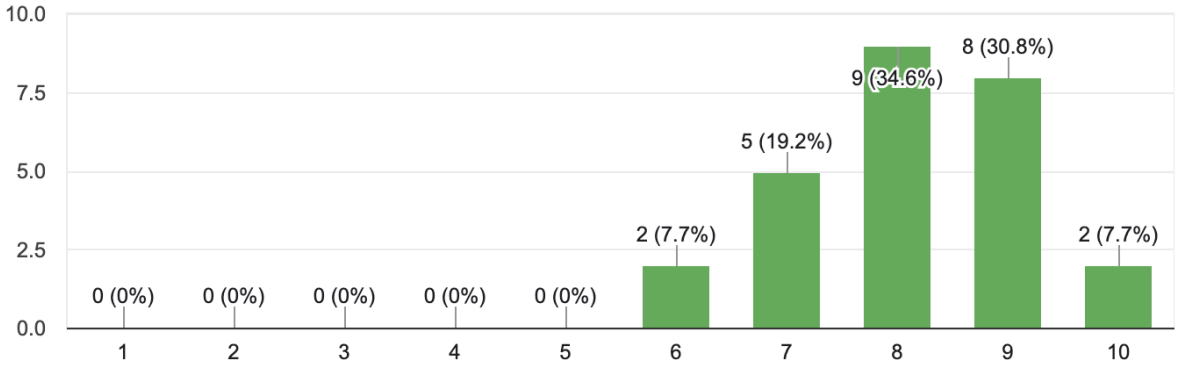


Rate our **current** prototype on the **Post items feature:**
(Changed to **white & yellow** background. User now can add **tags** to **categorize** their item so it is more organized for other users):

Copy chart

26 responses

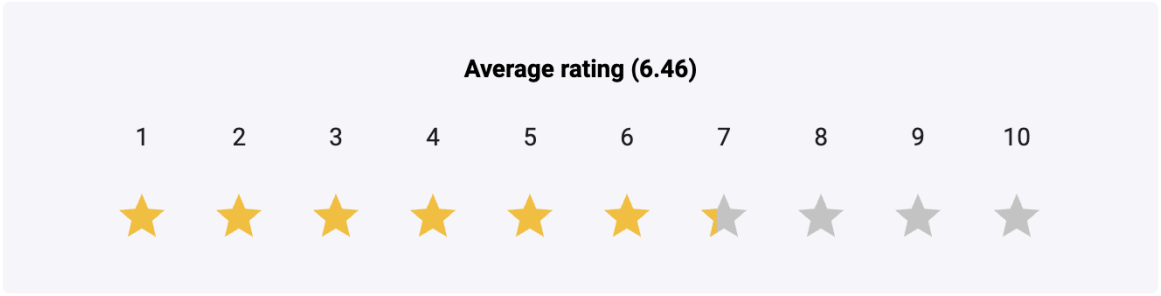


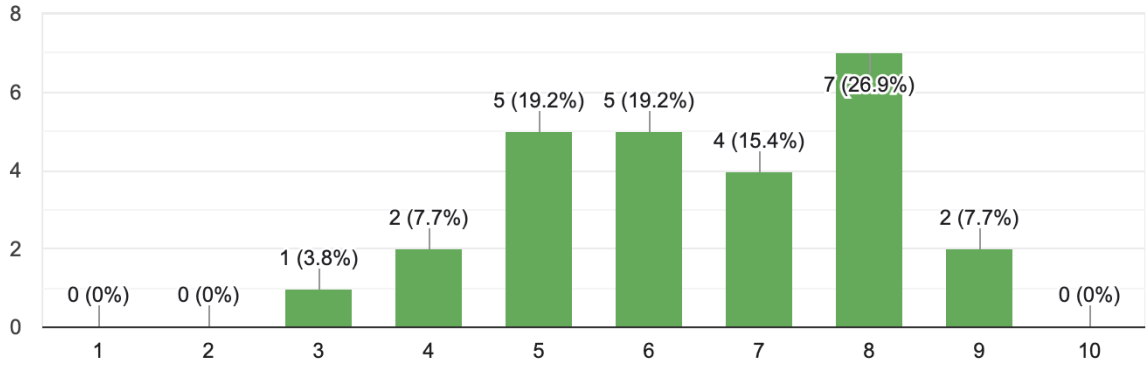


Rate our **previous** prototype on **interacting with video posts:** (user **scroll** through videos to view posted items; **tap** the 'heart' to like a post; **tap** the 'message' to chat; item **title** and **description** are **displayed** at the **bottom** of each video)

Copy chart

26 responses

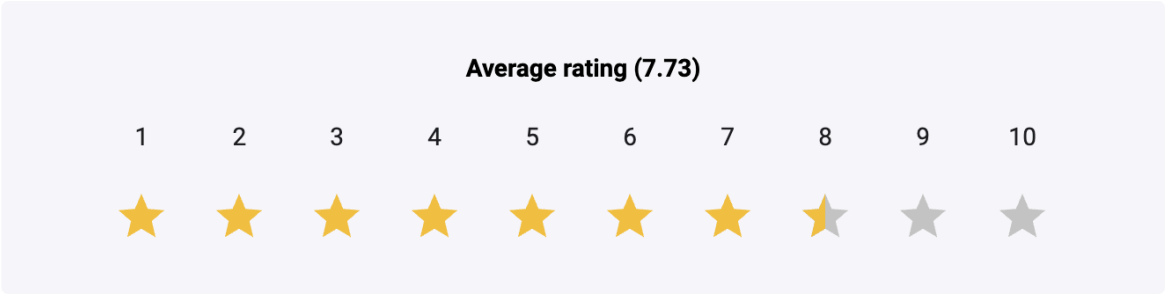


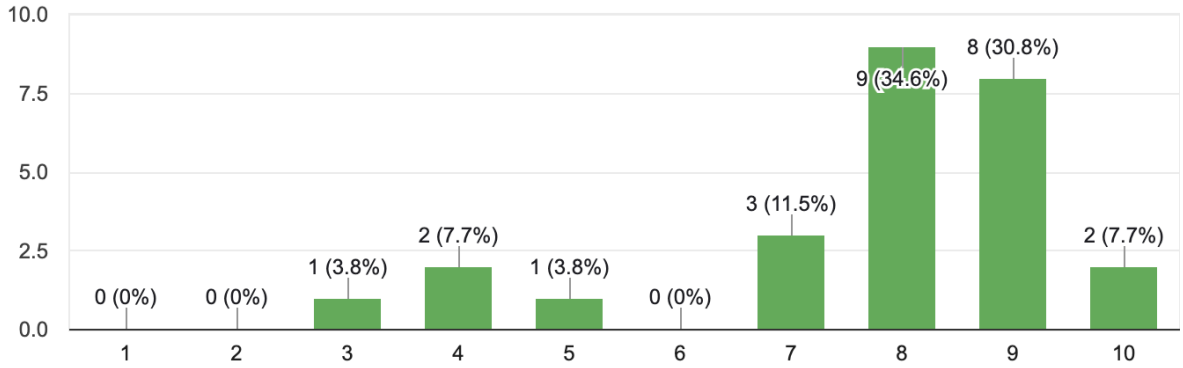


Rate our **current** prototype on i**nteracting with video posts:**
(User can **filter** items based on **categories**. User interact with **one** video post at a time: **swipe left** to '**dislike**' the item, **swipe right** to '**like**' the item, **swipe up** to view the item **description**, and **swipe down** to **save** the video for later. **Instant match** happens when two posted items 'like' each other.)

Copy chart

26 responses





Thank you for your time! It is optional but please briefly **share your thoughts/suggestions/concerns**:

2 responses

Swipe 4 different directions for different actions is confusing

What was the purpose of gathering the age and gender information from survey participants? I'm not criticizing; I just wanted to understand what you were hoping to achieve.

Sprint 3:

Interviews on the Selected Learning Prototype

Participants acknowledged popular e-commerce platforms that allow local bartering such as Facebook Marketplace and OfferUp, and they knew the `Problem Overview'. Participants used the PayNothing app as “Sign-up/sign-in -> Posting items -> Browsing videos -> Messaging others”. Participants are asked questions related to “Posting Item and Video Feeds” — our Selected Learning Prototype.

Questions are as such:

- Interviewee age, gender, social role (optional), and other demographic information (optional).
- How convenient is our app to use on a scale of 1-10 (Focus on primary use case)? May compare with popular e-commerce platforms that you previously used.
- How easy is it to browse videos on the home page, did you find any issues or some other features you wish it can have?
- How was your experience of posting videos? Were there any issues or something you wish could be different?
- How was your experience of messaging someone? Were there any issues or something you wish could be different?
- If you like an item you saw, would you message them? Did you app make it easy for you to contact them?
- For the overall functionality of the app, were you able to effectively browse video feeds, post videos, and message users? If not, why not?

Here are some of the collected responses with key feedback:

Interviewee 1: 24 years old, male, a current MSCS student at Tufts University

- I would consider this app an 8 on a scale of 1-10 overall.
- UI is okay but there is an alignment issue, so this feature is like a 7 on a scale of 1-10.
- I wish to have a “comments” area for each post to discuss the item, and also an item category to classify different items, which you can refer to the “Xian Yu” app.
- For better user experience, item description on the video can auto fold if it is too long.
- Why can I not simply upload videos from my phone?
- If this constraint has a reason then please introduce the reason in the guide when users use it for the first time.
- Messaging is necessary as I need to contact the user regarding the item I like, but currently the “inbox” is not displaying my previous chats, and I want the chat to show a timestamp as well.
- I'm able to effectively post and message users, but the keyboard blocks my view when messaging users and the keyboard can not automatically hide once finishing editing stuff.

- Also, I want my personal information not always displayed on the profile screen but they can stay in a setting page.

Interviewee 2: 25 years old, male, a current MSCS student at Northeast University

- I gave Facebook Marketplace (8/10) and OfferUp (7/10), so I would consider this app a 5 on a scale of 1-10.
- Items need to be categorized somehow I think. I would add a location display to only show items in that area. I would like to change the view of video posts in grid of 4.
- The search function can only search the exact keywords in title, smart search mechanisms and search history are needed I think.
- After signing-in via phone number or email address, I tapped 'Post' but cannot view the page and it keeps asking me to login to view.
- I wish I could tag the video in terms of item category so each posted item can be sorted in different categories.
- Can a post contain multiple videos? What if I want to post the item in different angles but one hand is holding the phone to record and another hand cannot move the item conveniently?
- I would want to know the time of each message, whether I actually sent the message, the user actually received my message, and read my message, so updating the status of each message would be helpful.
- I think the overall functionality satisfies my use for posting used items and messaging other users, but I would also like it to have a function somewhere to change the city/distance of item posts.

Interviewee 3: 50 years old, male, a business owner

- I would consider this app a 5 on a scale of 1-10 overall.
- How can this app be profitable besides Ads and subscriptions?
- Can you give some kind of rewards for active users who made 100 item transactions on the app?
- Will there be a special area on the home page for collectors who are bidding on gemstones and making antique trades?
- I want you to have a story for the app, such as describing that there exists such a person making trades from one item to another and ends up getting a car.
- What about adding a leader-board that shows the weekly craziest trades? For example, someone traded a pair of shoes for rare jewelry that turns out worth a lot.
- For each posted item, is there a quicker way to trade without messaging others such as showing all matched items for the user so they can pick one of them to exchange?

Interviewee 4: 25 years old, male, a local construction engineer

- I would consider this app a 6 on a scale of 1-10 overall.
- Lack of popular features such as item categories, `Save' and 'Share' videos to friends.
- I encountered `Console Error occurred while recording a video' while posting.
- Users should be able to add tags to categorize items.

- Messaging and Inbox features look too simple. I cannot see my profile image, chat history, and other people's information.
- I wish there's a way to match items and trade directly without chatting that much.
- I cannot record video for some reasons so it is hard to say the efficiency. All I can say is that this app is interesting compared with current platforms for buying, selling, and trading used items.

Interviewee 5: 45 years old, female, not in trading previously

- Rated the app 7/10 for convenience.
- Posting videos and messaging were easy, but the UI could be more aesthetic.
- Browsing videos was smooth (7/10), but the UI could be improved.
- Posting was straightforward, and messaging was easy.
- Overall, the app effectively handled posting and messaging.

Interviewee 6: 50 years old, male, not in trading previously

- Rated the app 6.5/10.
- Posting was easy, but more control over video editing (e.g., trimming, quality) was desired.
- Browsing videos was very easy (8/10), but the UI could be improved.
- Messaging was straightforward.
- Overall, the app met the primary use case but could be enhanced.

Interviewee 7: 21 years old, male, not in trading previously

- Rated the app 7.5/10.
- The primary use case was covered, but the app's look and color scheme could be improved.
- Browsing videos was smooth (9/10).
- Posting and messaging were easy and straightforward.
- Overall, the app was effective but needed UI improvements.

Interviewee 8: 20 years old, male, not in trading previously

- Rated the app 6/10.
- The app felt like a shell and needed more features to encourage posting and messaging.
- Browsing was easy (7/10), but the app needed more safety reassurance.
- Overall, the primary use case was covered, but the UI and features needed work.

Interviewee 9: 25 years old, male, not in trading previously

- Rated the app 8/10.
- Posting and messaging were seamless, but better categorization and filtering options were desired.
- Browsing was intuitive but felt cluttered (7.5/10).
- Posting was smooth, but scheduling posts or saving drafts would be helpful.

- Messaging was easy, but quick message templates could improve communication.
- Overall, the app was effective but needed better organization tools.

Interviewee 10: 21 years old, female, used Facebook Marketplace before

- Rated the app 7.5/10.
- The app achieved its purpose, but messaging could be more streamlined (e.g., automatic replies).
- Browsing was easy (8/10), but saving or favoriting posts would be helpful.
- Posting was easy, but cross-posting to other platforms would maximize reach.
- Messaging was straightforward, but more transparency (e.g., active seller indicators) would improve the experience.
- Overall, the app worked well but needed quality-of-life improvements.

Interviewee 11: 28 years old, male, not in trading previously

- Rated the app 7/10.
- Posting felt too basic, and messaging needed better notification settings.
- Browsing was smooth (9/10), but a "related items" section would be a great addition.
- Posting was easy, but previewing posts would be helpful.
- Messaging worked fine, but reading receipts would improve the experience.
- Overall, the app worked well but needed more personalization options.

Interviewee 12: 26 years old, female, used Facebook Marketplace before

- Rated the app 8/10.
- Posting and messaging were straightforward, but safety features (e.g., user verification) were lacking.
- Browsing was fine (7/10), but customizing the feed based on interests would improve the experience.
- Posting was easy, but templates or pre-set descriptions would speed up the process.
- Messaging was simple, but a "negotiation" feature would be helpful.
- Overall, the app worked but needed better personalization and security tools.

Interviewee 13: 53 years old, female, not in trading previously

- Rated the app 7/10.
- The app was smooth to operate but felt incomplete.
- Browsing was good (7/10), but a recommendation algorithm and product classification would improve the experience.
- Posting was easy, but automatic categorization would be helpful.
- Messaging was functional but needed more features.
- Overall, the app was effective but needed functional improvements.

Interviewee 14: 24 years old, male, used WeChat groups before

- Rated the app 7/10.

- The functions were clear, but some features seemed problematic.
- Browsing was smooth (8.5/10), but tags were missing.
- Posting was okay, but posts didn't automatically play after uploading.
- Messaging was basic, but message history was missing.
- Overall, the app was effective but needed feature improvements.
- Leader-board can be a complementary function that combined with the search bar.

Interviewee 15: 25 years old, male, used Facebook Marketplace before

- Rated the app 5.5/10.
- The app was functional but needed optimization.
- Browsing was smooth (7/10), but video tags, favorites, and forwarding were needed.
- Posting was okay, but categories/tags and UI optimization were necessary.
- Messaging was functional, but avatars and message status indicators were needed.
- Overall, the app worked but needed significant improvements. Leader-board is unnecessary.
- You can add it if you want to increase competitiveness.

Interviewee 16: 23 years old, male, not in trading previously

- Rated the app 8.5/10.
- The app needed improvements compared to TikTok-style apps.
- Browsing was easy (8/10), but saving videos and classifying items were needed.
- Posting was smooth, but combining pictures with videos would improve clarity.
- Messaging was okay, but read receipts were needed.
- Overall, the app was effective but needed feature enhancements.
- Leaderboard can be added as a recommendation for users.

Responses on the Survey Questions

In addition to interviews on our Selected Learning Prototype, we also conducted a survey for need-finding and improving aspects of the primary problem.

Here are the questions in our survey:

- Select your age:
    - 18-29
    - 30-39
    - 40-49
    - 50-64
- What is your gender?
    - Female
    - Male
    - Prefer not to say

- What is your frequency of buying/selling/trading USED items?
    - Very Frequently
    - Frequently
    - Occasionally
    - Rarely
    - Never
- What places would you expect to buy/sell/trade used items?
    - eBay, Poshmark, and other online-shopping platforms
    - Facebook Marketplace, Craigslist, OfferUp, and those that can meet up in-person
    - Goodwills and other local thrift stores
- What items you may consider to trade or giveaway?
    - Clothing and Apparel
    - Electronics
    - Books
    - Home Appliances
    - Furniture
    - Collectibles
    - Games and Toys
    - Fashionable Accessories
    - Sports Gears
    - Vehicles, Bikes, and Other Transports
- What prevents you from buying/selling/trading used items?
    - Taking photos, uploading them sequentially, writing descriptions, and repeating this tedious process
    - Paying extra fees such as shipping fees, seller fees, and sales tax
    - Disappointing experiences such as spam listings, fake items, non-delivery, or delivery of wrong items
    - None
    - I actively BST items and I have no concerns
- If you are willing to engage in local trades and giveaways, that would be:
    - Families looking items for children
    - Collectors seeking valuable exchanges
    - Sustainability-focused people who are eco-friendly
    - Students who enjoy thrifting
    - Work professionals need to clear out unwanted items
    - None of above, I would never do these
- Could you briefly share some thoughts or concerns on your experience of buying/selling/trading used items?

Collected survey responses are stored in response.csv.

Sprint 2:

Initial Interviews

All the e-commerce platforms that participants mentioned are reviewed and used as additional resources of previous knowledge. Below are my notes taken after the end of each chat using the *Notes* App by Apple.

Day 1 — Learning commons on campus (2 out of 4 quality interviews)

A 1st year grad student: She sometimes uses Amazon to buy things, and knew other apps such as Ebay and Facebook marketplace but never bought anything from them. She never bought used items from others and she is not familiar with OfferUp and Poshmark. She has/had items that she rarely uses but never sells them because it sounds like a lot of work. Bartering items sounds interesting but she feels lazy taking pictures, writing descriptions, and replying to people.

A 3rd year undergrad student: He recently bought a used guitar on Facebook marketplace, and it was a local pickup. He previously listed his sneakers and old textbooks to sell but it was like "hit or miss". He never sold anything on Offerup or Ebay, because he got used to the Facebook marketplace. For buying used items, he prefers local transactions because chances of good finds, but scammers can be a waste of time. He rarely sells his items because seller fees and the process of shipping. He never trades items with people online and does not know how that works.

Chats that did not go far: One student said he has a lecture soon and she cannot chat for long. Another student said he was working on something and asked if I could do this with somebody else.

Day 2 — At a local cocktail bar (3 out of 3 quality interviews)

A middle-aged man in casual wear: He previously sold used furniture and tools on Facebook marketplace. He also uses Craigslist and Reddit to find good tools and electronics for work purposes. He mentioned sometimes people flake on him or offer really low prices on items he tries to sell. He is not sure if online bartering is a good idea because cash is the king and he needs some money back for valuable things he bought before.

A lady and her boyfriend in business attire: She recently bought clothing from Free People online after trying them on in-store. Both her boyfriend and she knew e-commerce platforms/ apps such as Pacsun, Macy's, Nike, Chewy (her boyfriend has a dog). They prefer shopping malls than online because it is more fun to walk around and find stuff that you really like. The lady believes that it is hard to check the actual quality of an item from pictures, and that is why she returns things she bought online more often than things she bought in person. They both have some old clothes, shoes, and books they do not use anymore, but do not have time to post them online so they rather donate to Goodwill. They think online bartering is interesting, but need moderators to check the quality to prevent unfair situations.

Day 3 — At at Starbucks location (2 out of 3 quality interviews)

A young guy in his street-wear: He often does online shopping because it is convenient. The top 3 apps that he uses the most for buying things are Nike, Ebay, and Depop. He has done some local sneaker trades on Offerup before, and he mentioned that people should be cautious with buying fake shoes. He sold a few things on Ebay but his earnings are low due to all kinds of seller fees. Online bartering is a good idea, but he is concerned about shipping fees and whether users will receive the items as described, or will there be an authentication step? He used to buy and sell sneakers on StockX but it took too long to receive the shoes or funds because of the authentication step.

A middle-aged woman waiting for her daughter: She has Walmart and Amazon apps on her phone for online shopping, but she hasn't used them for a while. She said that her daughter, a high school student, bought some skincares and anime stickers from Tiktok influencers. Things she does not use anymore will either be donated or throwed away because they are broken or damaged. She prefers buying items in person because both shipping and returning take longer time than doing it in-person. Local bartering sounds more appealing than online bartering because exchangeable items are low-valued and shipping could be expensive and time consuming.

Chats that did not go far: The guy was waiting on his order, and he got his order in the middle of the conversation. He said he needed to go somewhere else, so I decided to end the conversation.