

## Summary

The ChooseNXT legacy project is a mobile application that provides a social media platform for a diverse group of fashion brands to efficiently promote various products to patrons and potential customers. This project is sponsored by Tito Chowdhury from FashionNXT. The aim of the project is to redeploy the app in Google Play and App Store in which there are added services and revised versions of the several features that the mobile app provides. The content of the app is specifically focused on fashion businesses showcasing their products, services and events. In addition, patrons of fashion brands are able to keep up to date on their most popular posts through the feed algorithm which populates users' feed.

The deliverables that the sponsor wants to be completed for this iteration of the project are the following: Direct Messaging feature, Login Page revision, Feed Algorithm feature, Releasing New Version of App in Google Play, Feedback from Patrons on Pricing and other UI redesigns and additions. These added features and revisions are necessary to enhance user experience and aid in promoting fashion brands on an interactive platform which allows effective networking amongst fashion enthusiasts and businesses. Upon the request of the sponsor who has the expertise in design, we revamped the design of the app such that it is easy to navigate and visually pleasing to users. Furthermore, the user accounts are divided into two categories which are Patrons and Brand. To avoid making the app look and function like other social media platforms where content can vary from a wide range of topics, the app only allows fashion businesses to populate content of the app with fashion related posts to promote their brands. Below are further details of the features and chores for this project.

## User Stories (54 Points)

### 1. Feature: Multi-Picture Image Display (2 points - Implemented)

#### **As a brand**

So that I can show consumers multiple items/views in one post

I want to be able to upload and display up to 5 images in a single post

No changes to lo-fi from legacy projects. We modified the original story of displaying multiple images in an e-commerce style by splitting it into two stories with the first being displaying multiple images (using the pre-existing partially implemented image slideshow) and the second being displaying those images in the e-commerce style. This was due to the first story being a fundamental change of how the app handles things on the back end with storing posts in the DB and passing the information through the files, while the second story is strictly a UI implementation once the first story is implemented. This gave us more flexibility in implementing other stories that may require more attention and reduces one high complexity story into two lower ones.

### 2. Feature: Display Images in E-commerce Style (2 points - Implemented)

#### **As the app owner**

So that consumers intuitively understand the app functionality

I want to have a single big image with smaller images to view multiple images of a product

(similar to an e-commerce site as opposed to a swiping gallery) so that 90% of users can figure out how to select the next image within 5 seconds



Figure 1: Display Images in E-commerce Style

### 3. Feature: Populate Feed of New Users (2 points - Implemented)

#### **As the app developer/administrator**

So that I can expose to the new users the content that they might be interested in

I want to the feed to not be empty so the app is intuitive and interesting to new users

The Lo-Fi design for this is not needed since it would be the same design as the previous project.

### 4. Feature: Prevent Patrons from Posting (1 point - Implemented)

#### **As the app owner**

So that the feed is focused on brand posts

I want patrons to not have the option of posting

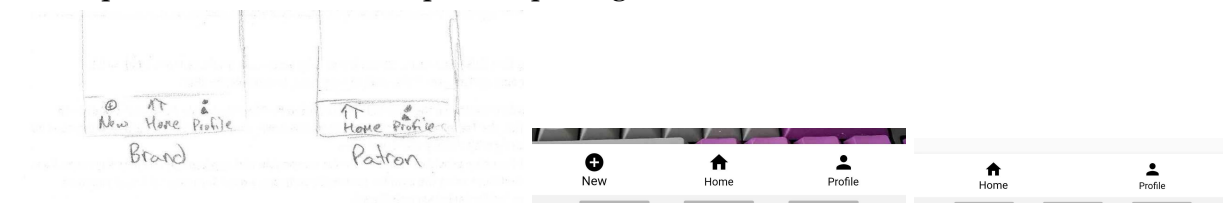


Figure 2: Prevent Patrons from posting

### 5. Feature: Deploy Feed Algorithm (3 points - Implemented)

#### **As the app developer/administrator**

So that I can provide users the content they may be interested in seeing

I want to be able to populate feed of a user through an algorithm that search posts by the following order: brands followed by the user, popularity of brands, filter selected, and time posted

No LoFi Design for this feature since the homepage screen is still the same and only the ordering of the posts is changed.

### 6. Feature: Filter Button (2 points - Implemented)

#### **As the app user**

So that I can choose what category to filter out the posts in my feed

I want the option of narrowing down posts that I want to see



Figure 3: Filter button

7. Feature: Add Category Option to Posts (1 point - Implemented)

**As a fashion brand**

So that I can label my posts into the category it belongs to

I want patrons to find my posts based on the category it belongs to when filtering their feeds

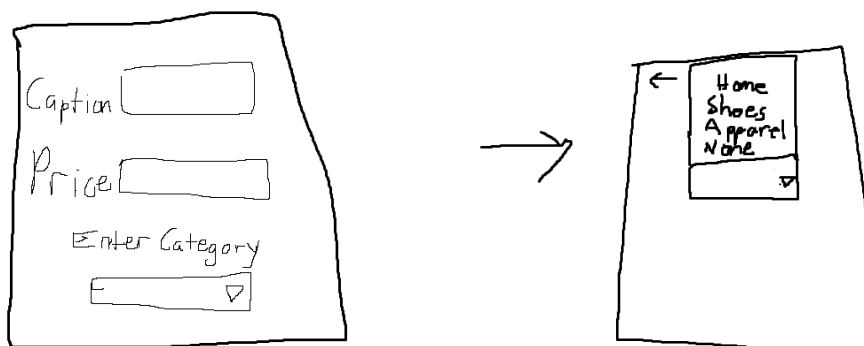


Figure 4: Category Option to Posts

8. Feature: Email and Password Sign up (2 point - Implemented)

**As the app user**

So that I can create an account with my email address with any domain

9. Feature: Email and Password Login (1 point - Implemented)

**As the app user**

So that I can login with my email address and password (Not limited to gmail, apple, and facebook sign-in)

10. Feature: change would-buy button to would-buy slider (1 points - Implemented)

**Description:** One main difference that the ChooseNEXT has in comparison to similar available applications in the market, is to provide the small businesses with an estimation of the price that the market is willing to pay for that product. So, the would-buy-button is removed and instead a would-buy-slider reflects how much the patron is willing to pay for this item/product. is implemented. In our initial discussion with our client this was to span between -20% and +20% of the actual price. This design was subject to changes in following iterations.

**As a fashion brand**

I want to drag a Would-Buy-slider from least interested to most interested based on the current price/product.

So that the fashion company can adjust their price/product based on a more informative feedback from patrons

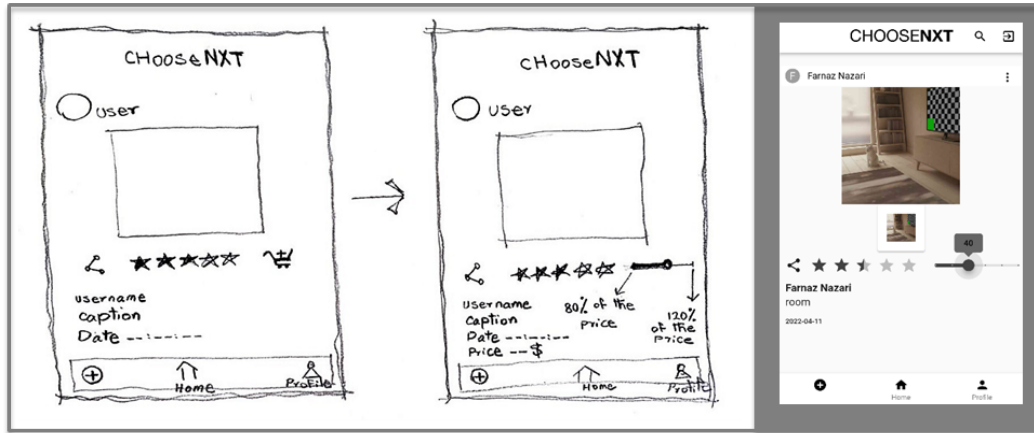


Figure 5: would-buy slider for posts

11. Feature: Customized icon for the price slider (1 points - Implemented)

**Description:** The flutter/dart libraries provide several icons and shapes, however in our case the desired logo or graphic did not exist in the libraries, therefore it took quite a large amount of time and effort to generate an applicable format to be used as the thumb shape for the slider. We provided the price slider with an option of choosing any image as the thumb according to future UI design of the application. No changes to lo-fi from the previous iteration.

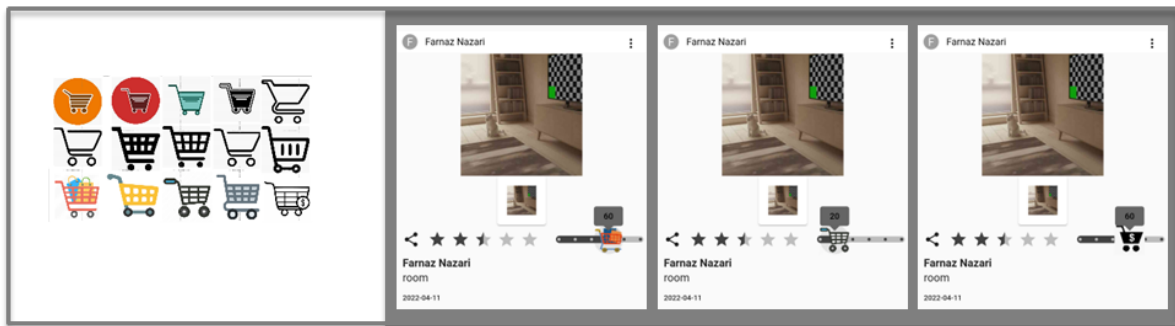


Figure 6: Customized icon for the price slider

12. Feature: Adding price field to the post (1 point - Implemented)

**As a brand**

So that I can enter a price for each product  
I want to be able to set a price once I am posting a new product

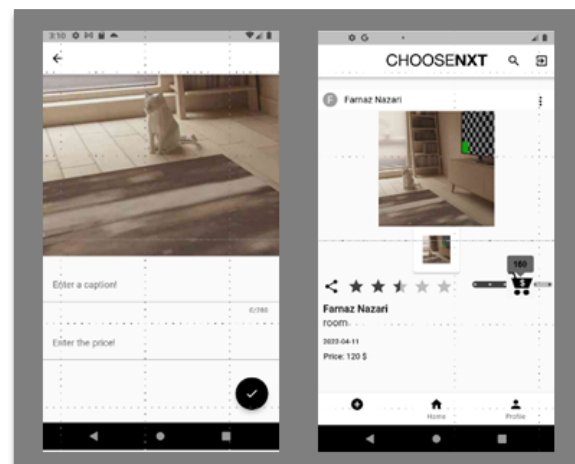


Figure 7: Adding price field to the post

13. Feature: Min-Max Price adaptive slider (3 points - Implemented)

**Description:** Based on our discussion with our client, the idea behind the would-buy-slider changed in a way that the actual price is hidden from the patrons, and instead the sellers enters a minimum price and a maximum price instead, and based on the number of patrons interested in this product and their proposed price, the seller will make the sell decision.

**As a patron**

So that I can see suggest how much I am willing to pay to purchase the item,  
I want to drag the price slider based on my opinion about the product

**As a brand**

So that I can have the option to sell my product at a higher price if it is popular or sell it at a lower price if not many patrons are interested in it.

I want to review the patrons reaction and then decide on the price accordingly.

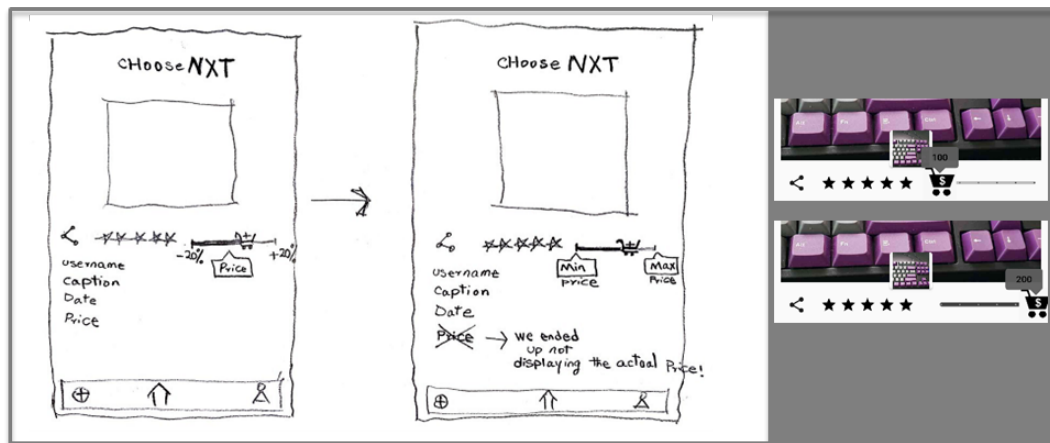


Figure 8: Min-Max Price adaptive slider

14. Feature: Share the app with friends (1 points - Implemented)

**As the app owner**

So that the reach of the app grows

I want users to be able to send an email to friends that includes a link to download the app.

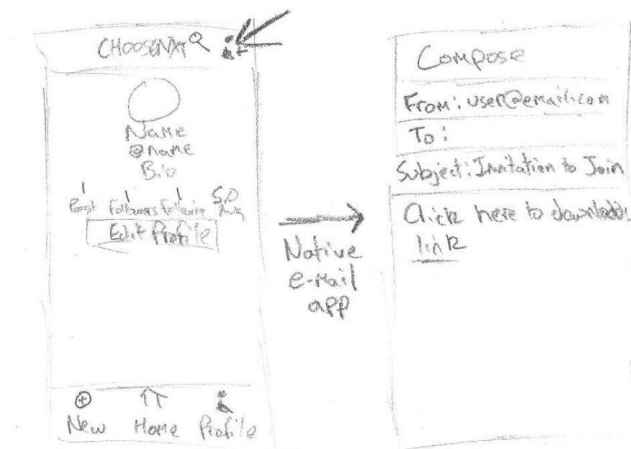


Figure 9: Share App with Friends

15. Feature: Register with Google (2 point - Implemented)

**As the app user**

I want to be able to sign up with my Google account and have a similar sign up experience with other methods

So that the app is convenient to sign up, and I do not re-enter my profile data from Google.

16. Feature: Sign in with Google (1 point - Implemented)

**As the app user**

I want to be able to sign in with my Google account.

So that the app is convenient to sign in.

17. Feature: Register with Facebook (2 point - Implemented)

**As the app user,**

I want to be able to sign up with my Facebook account and have a similar sign up experience with other methods.

Lo-fi design

Implementation

Figure 10: Register with email /password

So that the app is convenient to sign up, and I do not re-enter my profile data from Facebook.

18. Feature: Sign in with Facebook (1 point - Implemented)

**As the app user**

I want to be able to sign in with my Facebook account.

So that the app is convenient to sign in.

19. Feature: Password reset option (1 point - Implemented)

**As the app user**

So that the app has the option to reset my password

I want to reset my password if I do not remember it

20. Feature: Two tiers accounts separation (1 point - Implemented)

**As the app owner,**

I want separate accounts (patron and business) at the registration point.

So that the app provides the appropriate functionality for each account type.

21. Feature: Registration fields (2 point - Implemented)

**As the app owner,**

I want users to insert their unique username, profile name, and URL (if it is a business account) in the registration page.

So that the app asks for minimum and sufficient information required for account creation

22. Feature: Change or remove profile picture (1 point - Implemented)

**As the app user**

I want to change or remove my profile photo from the Edit Profile page.

So that, I have an updated profile photo or I do not have one if I wish to.

23. Feature: Enter Minimum and Maximum Price (1 point - Implemented)

**As a brand**

So that I can enter a price range to evaluate interest in my product

I want to be able to enter a minimum and maximum price when I create a new post

24. Display Minimum and Maximum Price (1 point - Implemented)

**As a brand**

So that I can view what price range I entered previously for a post

I want to view the price range in my profile page

25. Feature: Invite friends through email list/other socials (2 points - Partially Implemented)

**As a user**

So that I can invite my friends to use the app

I want to be able to click a button that shows a list of contacts I can choose from to send an invite link.

This feature was partially implemented in where you can send a preset email to one person but does not scrape the entire email list or use another social. This was due to the customer adding this in the last week and we ran out of time.

26. Feature: DM List (2 points - Partially Implemented)

**As a user**

So that I can get who I am chatting with and enter any chat I have,

I want to be able to click the corresponding user on my DM list and enter the DM screen with that particular user.

This feature was partially implemented where the DM list is present and can be clicked, but the actual DM screen is not. Also, the DM list is not sorted by the time of the last interaction.

27. Feature: Separate entries in DM List to have clients shown on top (1 point - Implemented)

**As a brand account holder**

So that I can access DM chats with my clients faster than other users,

I want to have the DM List separate the conversations I have into those with clients and those with other users, and show my clients on top.

28. Feature: Designate followers as clients or other users (1 point - Implemented)

**As a brand account holder**

So that I can know and designate some of my followers as clients,

I want to have a button on the follower's profile screen that when I click on it, will make that follower my client or vice versa, remove that follower as my client.

#### Chore: Deploy to App Store (3 points - Implemented)

As a product owner

I want to enable users with iOS operating system on their devices to have access to my product  
So that I can distribute my app among a large number of target audience, and make it accessible as well as seen.

#### Chore: Deploy to Google Play (2 points - Implemented)

As a product owner

I want to enable users with Android operating system on their devices to have access to my product  
So that I can distribute my app among a large number of target audience, and make it accessible as well as seen.

#### Chore: Firebase configuration for authentication (2 points - Implemented)

As a developer

I want to integrate the login through email, Google, and Facebook  
So that I need to setup a backend to store all the authentications in one place

#### Chore: Facebook sign in/up backend configuration for iOS (2 points - Implemented)

As a user

I want to have the option to log in to the app using facebook on my apple device  
So that it is more convenient for me to log in and use the app.

#### Chore: MongoDB backend support for conversations, messages, designating clients (3 points - Implemented)

As a developer

I want to query the database to retrieve/update information regarding conversations, messages and designating clients, so I can implement these features in the app. Also, it's preferable to have efficient operations: I want it to take constant time to query a document.

### **Legacy Code**

No member of our team had previous experience with Dart or Flutter so our first step was to get background knowledge and read through some example code and documentation to get used to the style. With a basic understanding of Dart and Flutter, then read through the code for the app the first time and run the app so that we can see how the code works in comparison to what we are doing inside the app. This gave us a strong foundation to work on implementing new features.

In terms of learning the previous code, the past groups have left a large amount of documentation in terms of their iteration reports, meeting logs, and final deliverables. Our first step was to review those to be able to determine what decisions they made, why they made them, and what issues they ran into. We also have the contact information of the previous team, were



able to set up an initial meeting which helped us set up the coding environment and get a basic understanding of functionality of the app.

The majority of the groundwork of the app has been laid out and as such the majority of the user stories that we implemented were refactoring previous user stories or adding quality of life features.

## **Team Roles**

Ehsan Jalilifar: Developer

Delaram Khakzad: Developer

Jialin Li: Developer

Eric Lloyd Robles: Developer

Farnaz Nazari: Developer

Jacob Strickland: Project Owner & Scrum Master

We did not have any changes throughout the course of the project.

## **Scrum Iterations**

Iteration 0:

- Setting up the coding environment. (3 points)

Iteration 1:

- Basic multiple image display. (2 points)

Iteration 2:

- E-commerce image display (2 points)
- Populate Feed of New Users (2 points)
- Sign-up with Email (2 points)
- Registration field basic setup (0 point)
- UI-Design-post screens (3 points)

Iteration 3:

- Separation of brand and patron accounts (1 point)
- Prevent patron users from posting (1 point)
- Deploy feed algorithm (3 point)
- Add Category Option to Posts (1 point)
- Customized icon for the price slider(3 points)
- Firebase setup for Email authentication (1 point)
- Registration page UI Design (0 point)
- Sign-up with Google (2 points)

Iteration 4:

- Firebase setup for Google authentication (0 point)
- Sign in with Email and Google (2 points)
- Scrollable Filter Icons (2 points)
- Min-Max Price adaptive slider (3 points)
- Registration page UI improvement (2 points)
- Password reset option (1 point)

- Registration field completed (2 points)
- Deploy to Test Flight (2 points)
- Deploy to App Store (2 points)
- Deploy to Google Play (1 point)

#### Iteration 5:

- Firebase setup for Facebook authentication (1 point)
- Facebook backend setup for iOS (2 points)
- Facebook sign-in (1 point)
- Facebook sign-up (3 points)
- Change or remove profile photo option (1 point)
- Log-in page UI redesign (3 points)
- Privacy issues were fixed (1 point)
- Testing through Test Flight (1 point)
- Deploy to App Store bug fixed (2 points)
- Deploy to App Store (1 point)
- Deploy to Google Play (1 point)
- Final report including documentations (6 points), poster (3 points), poster presentations (2 points) and demo (3 points)

## Customer Meeting Dates

### Meeting 1: Friday March 4th, 2022 at 3pm CST (pre itro)

In this meeting we had an introductory meeting with the customer and got a feel for what the goal of the project was and set preliminary user stories.

### Meeting 2: Friday March 11th, 2022 at 3pm CST (post itro)

This meeting was after we had gotten the coding environment up and running and were able to discuss the user stories in more detail.

### Meeting 3: Tuesday March 22nd, 2022 at 2:30pm CST (post itr1)

This meeting was after the basic multiple image display was working.

### Meeting 4: Monday April 4th, 2022 at 3pm CST (post itr 2)

This meeting demoed the multiple image display to be in the e-commerce style and we populated the feed of new users with a basic feed algorithm.

### Meeting 5: Thursday April 7th, 2022 at 3pm CST (pre itr3)

This meeting did not demo any new stories but was used as a checkpoint to get a new set of user features to work on for the remainder of the project timeline.

### Meeting 6: Tuesday April 12th, 2022 at 3pm CST (pre itr3)

This meeting demoed separation of brand and patron accounts.

### Meeting 7: Thursday April 14th, 2022 at 3pm CST (pre itr3)

This meeting demoed the feed algorithm with filters.

### Meeting 8: Monday April 18th, 2022 at 3pm CST (post itr3)

This meeting demoed the account creation with email/password and the start of the buying interest slider. A large list of UI changes that did not merit full user stories were also suggested/completed.

### Meeting 9: Monday April 25th, 2022 at 3pm CST (post itr4)

This meeting demoed the share app with friends and the new login and registration fields and UI. There was another list of changes that did not merit full user stories.

## **BDD/TDD Process**

One problem we faced regarding test driven development (TDD) is that flutter and dart do not have any automated testing that we could find. This is in line with all of the past three teams that have worked on this project, which suggested using behavior driven test cases as they could not find any automatic testing.

One major limitation of using behavior driven development (BDD) was that all of our tests had to be done by hand, which was time consuming. There also were risks in-terms of new features breaking previous features. While ideally they would not, due to it being legacy code certain areas were related in ways that we did not expect and that led to features breaking. If all features were not checked by hand each time a new feature was implemented there was no way to ensure that this was not the case.

## **Configure Management Approach**

We first forked the repository from the Fall 2021 team into Woolala2022Spring and then started our development on it. The master branch was always kept stable and whenever a feature branch was completed, we merged the code into master and tested it.

## **Production Release to Heroku**

This project did not include a web app and a web app release. What the client needed was to develop new features and improve existing features for a mobile app, and release the mobile app to Apple Store and Google Play Store. The main issue regarding the release to these two stores was lack of proper documentation from the previous team and the client. The client did not obtain the key to produce a release for Google Play Store and it was very time consuming to find the person who had access to the key. It was also time consuming to understand the process of releasing to the Apple Store and no documentation was provided from the previous group who did release a version of the app for test to TestFlight. Another issue was that many of the packages and dependencies to properly run the application were outdated and some of them have not been updated since the very first group started working on the project more than a year ago. This caused many obstacles in building a functional build for release specially for the Apple Store release.

## **Issues with AWS, GitHub, or Other Tools**

Besides a couple merge issues due to branches being on old versions of the code, we ran into no issues with any tools.

## **Other tools/GEMs used and their benefits**

We used the java keytool to generate .jks keys, which are needed to build the project. These keys should not be uploaded to github, and users should regenerate their own keys. We also had to use Gradle to fix dependency problems as well as acquire the required SHA1 and SHA256 keys for Firebase.

We used Android Studio as our IDE, as it includes support for Flutter, and has mobile device emulators to test our changes

## **Links**

Github: <https://github.com/JStrick510/Woolala2022Spring>

CRM (not in scope of project): <https://github.com/JStrick510/woolala-analytics-app>

Pivotal Tracker (obsolete): <https://www.pivotaltracker.com/n/projects/2556921>

Jira: [CHOOS board - Agile board - Jira \(atlassian.net\)](#)

Slack: [se-team10workspace.slack.com](https://se-team10workspace.slack.com)

Play Store:

[https://play.google.com/store/apps/details?id=com.fashionxt.choosenxt&hl=en\\_US&gl=US](https://play.google.com/store/apps/details?id=com.fashionxt.choosenxt&hl=en_US&gl=US)

Apple Store: <https://testflight.apple.com/join/Mgf4QxrX>

Demo and Presentation: <https://youtu.be/HTAE35F2mnQ>