ChooseNXT Fall 2021 Final Report

Team Laggies:

Ryan Tansey

Corbin Petersheim

Rahul Murthy

Jingcheng Xia

Project Summary

ChooseNXT (of FashionNXT, founded by primary stakeholder Tito Chowdhury) is a mobile application on both Android and iOS intended to create engagement and connection between patrons and fashion brands in order to drive sales and capture market analytics. The app is meant for brands - especially smaller businesses - to gauge interest and price acceptance of their prototypical or small batch products before they invest into mass production, and for patrons to connect with cutting-edge fashion brands and gain exclusive access to their products. A brand user can create posts of a product with various pictures and patrons can express interest regarding the product and its price. Therefore, the patrons can find desirable items at good prices, while the brand can more accurately gauge public interest. This is different from other social media in that it cuts out unnecessary interactions, and ideally, the user interest would more directly correspond to willingness to spend money.

Currently, the product is set where the users all have the same account type, and can follow one another to surface their posts on the main feed (much like Instagram). However, it needs to evolve into different account types (Patron vs Brand) in order to better meet the goals of the ChooseNXT business. Patrons will only be able to follow Brands, and only Brands will be able to post content onto the Patron feeds. This keeps the feed focused and limits the amount of 'noise'. It strengthens the relationship between Patrons and Brands, and facilitates the market analytics goal of ChooseNXT. With this account type structure in place, features that meet the customer's needs can be better implemented in isolation. However, Tito also expressed concerns regarding the speed of the app, such as loading posts and profiles. While added features are improving the app's robustness, it might be optimal to eventually rewrite the app from scratch to address scalability in real world traffic, as suggested by the previous team.

User Stories

Below is a list of user stories we considered, both implemented and unimplemented. All lo-fi sketches, where applicable, as well as screenshots showing the corresponding changes, will be included at the end of the document.

1. **Feature: Create a Patron user account during sign up (3 points, not implemented)**
    - As a fashion fan,
    - So that I can follow, view and connect with fashion brands
    - I want to be able to create a patron user account on ChooseNXT
2. **Feature: Create a Brand user account during sign up (3 points, not implemented)**
    - As a fashion Brand
    - So that I can share my products and connect with potential Patrons
    - I want to be able to designate my account as a Brand during sign up on ChooseNXT

The purpose of these two features is to differentiate brands and patrons accounts so that specific actions can be assigned to each in the future. This will likely require an additional field in the customer database, a database migration, as well as a new account creation process. These two features were unimplemented due to time constraints and the scope of their impact across the application. Note that the features referring to Patron and Brand were made with this feature in mind. Current implementations of new features work on the existing universal "User" account.

3. **Feature: Patron feedback on Product prices (1 point, not implemented)**
    - As a Patron,
    - So that I can provide feedback on a Product's price,
    - I want the Product's price to be visible on the Product post
    - AND be able to select a Yes/No option based on my opinion of the price.
    -
4. **Feature: Set target price for a Brand's product (1 point, implemented)**
    - As a Brand,
    - So that I can let Patrons provide feedback on prices,
    - I want to set the target price for an item when creating a new product post.

These two features enable users to set a price on posts, and allow patron users to provide feedback on these prices. The first has been implemented, but users currently still cannot interact with these prices. This would probably require an extra button to be implemented on the post viewing screen as well as an extra field in the database to store the user feedback.

5. **Feature: Attach multiple photos to single post (3 point, partially implemented)**
    - As a Brand,
    - So that I can provide multiple looks at a single product for Patrons to view,
    - I want to be able to upload multiple photos while creating a new post.

The ability to show multiple photos in a slideshow on a single post has been added. However, there currently is no way to actually upload multiple photos to a single post. Adding this feature should probably be another ticket, as it would require implementing at least another button and another call to the database, as well as refactoring the database methods to deal with list of images, as opposed to single images.

6. **Chore: Change user handle in edit profile (2 points, implemented)**
    - As a User,
    - So that I can update my profile username/handle accordingly,
    - I want to be able to edit my user handle on the edit account page.

Whenever a user creates an account, they are assigned a handle as well as the displayed username. Previously, the handle could have a garbage string generated rather than a proper username handle. This change allows users to change their handles to something more meaningful of their choice.

7. **Feature: User confirmation pop up for block action (1 point, implemented)**
    - As a User,
    - So that I don't accidentally block another User,
    - I want a confirmation message asking if I should truly block a User.

The app has a block button for blocking other users. This feature was a requirement that had to be met to be released into app stores. However, previously, pressing the button immediately blocked the user. We added a confirmation message to prevent users from accidentally blocking each other and ensure the action is more deliberate.

8. **Chore: Image compression on upload (1 point, implemented)**

- As an Admin,
- So that the application is efficient and not wasting resources,
- I want user uploads to be compressed to an acceptable quality upon upload.

Tito mentioned that uploading images would occasionally take a very long time. Previously, the app did not try to compress uploaded images which could be tens of Megabytes with modern smartphone cameras. This functionality has been added, and the app will automatically compress uploaded images to a reasonable size while maintaining acceptable image quality.

Legacy Project Discussion:

Discussion of Existing Code:

This project is written in a programming language called Dart, using a framework called Flutter. Dart is an object-oriented language with C# or Java style syntax, which can be used for both Android and iOS apps. Syntactically, this should not be difficult to understand, but knowing Android development is highly recommended before taking on this project. Flutter, in addition to providing libraries, also acts as a dependency manager and package building tool. The best way to understand Flutter is to watch tutorials and read the documentation at https://docs.flutter.dev/. In Dart, nearly everything is a widget and is a child of or has children components within. Developers then implement the widget behavior based on user interaction. The **lib/** folder contains code cross-compiled for Android and iOS devices. The **pubspec.yaml** file is like a manifest file where all package declarations are made.

In addition, the app uses mongoDB, which is a noSQL data management system. The data is stored as JSON like objects called BSON (https://docs.mongodb.com/manual/core/document/#std-label-bson-document-format) rather than tables, which is different from the sqlite or postgres tables we discussed during the course. Most of the functions for accessing the database already exist, but it is a good idea to understand the structure of the data.

Finally, some aspects of Flutter rely on gradle (https://gradle.org/), so familiarity with gradle can be useful. **Gradle.build** and **gradle.properties** may need modification based on the versions of the Android SDK and Java JDK you are using.

Android Studio is not necessary to develop in, but is likely to be the IDE most will use when developing APKs and using Flutter/Dart/Gradle. Android Studio allows you to run an android device emulator which is convenient and necessary if you do not have an Android device physically. Being familiar with how to set up Android Studio and navigate the UI would be helpful. For iOS development, XCode is required for building the IPA iOS app file. Adhering to the official Android and iOS guidelines is required if you want to deploy the app to the Play Store or Apple TestFlight.

Code Refactoring/Modifications

We did not perform a significant refactoring of the code besides what the stories mentioned above. However, aside from the code itself, we had to make several modifications to files such as build.gradle and pubspec.yaml in order to address configuration issues. Although the previous team left some documentation on this, much of it was either outdated or vague leading to confusion. We will leave a modified environment set up document expanding on the build process to mitigate the issues we faced.

Team Roles

Member	**Role**
Ryan Tansey	Scrum Master
Jingcheng Xia	Developer
Corbin Petersheim	Product Owner
Rahul Murthy	Developer

Although we had plans to have more defined team roles, due to some circumstances with

the initial project, we had to work together to resolve some unexpected issues, making the distinctions less clear. The assignment of the product owner role was Corbin Petersheim and he made initial communications with our first customer and with Tito, and kept communication with Dr. Walker and Gayathri during the process. The role of scrum master can be attributed to Ryan Tansey, as he was the one to submit the various iteration reports. During our meetings with Tito, all members of the team took part in communicating with the customer. With only four members on the team, the lines between roles blurred a bit.

Iteration Accomplishments

Iteration 0:

- Met with the customer, planned out some stories, and began setting up the development environment.

Iteration 1:

- Resolved dependency issues to build the project and successfully set up the development environment. Recorded necessary changes for future teams and began looking at some feature stories.
- APK could be built at this point, but progress was blocked by app login issues, and there are no ways to bypass the login screen of the app, even as developers. Facebook and/or Google authentication was required to access the rest of the application and see any changes to the code. Fixing this was the next top priority in order to unblock development.

Iteration 2:

- Solved the login issues with Google Firebase. Successfully built and logged in to the project.
- Continued making progress on features and chores.

Iteration 3:

- Continue working on features and chores

We had an extremely delayed start, due to our initial assigned customer (Big Event committee) dropping a project thus having to be reassigned. Therefore, some of our iterations were cut short and timelines shifted to accommodate. During the iterations, we also ran into several configuration issues within the project that we had to resolve before being able to work on any stories. In addition, we suffered from a lack of manpower, as our group was smaller than the expected minimum due to one of our group members dropping during the semester without informing anyone ahead of time.

Customer Meetings Summary

Date	Summary
10/28/21	Initial meeting with Tito Chowdhury, the customer. Looked over the purpose of the app, ChooseNXT, as well as the current state. Reviewed potential tasks and created some user stories.
11/4/21	Discussed the issue building the application that was blocking our progress. It was determined that the previous team most likely used a previous Flutter version. Also set up a meeting with the previous team to resolve build issues.
11/18/21	Demonstrated the configuration which let us successfully build the APK, and made plans regarding solutions for future teams.
11/23/21	Meeting shifted due to the Thanksgiving holidays. Discussed issues with building the app on the iOS side. Ryan has an M1 Macbook Air, but is unable to build an iOS ARM app for ChooseNXT and run it natively on the laptop. Without an iOS device locally (simulator would not work), this was an issue. An iPhone 13 was en route from Apple, but UPS lost the package, so a replacement is being sent. Also discussed issues where neither Facebook nor Google authentication was working despite the steps followed in the environment setup guide from the Spring 2021 team.
12/2/21	Demonstrated the ability to log in to the app, and discussed finally working on

	some of the planned features. In addition, Tito requested some new tasks to look into during development.
12/9/21	Met with Tito to demonstrate progress on the requested features. Discussed plans for final coursework (presentation, demos, etc) and logistics for wrapping up the project.

BDD/TDD Test Process

We used the BDD process to create user stories based on feature requests from Tito for the next iteration of the ChooseNXT mobile app. The goal was to have a dialogue with the customer on top features and how to best align the business goals with development, and thinking strategically about the user experience. A challenge with this process was fully understanding the client's business goals and how the requested features facilitated those goals. For example, if the application is about market research and gauging interest from patrons, how should information be displayed to the user regarding price and how should feedback be given? Is this meant to be social media or eCommerce?

Regarding TDD, since the project was not written in rails, there are no cucumber or rspec tests. However, there is a test.dart file containing existing unit tests which we could test against. We did not implement any new tests. Some of the tasks we completed were fixes, which did not require any additional testing. We chose to complete smaller tasks due to a lack of time caused by some unexpected issues, but this also means that we did not implement many testable features.

Configuration Management

We used Github for version control. At the start of the project, we forked a repository off the previous semester's team repository. After that, each member would create a new branch for any story we worked on. Once a story was completed, we would merge the ticket back to the main branch. We did not have any spike stories.

We did not have any major releases. A majority of our time was taken either due to logistical issues, or resolving blocking issues in the project unrelated to new features. As such, most of our completed tasks were chores that did not warrant a new full release, but the features added were merged into the master branch at the end.

Releasing to Heroku

The previous team had issues with Heroku speed due to using a free tier of Heroku. However, we were using a paid tier of Heroku from the start, and we did not deploy any new versions to Heroku, so we did not encounter any issues. No changes were made to the Heroku configuration used by the previous team. The URL from the previous team was (hidden-caverns-85596.herokuapp.com) and https://woolala-analytics-app.herokuapp.com/, but it is unclear looking at the code or final report from Spring 2021 what these applications are used for and how the application interacts with them.

Issues With Tools

.jks keys must be added into the project folders, but should not be uploaded to Github. There are multiple projects in Firebase, and SHA1 and SHA256 keys had to be added to all of them, not just the initial project or the 2021 project the Spring 2021 team used. We had major issues with Flutter and Dart versioning, as there were major updates to packages as well as the framework/language itself between when the previous team finished in Spring and we began in late Fall. Many functions were either deprecated or renamed, or received additional arguments, so we had to hunt for the appropriate versions to use.

Running “flutter build apk” on a fresh pulldown of the Spring 2021 project, the command line did not provide any helpful information as to what the problem was. The resulting error was not searchable on Google or StackOverflow which led to a lot of churn and confusion. Eventually we were able to resolve package dependencies and other gradle/java version issues to get the build to successfully complete.

Other Tools

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 (generating these keys with keytool as recommended by the previous documentation did NOT work).

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

Project Links

- Pivotal Tracker: https://www.pivotaltracker.com/n/projects/2536019
- GitHub: https://github.com/CDPete/Woolala2021CF
- Slack: https://app.slack.com/client/T02J9F2CC06/C02K5TU8ZQQ
- Heroku for Woolala user app: https://hidden-caverns-85596.herokuapp.com/
- Heroku for analytics app: https://woolala-analytics-app.herokuapp.com/
- Testflight link: https://testflight.apple.com/join/pTevnoC7
- Play Store link: https://play.google.com/store/apps/details?id=com.fashionxt.choosenxt

Note that the last 4 links were from the Spring 2021 team. This iteration did not deploy the app to the App Store or Play Store. There is no App Store link because Apple will discontinue apps if inactive over a certain period of time. Google will let the app remain.

Repository Content Discussion

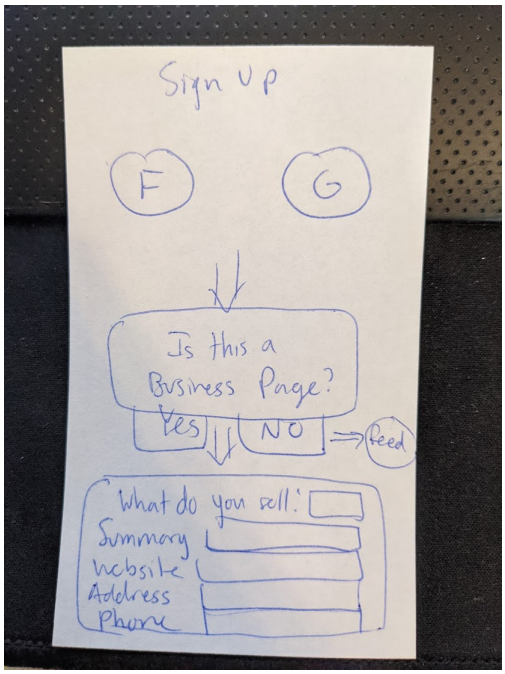
Building the app requires a .jks keyfile, as well as an associated key .properties file. These are NOT uploaded to the Github repository, so any user who clones the repository and wishes to build the project must first generate these via Keytool. We will leave an additional document called updated_setting_up_dev describing the setup process in detail.

Poster and Video Links

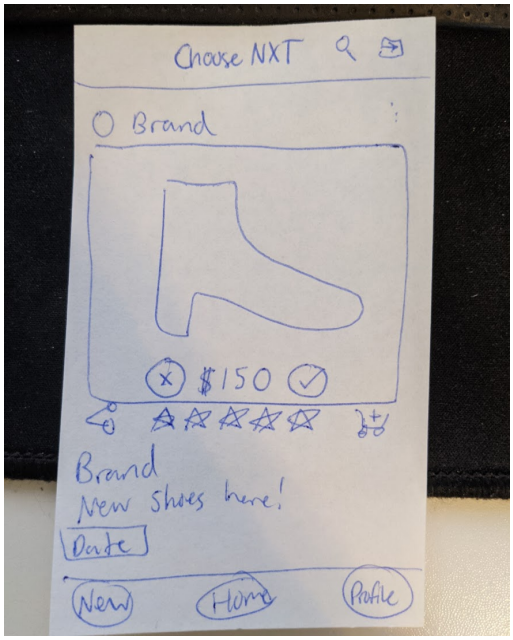
- Poster: https://docs.google.com/presentation/d/1Z-s1KyTWE2HRI_GFM930J_VNC5xu2W1lw3HxiZUhCR4/edit#slide=id.g1072206d2a2_0_122
- Video: https://drive.google.com/drive/u/1/folders/1Dh-pImLJ7yNiAUJKXbd4-1wYb8QN8sUE

Lo-Fi sketches and screenshots

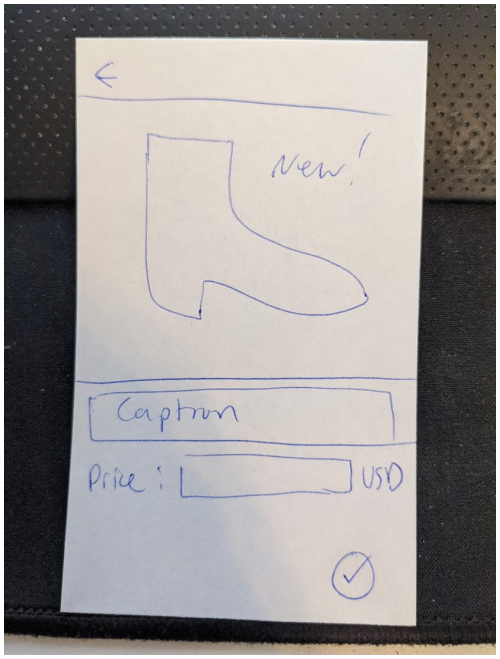
User sign up as Business vs Patron:



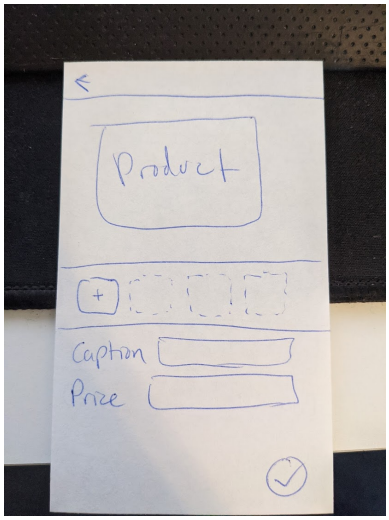
Patron view of product post with target price:



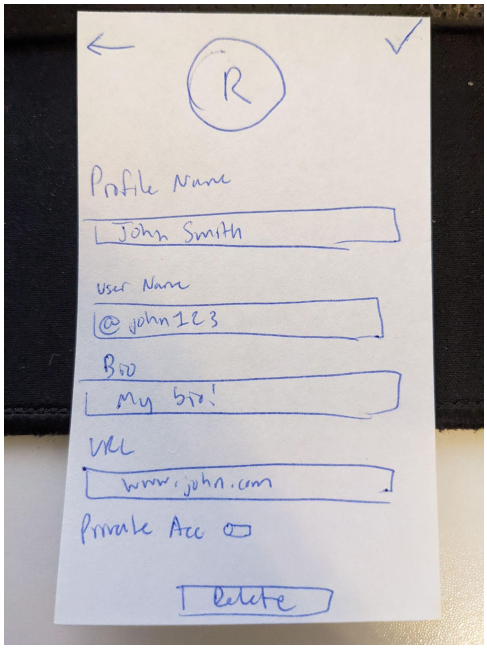
Brand view of setting price for new posting:

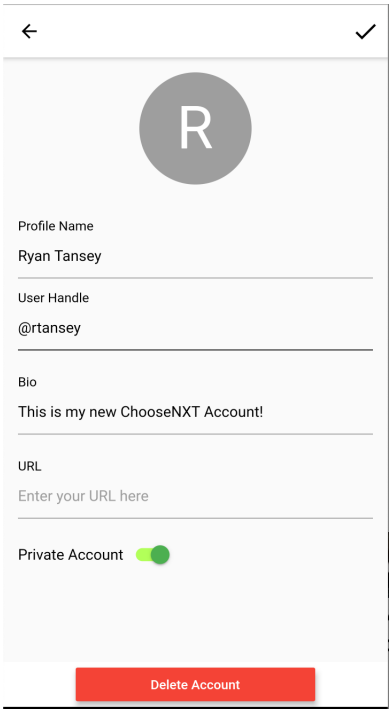


Uploading multiple photos for a post for a slideshow:



Edit profile username / handle:





Block user confirmation popup:

Slideshow view screenshot: