# Document for Future Team

## Issues faced during dev env setup: (Explained in detail in ‘Setting_up_dev_env’ file PLEASE READ THIS)

- KeyStore requirement
    - Android apk build needs a key to sign the app
    - All release versions and login integrations require the key hash.
    - New keystore requires additional hashes to these existing ones.
- Android issues
    - Keystore key path need to be set
    - Create a key.properties file inside the Android folder and add the keystore details.
- iOS issues
    - If you get errors on pod installs not working, you should delete the pod.lock file as well as the pod files within the app’s ios directory. Then run ‘flutter clean’ and ‘flutter run’ again
    - iOS development requires a Mac
    - Certain features do not work without an actual iOS device (aka will not work on the iOS simulator). These features include uploading or taking an image, sharing to social media (b/c social media apps are not installed on the iOS simulator), and sending mail from the DM feature (b/c mail app is not installed on iOS simulator)
    - It is better to run the app on a physical device
    - Sometimes you might get an error that says “iOS version xx is not compatible for this release” - you just do “flutter build --release/debug” and then “flutter install” to resolve this issue.
    - **IMPORTANT: The current GitHub app version has two info.plists → info-DEBUG.plist and info-RELEASE.plist. When running ‘flutter run’, the debug plist will be used. When running ‘flutter run --release’, the release plist will be used.**
- Flutter issues
    - “Flutter/dart sdk not set in local.properties” is a common error
    - Locate the flutter installation directory on your system
    - Paste the path under flutter.sdk or dart.sdk
- Insecure http Error
    - If you are attempting to run a local node.js server on your computer and you are running into an error that is either not allowing for insecure http or spitting out an “Unrecognized character at 0 <Doctype HTML>,” do the following.
    - For Android, see this link: https://github.com/flutter/flutter/issues/66275#issuecomment-710161329.

        - Add the code- android:usesCleartextTraffic="true"- to the path: under line 14 of android/app/src/main/AndroidManifest.xml
    - For iOS, you must set some variables in your info.plist
        - https://stackoverflow.com/questions/64197752/bad-state-insecure-http-is-not-allowed-by-platform

## Performance Issues

Currently, there are many aspects that contribute to the slowness of the overall application

- Naive querying: when searching for all users (aka search page), the app will request for **ALL** users in the Users collection to be sent to the app
    - With small dataset, this is not an issue, but it does not scale well
- Large data document per user
    - Profile pictures take up the majority of the user's document in MongoDB even after compression
    - Suggest splitting the user document and profile picture into two collections and only query for the user's profile picture when needed
- Slow MongoDB cluster -- the current M10 database is usually meant for development environments. For production environments, an M30 or higher is recommended
- Loading circles give a feel of slowness: suggestion is to eliminate loading circles and let the Futures build themselves from blank spaces (this is not a fix for performance but rather a trick into having the users *think* the app is not as slow)

## Bugs that still need to be fixed:

- Cannot re-rate a post.
    - Not sure if this is an actual bug or a feature that was mis-interpreted
- Back button saves the unwanted changes
    - While editing your profile, once you select a profile picture, you cannot deselect it or come back from it.
    - Both the apply and the back button on the app saves your changes.
- Users should get a chance to get back from an unwanted change if they have change of mind. Encoded usernames when certain users cannot change their username
    - During account creation, the user is assigned an encoded username based on their email string
    - After accepting the EULA page, the user account is created and the user is sent to a page where they should change their username

    - If the user gets to this page and exits the app, or the app crashes and they relaunch the app, they will not be able to change their username anymore
    - This causes users to have a “weird username” which is actually just their email encoded in base64.
    - Remember that a user cannot change their username after they leave the “Change username” page.
    - Possible solutions to this are to allow all users to change their usernames after account creation or to **guarantee** that users see the “change username” page even if the app crashes or they exit the app.

## Future features/works

- Search by Hashtag
    - First, there is currently no way to search for posts at all, only search by users. This is the first step of implementation.
    - Add ability to tag users that appear within posts.
    - Add new fields to the database for posts to allow for searching by users or hashtags.
        - This falls somewhat under the slowness of database queries right now as well. Each search pulls the entire user document for every single user, and hence pulls all of their posts as well.
        - Adding fields for hashtags and users that appear within the post’s image, when each post may and- let’s be honest- will have multiple to many hashtags or tagged users will result in further slowness of the database if current query structure persists.
    - *Implement searching of posts by hashtags.*
    - *Implement searching of users or posts by tagged users.*
    - The italicized features listed above must follow after implementation of all of the above initial changes due to slowness concerns of the database if not changed first.

- Would Buy screen sliding price bar
    - Currently the app does not have any concept of *price.* Developers have to introduce this concept by adding a new column in the database first.
    - In the app itself, the “would buy” (shopping cart) button needs to be expanded to a sliding “price willing to pay” feature where users slide to indicate how much they would be willing to spend for the product portrayed within the post.

    - A further concern is maintaining the accessibility of post owners to access the list of people who have expressed willingness to buy. Currently, this is accomplished by pressing and holding the shopping cart button, but may have to be changed.
    - Finally, there is no way for current post owners to know they have to press and hold for the list of interested buyers. The knowledge of how to get to this list needs to be made more accessible.
- A real DM feature
    - An in-app direct messaging feature requires a new db collection that saves the users' message between each other
    - Requires a new page with messages to users as well.

- Use of camera for Profile picture
    - Currently, users can only choose an image from gallery or drives/storage
    - Users cannot use the camera to upload a profile picture.
    - This feature is already there for uploading a post. Developers just need to add it to the upload profile picture section.

## CRM

- Front end: Improve data visualization for the admin
- Back end:
    - Add recommendation features (i.e. suggesting users posts that are similar to what they "would buy")
    - Notify users whose accounts are suspended.
    - Notify users whose posts are reported.
    - Auto record user count at the end of each day.
    - Select the most popular post based on interaction.
    - Select the most active users .
    - Integrate with the user app

## Deployment

- The Spring 2021 team successfully cleared all requirements made by Apple and Google to deploy to their respective application stores.

- The current app has not been distributed yet in these stores because it is feature-incomplete and still has many things to work on, however, the Spring 2021 team has helped clear the legal obstacles of App distribution in the App Store and Google Play store.
- This app has user generated content, what does that mean?
    - We are under legal obligation to somehow moderate the content that users post (aka no NSFW posts) if we are to deploy to the App Stores
    - There must be features within the app that can help users protect themselves from unsafe content and also ways for developers to moderate the content
- Steps we took to achieve this:
    - Created a EULA (end user license agreement) for new members to agree to
    - Created a report/flag posts feature where users can report unsafe content
    - Created block button within profile pages of users so that a user can block all content from a specific user
    - Created an administration app that allows admins/developers to see certain posts and mark them for deletion
- After implementing these steps, we were able to gain approval by Apple and Google to distribute the app to the App Store and Google Play Store. The future team now must maintain these features and make sure they still work in future iterations of the app. Think of this as peace of mind that this app will be accepted into the app store, and you no longer need to worry about making features not specifically asked by the client.