

## Project Handover

### Project repos link:

<https://github.com/ebullion-tech/ebullion-mobile>

### Command for Building ipa and app bundle:

```
flutter build apk --flavor dev -t lib/dev.dart --no-tree-shake-icons  
flutter build appbundle --flavor prod -t lib/prod.dart --no-tree-shake-icons  
flutter build ipa --flavor dev -t lib/dev.dart --no-tree-shake-icons
```

### App store Credentials:(also has to update the app version in api using postman)

App certificates to build and upload ipa.

[sanjeevi@ebullion.in](mailto:sanjeevi@ebullion.in)  
eBullion@2024

### Google play console credentials: (also has to update the app version in api using postman)

Keystore file to build and upload apk

Credentials with tech lead.  
Neelam Gali  
9021379633

Api to show update dialog for user: (Update this soon after app upload on store so that user can update app)

### Production

Android/Ios: <https://api.ebullion.in/create-mobile-version>

Body:

```
{  
  "deviceType":"Android", //change name to ios  
  "versionNo":"1.3.2",  
  "releaseType":"Production"  
}
```

Bear Token : from <https://secure.ebullion.in/signin>

Get version from api: <https://api.ebullion.in/get-version?deviceType=Android> // change name ios

## QA

[//get details](https://qa-ebullion.in/get-version?deviceType=ios)

[// create version](https://qa-ebullion.in/create-mobile-version)

### **Test Credentials:**

B2C: You can also use your account for testing on qa

Username: 9930385938

Password: Test@123

B2B:

Username: AAACH1111J

Password:Test@123

### **Branches for code:**

prod\_new - Production code all release builds are created from this branch.

qa\_new- Once a feature is complete, a build is created for Manual testing from this branch.

dev\_new- this branch will be used to create a feature branch and then merge back into dev\_new.

### **Project Overview:**

1. Account creation for B2C Customer flow and creating Mugdha's account.
2. PAN verification for user , PAN card verified.
3. Bank Details Updation in profile.
4. Buy process for metal, bought metal (Gold in rupees).
5. Provided overview on Payment Gateway(Razorpay) for UPI, Netbanking, card payments.
6. Account Creation for B2B Needs to be done from Website.
- 7.

### **Login Process Overview:**

1. Log in using Password and Username.
2. Login in with OTP and Mobile number.
3. Store credentials and login with Biometric process.

### **Buy Metal Process:**

1. Selecting Metal from Dashboard.
2. Amount of metal to buy.
3. Quote generation for metal.
4. Payment through wallet , UPI and netbanking.
5. Sell Metal Process to and get money to the bank account.

### **Create SIP Process:**

1. Selecting weekly or monthly SIP.
2. Metal selection, amount addition.
3. Also Showed the manual pay options and E Mandate for payments.

### **Enterprise Account and Specifications:**

1. Showed login of enterprise account.

### **Gifting Feature:**

1. How to send Gifts to users.

2. Received gifts to collect in the user's account.

#### **Update pop Up Dialog Configuration:**

1. Showed for both IOS and Android.
2. Api will be shared in this document during code KT.

#### **Deep Understanding on Below Module:**

3. Went through the code structure for the Auth feature.
4. SIP code structure understanding and API, model, state, notifier, details shared with mugdha.
5. Account creation API used and how AWS cognito is used for user Authentication and user management.
6. Routing, and navigation structure used in the app.
7. Biometrics credential store in app feature given detail KT on same.

#### **Deep Understanding on Below Module:**

8. Gifting feature and its folder structure.
9. Razorpay for payments how to initialise and handle payments.
10. Given deep understanding on profile update
11. Price chart dependencies and its usage along with apis used and models used in feature
12. Profit loss data api integration and folder structure and its implementation.

#### **Procedure For IOS app deployment:**

13. Once a feature is completed merge feature branch in dev\_new branch.
14. Update the version number and build number in Xcode by opening code in Xcode, from general options.
15. Then move dev\_new branch to qa\_new branch, and proceed with testing.
16. Move code from qa\_new merge into prod\_new and build ipa using the command  
flutter build ipa --flavor prod -t lib/prod.dart --no-tree-shake-icons.
17. From finder go to the location to find Runner.archive, click on it to distribute the app.
18. Then go on hitting next, no need to modify anything.
19. After completion of the process in a few minutes the app will be pushed on testflight.
20. Then create a new version of the app by clicking + symbol on the dashboard.
21. Add build from testflight and save the content then click on the Submit review button.
22. After this update version in <https://api.ebullion.in/create-mobile-version> so that users can get a dialog to update the app.

