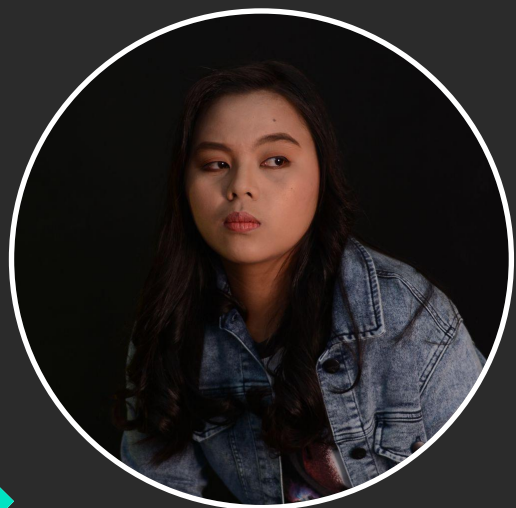




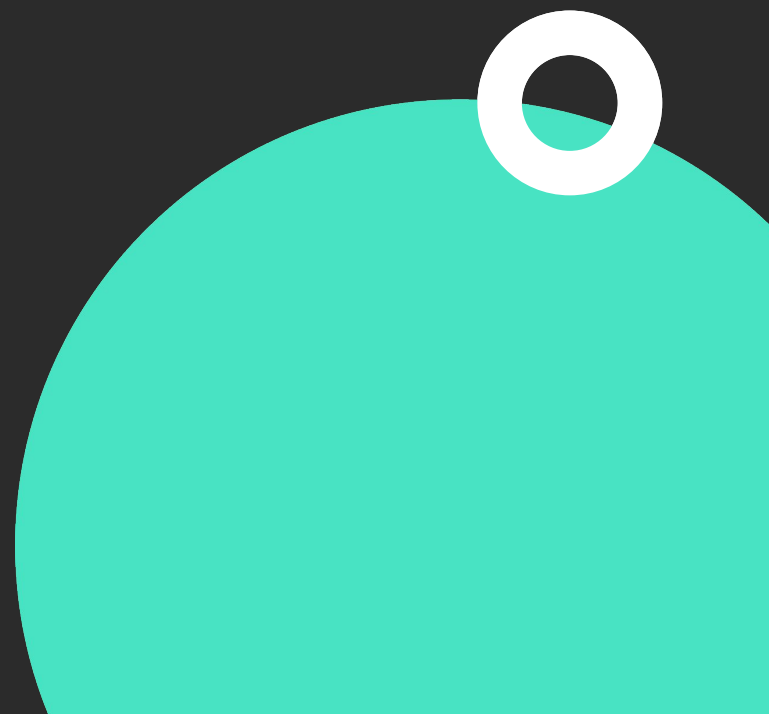
# Guide to Google Play In-App Reviews API with Android Jetpack Compose



Loreleen Mae "Yoye" Sablot

 @yoyesablot

 in/loreleensablot



# Course Prerequisites

- Published **app** in **play store** or;
- Have your own **Google Play Console Account** (even you don't have an app published)



# Course Overview

- Getting started
- Setup the **dependencies**
- Add the **InAppReview** Class
- Implement the **review** dialog
- Handle **Review Flow Completion**
- How to test your app via **google play console**
- **Troubleshooting** common errors
- **Coding challenge**



# What is Google Play In-App Review API?

- Tool provided by Google
- Prompts users for app ratings and reviews
- No need to exit the app interface
- Eliminates the inconvenience of switching to the Play Store

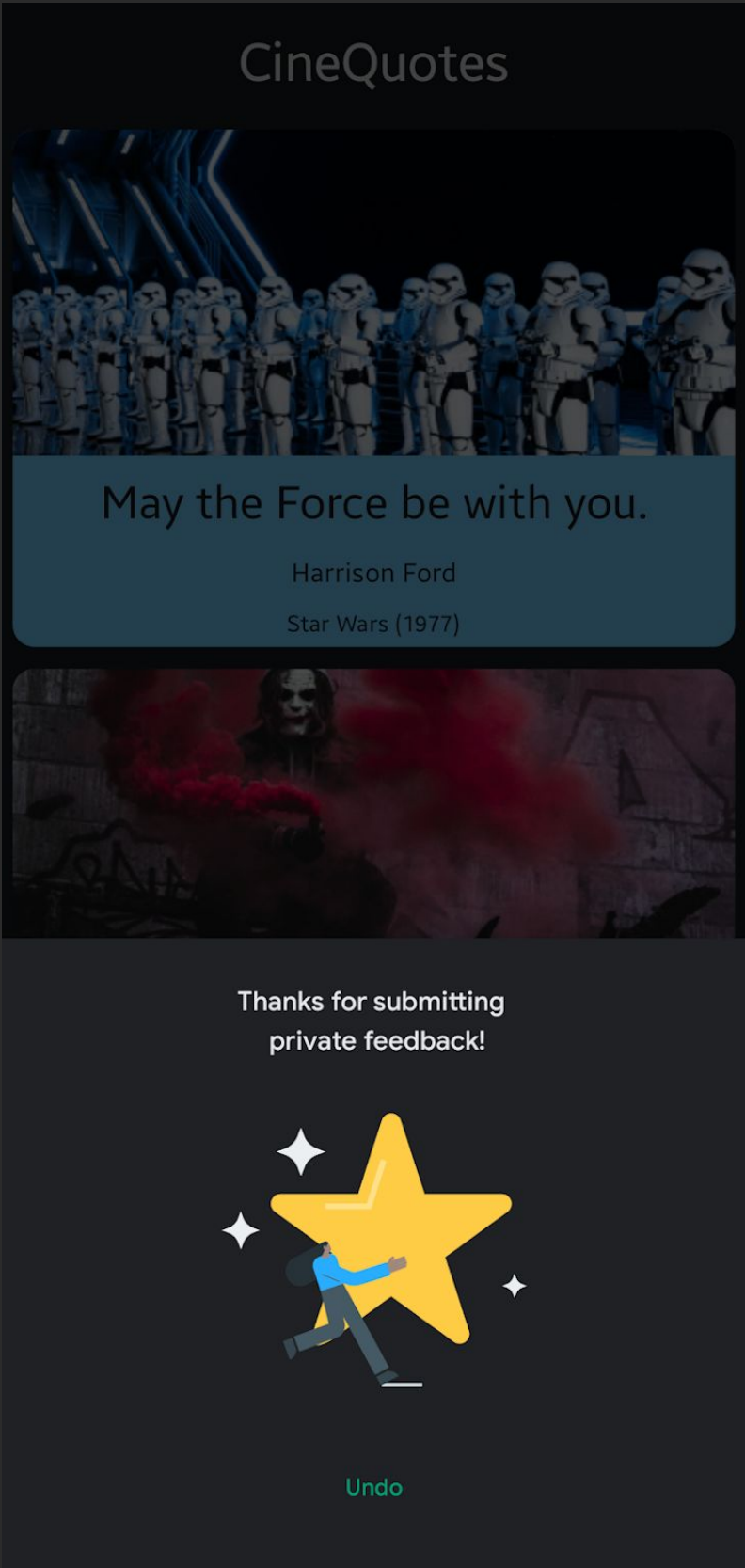
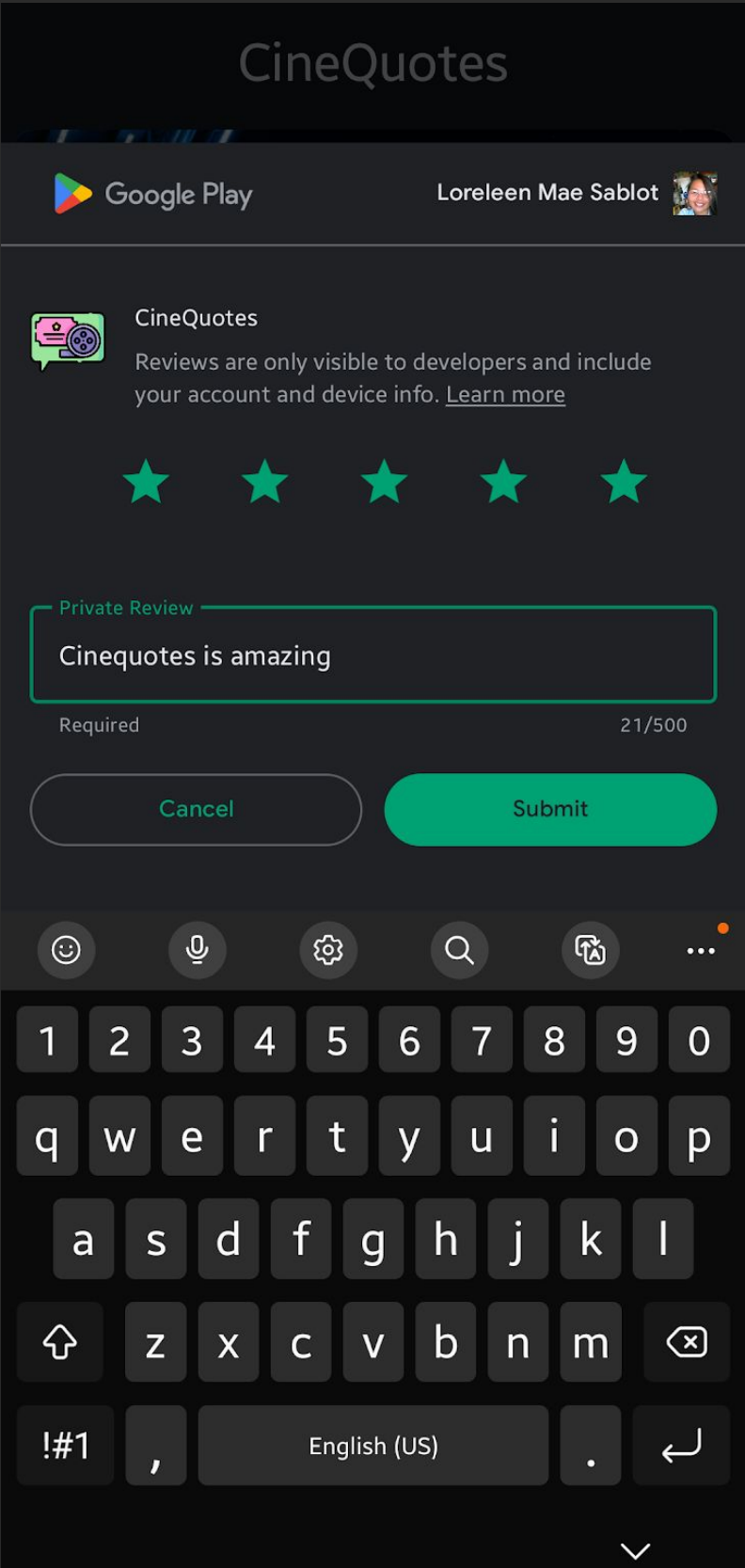
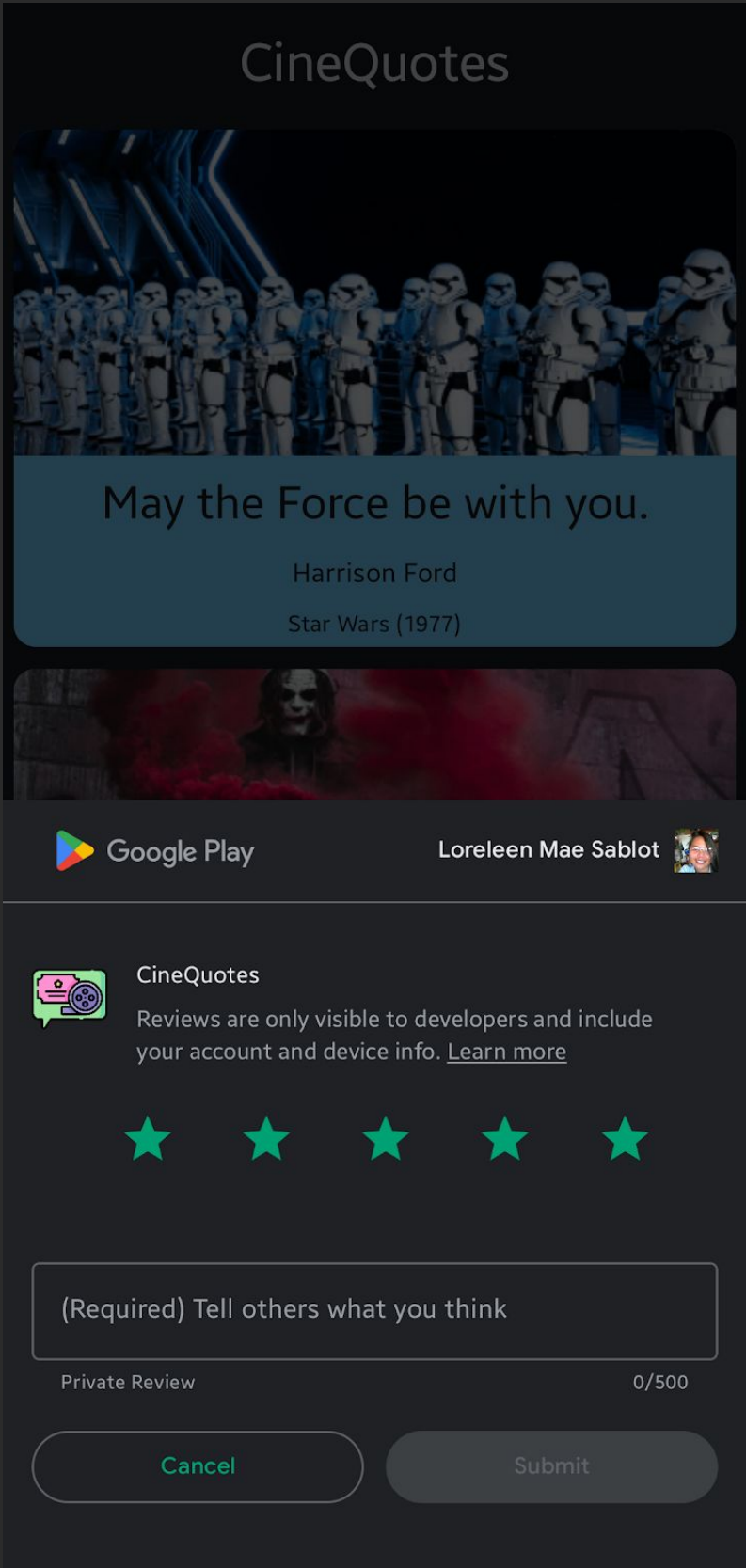
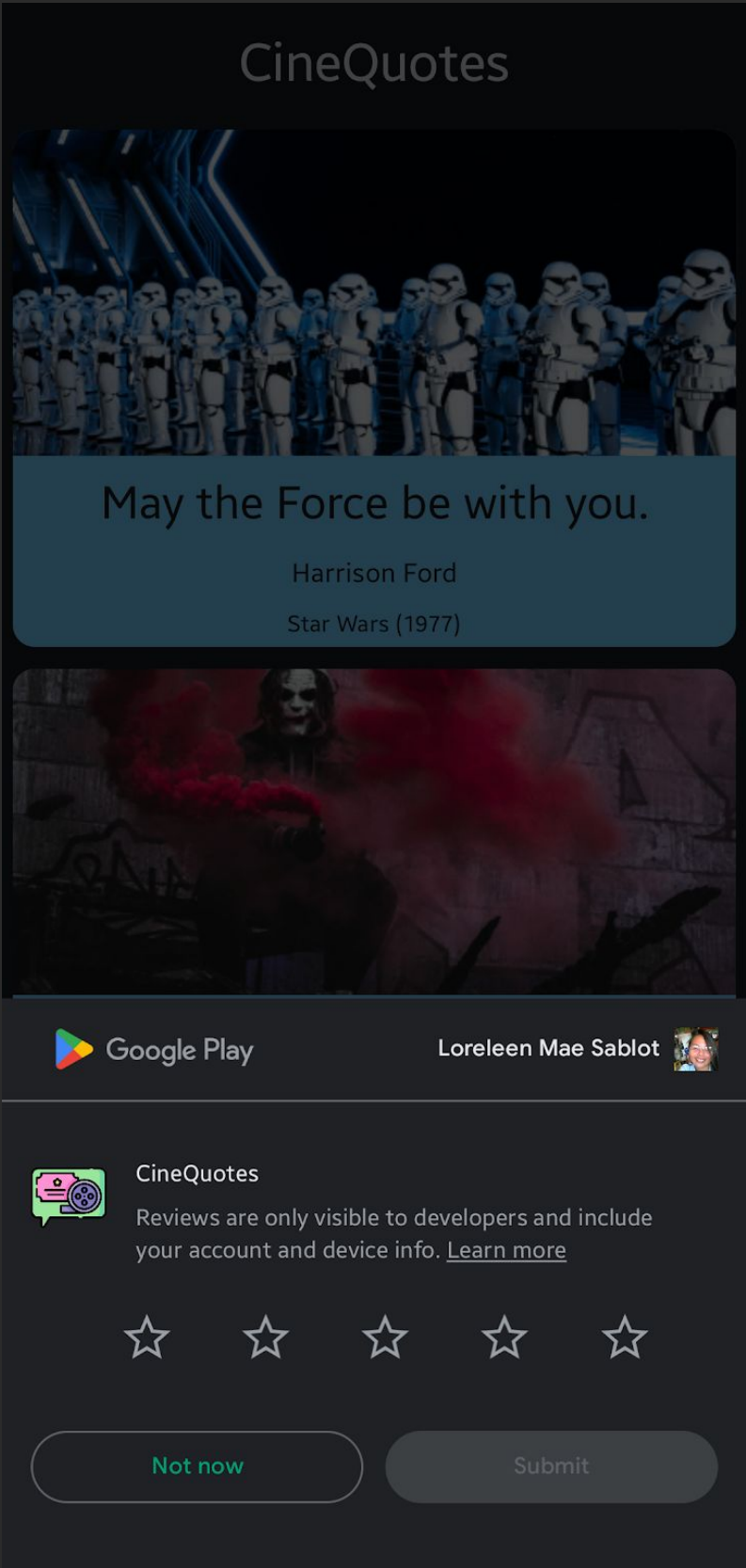


# Why use Google Play In-App Review API?

- Seamless Experience
- Maintained engagement
- Efficient Implementation
- Developer control
- Guided User Flow
- API Support



# Sample In-App Review Flow



# Get the Starter Project



# Right way to request in-app reviews

- Timing
- Frequency
- Questioning and Design
- Overlay and Manipulation
- Quotas







# Add the `InAppReview` Class



```
class InAppReview {  
    fun requestReview(activity: ComponentActivity) {  
        val reviewManager = ReviewManagerFactory.create(activity)  
    }  
}
```

Add the ReviewManagerFactory



```
class InAppReview {  
    fun requestReview(activity: ComponentActivity) {  
        val reviewManager = ReviewManagerFactory.create(activity)  
        val request = reviewManager.requestReviewFlow()  
    }  
}
```

Add the `requestReviewFlow()` function from `reviewManager`





Request a **reviewInfo** object





# Handle and Launch the Review Flow





Let's add the `InAppReview` class in our `MainActivity`





# Helper Files





# How to test your app via google play console







# Test in Google Play Store





# Test using Internal App Sharing





# Test using Internal Test Track





# Troubleshooting common errors



# Issue: Your app is not published yet in the Play Store.

- **Problem:** Unable to review an unpublished app.
- **Solution:** App doesn't need to be published for testing. Ensure app's application ID is available in internal testing track.



# Issue: The user account can't review the app.

- **Problem:** Reviewing not possible for some users.
- **Solution:** App must be in user's Play Store library. Download using the account you want to review with.



# Issue: The user has already reviewed the app.

- **Problem:** Need to edit or delete an existing review.
- **Solution:** Edit or delete reviews directly from the Play Store.



# Issue: The quota has been reached.

- **Problem:** Unable to review due to quota issues.
- **Solution:** Consider using internal test track or internal app sharing.





# Issue: There is an issue with the Google Play Store or Google Play Services on the device.

- **Problem:** Play Store or Play Services not functioning properly.
- **Solution:** Use a device with valid Play Store and Play Services versions.





# Code challenge



# Handle `ReviewErrorCode` of Review API

- Handle the `ReviewErrorCodes` in the in-app review flow



# Summary

- Understood what is Google In-App Review API is
- Added dependencies for the api
- Created InAppReview class for implementation
- Added the ReviewManagerFactory
- Added the Review Flow in the app
- Able to upload your unpublished app in Internal Testing Track or Internal App Sharing
- Test the user flow of the api

