Harmony

Deployment Plan

Ali Doğan Çamur 231101017

Savaş Solak 231101025

Ege Azca 231101057

Yankı Yağız Ozan 231101059

Table of Contents:

Task Matrix:

- Deployment Overview and Deployment Process: Ege Azca and Ali Doğan Çamur
- Configuration Plan: Savaş Solak and Yankı Yağız Ozan

Deployment Overview:

- We built the app using Android Studio and tested it on both physical devices and multiple emulators.
- Debugging was done using Logcat and Firebase Crashlytics for error tracking.
- The app integrates with Google Sign-In for authentication.
- Firebase was used for User Data.
- For version control and Collaboration, we have used GitHub.
- Our target platform is Android.

Deployment Process:

Setting Up the Development Environment

- Before deploying, the necessary tools and dependencies must be installed:
- Android Studio (latest stable version) is used for development and debugging.
- The required Android SDK versions are installed.
- An Android Virtual Device (AVD) is set up for testing on an emulator.
- A physical Android device is prepared for real-world testing.

Configuring API Keys & Third-Party Services

- Google Services (Google Sign-In & Maps API)
- A project is created in Google Cloud Console.
- The Google Maps SDK for Android and Google Sign-In API are enabled.
- An API Key for Google Maps is generated and added to the application settings.
- OAuth 2.0 credentials for Google Sign-In are configured.
- The google-services.json file is downloaded from Firebase and placed in the app directory.
- Spotify API Configuration
- The app is registered on the Spotify Developer Dashboard.
- The Client ID & Client Secret are retrieved for authentication.
- The Spotify Web API is set up to extract the currently playing song from the user's account.

Building the APK for Deployment

- Building Debug Version for Testing

- The app is tested within Android Studio to check for errors.
- A physical Android device is used for real-world testing, ensuring compatibility.
- The debug APK is generated using Gradle and manually installed on test devices.
- Building Release Version for Deployment
- A signed APK is created for production use.
- A Keystore file is generated or reused for signing the app securely.
- The release version is built and optimized for performance.
- The final APK is tested before deployment.

Deploying to a Test Environment

- Testing on Emulator & Real Device
- The app is tested on different Android versions to ensure compatibility.
- Both the emulator and physical devices are used to verify UI/UX, location accuracy, and API interactions.
- Debugging & Fixing Errors
- The Logcat tool is monitored for error messages.
- Firebase Crashlytics is enabled to track and resolve crashes.
- API requests for Google Sign-In, Maps, and Spotify are tested to confirm proper integration.

Deploying for Demo Presentation

- Prepare a Test Account for demonstration, ensuring login credentials are available.
- Test the Core Features, including:
- Signing in with Google.
- Displaying the user's location on the map.
- Showing the currently playing song from Spotify.

Configuration Plan:

- **Firebase Setup:**
  Download and add google-services.json from Firebase Console.
  Set up Firebase Authentication to handle user logins.
  Configure Firebase Cloud Firestore or Realtime Database if used for user data.
- **Google Maps API Key:**
  Add google_maps_api_key in strings.xml.
  Ensure the API key is enabled for use in the Google Cloud Console.
- **Spotify Authentication:**
  Register the app with Spotify Developer Console and note down the Client ID and Redirect URI.
  Add the Client ID to your app as a constant.
  Configure the OAuth flow to get the access token for the Spotify API.
- **Permissions:**
  Ensure permissions for location access (in AndroidManifest.xml).

Request permissions at runtime for location services.

- **Build Configuration:**
  Set up Build Types in build.gradle for development and production environments (e.g., debug vs. release builds).
  Configure Firebase and other services for both the debug and release environments.