OPSC7312

README

By: Ebrahim Mohammed

ALPHA EVENT PLANNER

Table of Contents

Alpha Event Planner - README

Overview

Alpha Event Planner is a mobile application designed to help users plan and manage their events efficiently. It allows users to create events, track their details, and categorize them for better organization. The app provides insights into upcoming events and ensures a user-friendly experience with features like search, edit, and delete options. All event data is stored locally and synced with JSONBIN, a cloud-based JSON storage service, for real-time synchronization and multi-device access.

Features

1. Event Management:
   - Users can create, edit, and delete events.
   - Events are saved locally and synced with JSONBIN for real-time access across devices.
2. Search Functionality:
   - Users can search for events using keywords to quickly find specific entries.
3. JSONBIN Integration:
   - Data is stored both locally (SharedPreferences,RoomDB) and synced with JSONBIN for real-time updates and synchronization.
4. Action Menu:
   - The app includes an About page, FAQ page, privacy policy and a share feature.
5. User-Friendly Interface:
   - The app offers a user-friendly and clean interface that enhances the overall experience.
6. Offline mode and Synchronization:
   - Users can input event details while being offline thanks to RoomDB.
7. Map functionality:
   - Users can find the location of their event on the map view.
8. Calendar Integration:
   - Users can view events on a calendar. When a date is selected, all events scheduled for that date are displayed in a bottom sheet. If multiple events exist for the same date, they are shown as a list.

Functional Requirements

1. User Authentication:
    - Secure sign-up and log-in features for users to access their events across devices. (Google, firebase authentication and biometrics)
2. Event Tracking:
    - Users can create, edit, and delete events, tracking the date, time, and location of each.
3. JSONBIN Integration:
    - All event data is stored and synced with JSONBIN, providing real-time access and multi-device synchronization.
4. Calendar Integration:
    - Users can view events on a calendar and quickly access all events on any selected date.

Non-Functional Requirements

1. Usability:
    - The app provides an intuitive interface, ensuring ease of use for all users.
2. Performance:
    - The app is fast and responsive, even when handling a large number of events.
3. Reliability:
    - The app delivers stable performance, with minimal crashes. Data is securely stored in JSONBIN.
4. Security:
    - Data is encrypted during storage and transmission. User authentication is required for sensitive actions.
5. Compatibility:
    - The app is compatible with a wide range of Android devices.

User Prerequisites

1. Android Device:
    - Users should have an Android device running on a compatible version of the Android operating system (Android 5.0 or higher).

2. Internet Connection:
    - A stable internet connection is required for syncing data with JSONBIN and accessing real-time updates across devices.
3. Basic Smartphone Skills:
    - Users should be comfortable with navigating mobile apps and performing basic tasks such as creating, editing, and deleting events.
4. Familiarity with Event Management:
    - Users should understand the concept of creating and managing events, including setting dates, times, and categories.
5. Data Privacy Awareness:
    - Users should be aware of data privacy best practices and understand the need to protect personal information stored in the app.
6. Willingness to Track Events:
    - Users should be interested in actively tracking their events and updating them as needed to benefit from the app's features.
7. Adherence to App Guidelines:
    - Users should follow app guidelines and policies for smooth and secure usage of the platform.

Developer Prerequisites

1. Android Development Experience:
    - Basic to intermediate knowledge of Android Studio and Android development.
2. Firebase Knowledge (Optional):
    - Understanding of Firebase if planning to integrate additional features like authentication or real-time database, though the current setup uses JSONBIN.
3. JSONBIN API Familiarity:
    - Knowledge of how to work with JSONBIN for storing and retrieving data via its API.
4. Networking Skills (Retrofit):
    - Familiarity with Retrofit for handling HTTP requests and interactions with external APIs (JSONBIN).
5. Understanding of SharedPreferences:
    - Knowledge of SharedPreferences for local data storage.
6. Basic Knowledge of Git:

    - Familiarity with version control using Git for collaboration and version tracking.

Frequently Asked Questions (FAQ)

1. How do I create an event?

To create an event, follow the labels on the event creation screen.

2. How can I edit or delete an event?

To edit or delete an event, tap on the event in the list. You will see options to edit or delete it.

3. How do I sync my events across devices?

Your events are synced using JSONBin, a cloud-based JSON storage service. Make sure you are connected to the internet for real-time synchronization.

4. What happens if the network goes down?

If the network goes down, you can still create events. They will be saved locally and synced once the network is restored.

5. How does the search functionality work?

Use the search bar at the top of the screen to filter events by title, date, or location. The list of events will update automatically as you type.

6. How do I view events on the map?

To view events on the map, navigate to the map section of the app. You will see all event locations marked, and you can tap each marker for more details.

Installation

Steps to Install:

1. Download the APK file from the releases page.
2. Enable installation from unknown sources in your device settings.
3. Open the APK file and follow the on-screen instructions to install the app.

Minimum Device Requirements:

- Android Version: Android 5.0 (Lollipop) or higher
- RAM: 2GB or more
- Storage: 50MB free space

Developed With

- Android Studio
- JSONBIN for real-time event data storage and synchronization
- Retrofit for networking and API interactions with JSONBIN

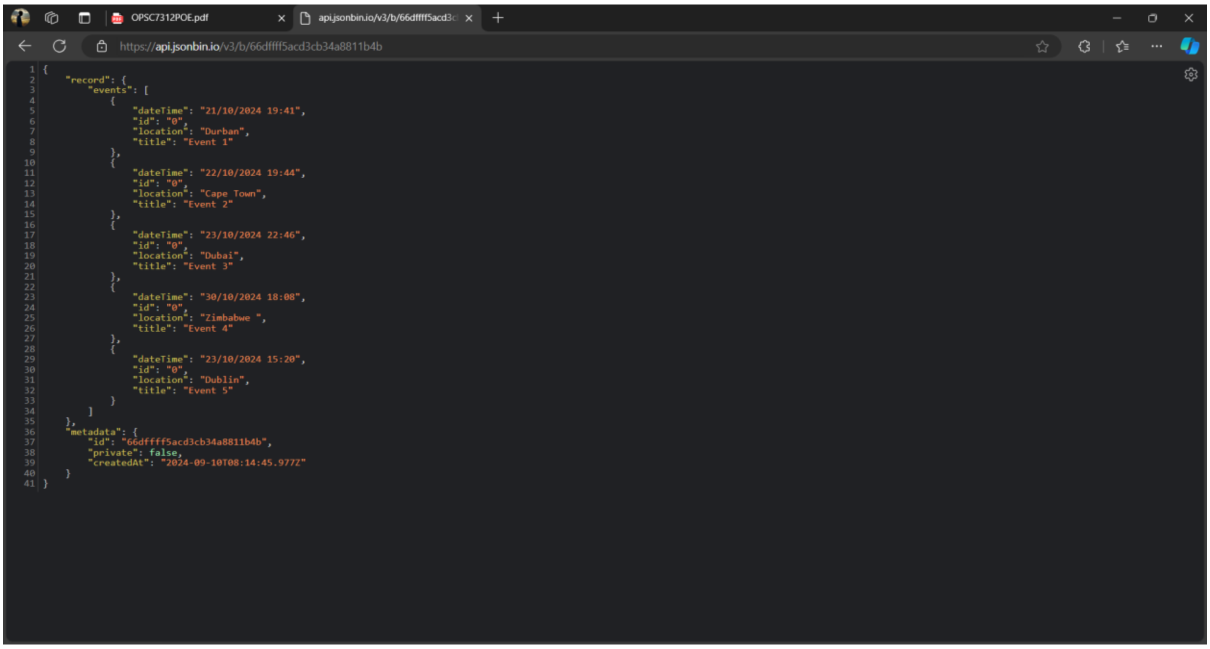
Feedback and Support

If you have any feedback, suggestions, or encounter issues, please reach out to us at:

- Email: ST10084812.vcconnect.edu.za

We are continuously working to improve the app and value your feedback.

Screenshots

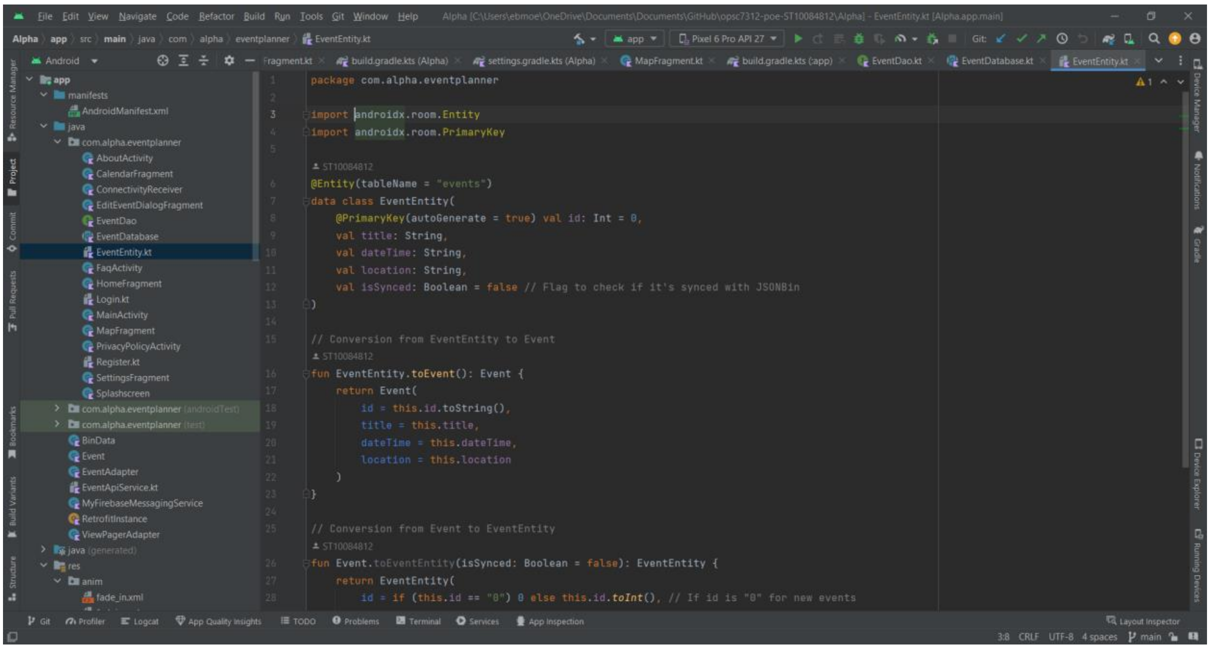

OPSC7312POE.pdf
api.jsonbin.io/v3/b/66dffff5acd3c
https://api.jsonbin.io/v3/b/66dffff5acd3cb34a8811b4b

```
{
    "record": {
        "events": [
            {
                "dateTime": "21/10/2024 19:41",
                "id": "0",
                "location": "Durban",
                "title": "Event 1"
            },
            {
                "dateTime": "22/10/2024 19:44",
                "id": "0",
                "location": "Cape Town",
                "title": "Event 2"
            },
            {
                "dateTime": "23/10/2024 22:46",
                "id": "0",
                "location": "Dubai",
                "title": "Event 3"
            },
            {
                "dateTime": "30/10/2024 18:08",
                "id": "0",
                "location": "Zimbabwe ",
                "title": "Event 4"
            },
            {
                "dateTime": "23/10/2024 15:20",
                "id": "0",
                "location": "Dublin",
                "title": "Event 5"
            }
        ]
    },
    "metadata": {
        "id": "66dffff5acd3cb34a8811b4b",
        "private": false,
        "createdAt": "2024-09-10T08:14:45.977Z"
    }
}
```

JSONBIN: User data is stored securely with JSONBIN. (For testing purposes use:

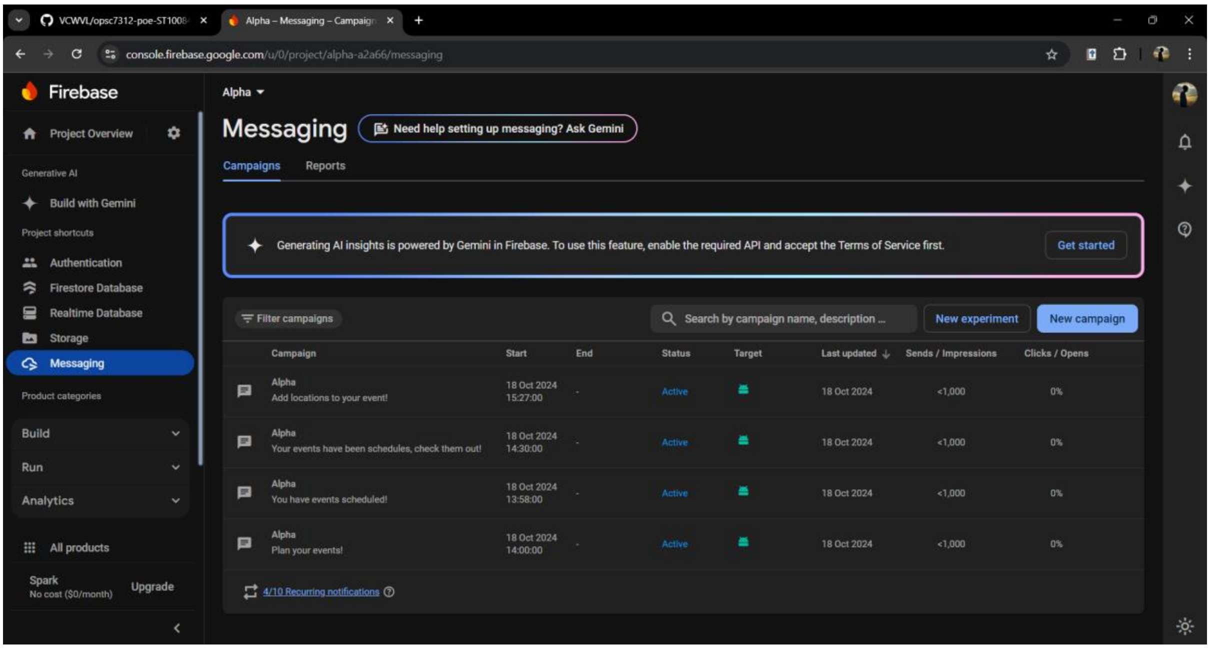
api.jsonbin.io/v3/b/66dffff5acd3cb34a8811b4b

to view data being stored in real time whilst making use of the application.)



```kotlin
package com.alpha.eventplanner

import androidx.room.Entity
import androidx.room.PrimaryKey

@Entity(tableName = "events")
data class EventEntity(
    @PrimaryKey(autoGenerate = true) val id: Int = 0,
    val title: String,
    val dateTime: String,
    val location: String,
    val isSynced: Boolean = false // Flag to check if it's synced with JSONBin
)

// Conversion from EventEntity to Event
fun EventEntity.toEvent(): Event {
    return Event(
        id = this.id.toString(),
        title = this.title,
        dateTime = this.dateTime,
        location = this.location
    )
}

// Conversion from Event to EventEntity
fun Event.toEventEntity(isSynced: Boolean = false): EventEntity {
    return EventEntity(
        id = if (this.id == "0") 0 else this.id.toInt(), // If id is "0" for new events
```

Room DB: For offline mode (Users are able to create events offline, once a connection is found events get synced across the application.)



Alpha

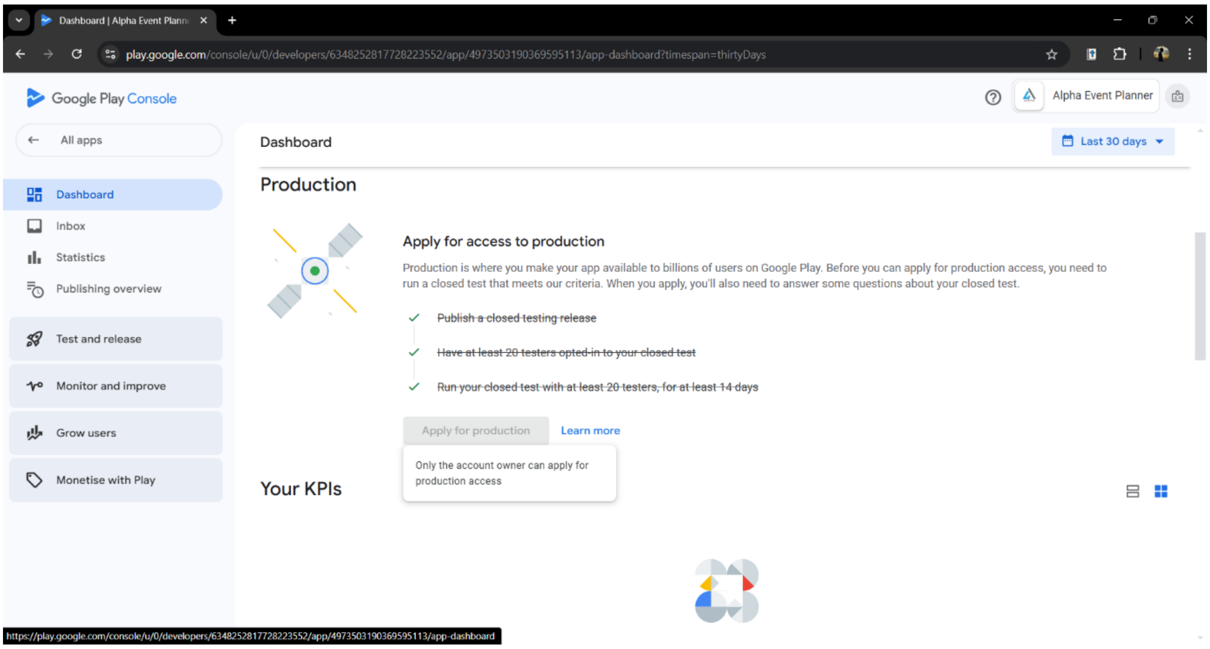
Messaging

Generating AI insights is powered by Gemini in Firebase. To use this feature, enable the required API and accept the Terms of Service first.

Campaign	Start	End	Status	Target	Last updated	Sends / Impressions	Clicks / Opens
Alpha Add locations to your event!	18 Oct 2024 15:27:00	-	Active		18 Oct 2024	<1,000	0%
Alpha Your events have been schedules, check them out!	18 Oct 2024 14:30:00	-	Active		18 Oct 2024	<1,000	0%
Alpha You have events scheduled!	18 Oct 2024 13:58:00	-	Active		18 Oct 2024	<1,000	0%
Alpha Plan your events!	18 Oct 2024 14:00:00	-	Active		18 Oct 2024	<1,000	0%

4/10 Recurring notifications

FCM: Created generic push notifications for Alpha Event Planner. (As well as in app messages locally using shared preferences this is included in the demo video).



Dashboard

Production

Apply for access to production

Production is where you make your app available to billions of users on Google Play. Before you can apply for production access, you need to run a closed test that meets our criteria. When you apply, you'll also need to answer some questions about your closed test.

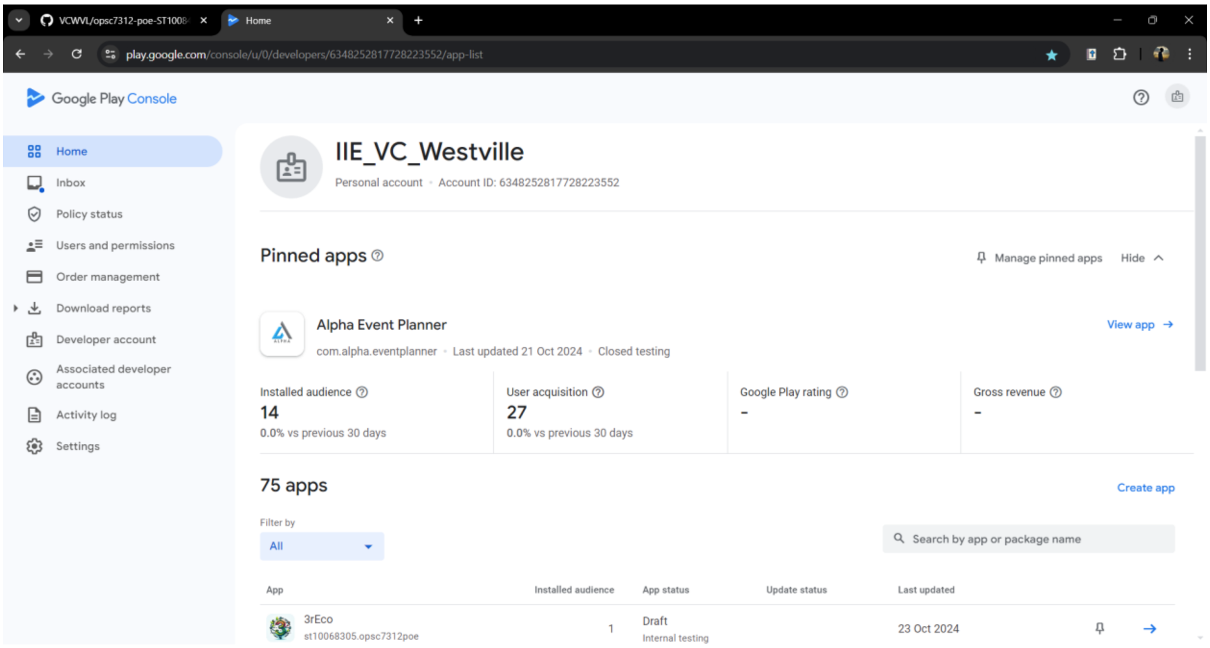
- ✓ ~~Publish a closed testing release~~
- ✓ ~~Have at least 20 testers opted-in to your closed test~~
- ✓ ~~Run your closed test with at least 20 testers, for at least 14 days~~

Only the account owner can apply for production access

Your KPIs

https://play.google.com/console/u/0/developers/6348252817728223552/app/4973503190369595113/app-dashboard

Google Play Console(Proof 1)



IIE_VC_Westville

Personal account · Account ID: 6348252817728223552

Pinned apps

Alpha Event Planner

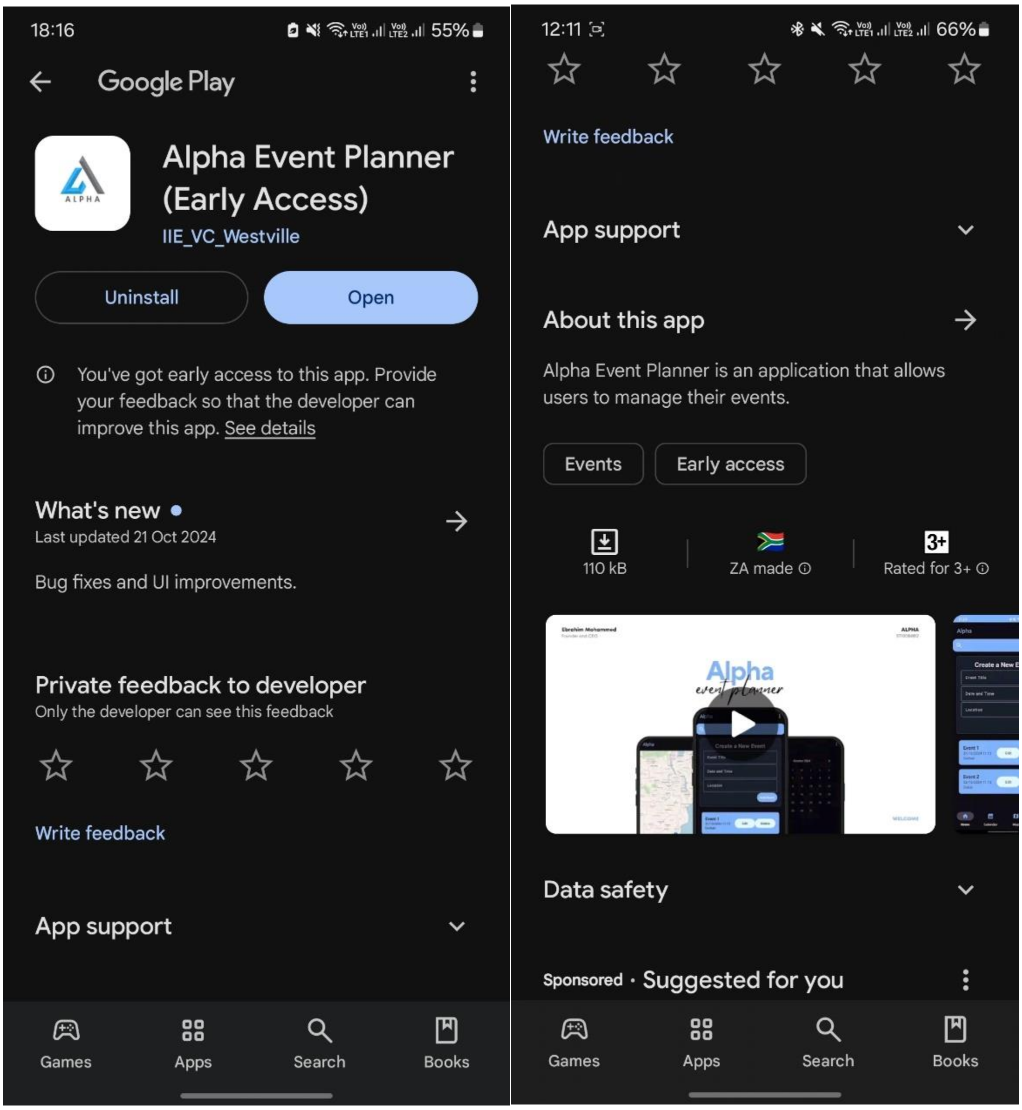
com.alpha.eventplanner · Last updated 21 Oct 2024 · Closed testing

Installed audience	User acquisition	Google Play rating	Gross revenue
14 0.0% vs previous 30 days	27 0.0% vs previous 30 days	-	-

75 apps

App	Installed audience	App status	Update status	Last updated
3rEco st10068305.opsc7312poe	1	Draft Internal testing		23 Oct 2024

Google Play Console(Proof 2)



Alpha Event Planner (Early Access)

IIE_VC_Westville

You've got early access to this app. Provide your feedback so that the developer can improve this app. See details

What's new

Last updated 21 Oct 2024

Bug fixes and UI improvements.

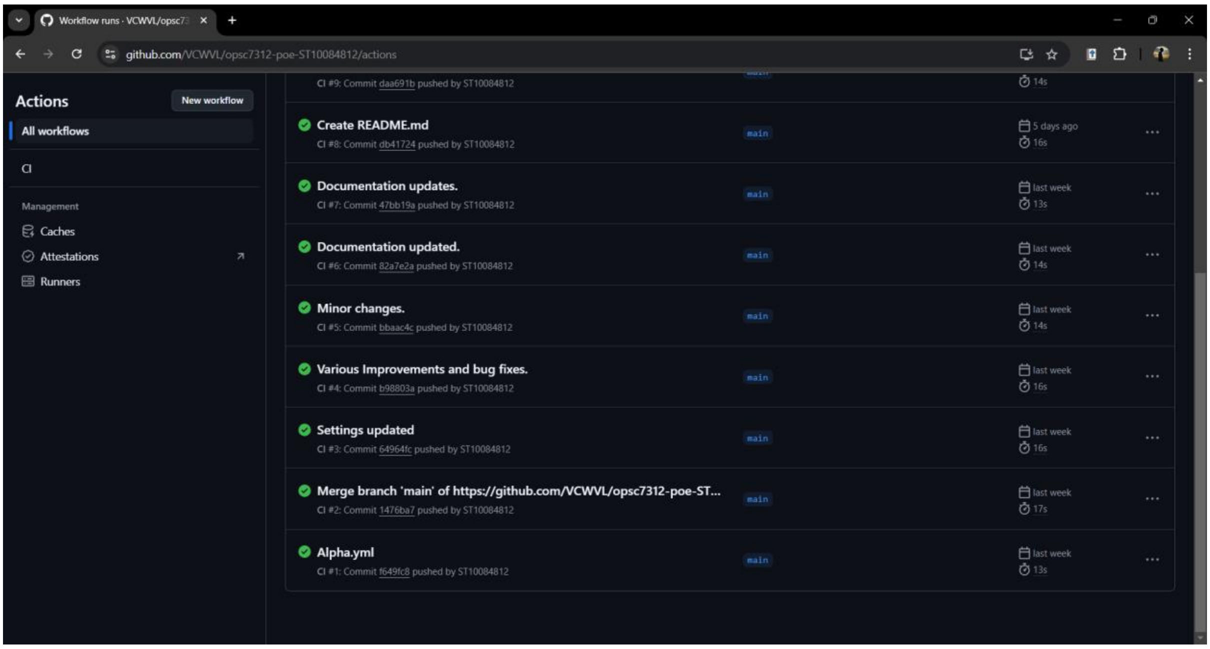
Private feedback to developer

Only the developer can see this feedback

About this app

Alpha Event Planner is an application that allows users to manage their events.

Google Play Store(Proof 3)


Workflow runs · VCWVL/opsc73
github.com/VCWVL/opsc7312-poe-ST10084812/actions
Actions
New workflow
All workflows
CI
Management
Caches
Attestations
Runners

CI #9: Commit daa691b pushed by ST10084812 — 14s

Create README.md
CI #8: Commit db41724 pushed by ST10084812 — main — 5 days ago — 16s

Documentation updates.
CI #7: Commit 47bb19a pushed by ST10084812 — main — last week — 13s

Documentation updated.
CI #6: Commit 82a7e2a pushed by ST10084812 — main — last week — 14s

Minor changes.
CI #5: Commit bbaac4c pushed by ST10084812 — main — last week — 14s

Various Improvements and bug fixes.
CI #4: Commit b98803a pushed by ST10084812 — main — last week — 16s

Settings updated
CI #3: Commit 64964fc pushed by ST10084812 — main — last week — 16s

Merge branch 'main' of https://github.com/VCWVL/opsc7312-poe-ST...
CI #2: Commit 1476ba7 pushed by ST10084812 — main — last week — 17s

Alpha.yml
CI #1: Commit f649fc8 pushed by ST10084812 — main — last week — 13s

Code Attribution

The Alpha Event Planner app makes use of the following external libraries and tools:

- **AndroidX Core KTX** (androidx.core:core-ktx:1.13.1): A set of Kotlin extensions that helps make Android development more concise and idiomatic.
- **AndroidX AppCompat** (androidx.appcompat:appcompat:1.7.0): Provides backward-compatible versions of Android features.
- **Material Components** (com.google.android.material:material:1.12.0): Implements Material Design UI components for Android apps.
- **ConstraintLayout** (androidx.constraintlayout:constraintlayout:2.1.4): A layout manager for creating flat, flexible UI layouts.
- **ViewPager2** (androidx.viewpager2:viewpager2:1.1.0): A widget for sliding between fragments with a smooth swipe gesture.

Firebase Integration

- **Firebase BOM** (com.google.firebase:firebase-bom:33.2.0): Ensures consistent Firebase versioning for multiple Firebase services.
- **Firebase Authentication** (com.google.firebase:firebase-auth:23.0.0): Provides user authentication support, including email/password and third-party authentication like Google Sign-In.
- **Firebase Realtime Database** (com.google.firebase:firebase-database): Enables storing and syncing data in real-time across all clients.
- **Firebase Firestore** (com.google.firebase:firebase-firestore): A flexible, scalable database for app development.
- **Firebase Storage** (com.google.firebase:firebase-storage): Provides cloud-based storage for user-generated content, like photos and videos.
- **Firebase Messaging ("com.google.firebase:firebase-messaging"):** Allows for notifications.

Google Sign-In

- **Google Play Services Auth** (com.google.android.gms:play-services-auth:21.2.0): Supports OAuth 2.0 authentication, allowing users to log in with their Google account.

Image Loading

- **Glide** (com.github.bumptech.glide:glide:4.15.1): Efficient image loading and caching for Android applications.
  - **Glide Compiler** (com.github.bumptech.glide:compiler:4.15.1): Annotation processor to generate APIs from Glide.

Testing

- **JUnit** (junit:junit:4.13.2): A widely used testing framework for Java applications.
- **AndroidX JUnit** (androidx.test.ext:junit:1.2.1): Extends JUnit with Android support for testing Android applications.
- **Espresso** (androidx.test.espresso:espresso-core:3.6.1): A testing framework for writing concise and reliable Android UI tests.

Networking and API Interactions

- **Retrofit** (com.squareup.retrofit2:retrofit:2.9.0): A type-safe HTTP client for making API requests.
- **Retrofit GSON Converter** (com.squareup.retrofit2:converter-gson:2.9.0): Converts JSON responses into Java objects using GSON.
- **GSON** (com.google.code.gson:gson:2.10.1): A library for serializing and deserializing Java objects into JSON.

Biometrics

- **AndroidX Biometric** (androidx.biometric:biometric:1.4.0-alpha02): Provides biometric authentication support for fingerprint, face, and other types of biometric authentication.

Maps and Navigation

- **OSMDroid** (org.osmdroid:osmdroid-android:6.1.16): An open-source mapping library that allows the display of offline maps and provides various map utilities for Android applications.

Demo Video

A demo video demonstrating the core functionality of Alpha Event Planner is included in the linked video.

YouTube Link: https://youtu.be/v52sElzbMDQ?si=nwj9P6R0ArGOkLpH

GitHub

Link: https://github.com/VCWVL/opsc7312-poe-ST10084812.git

GitHub Workflows (screenshot above.)

References

- Global Logic Non-functional Requirements for Mobile Apps
- W3Schools
- YouTube Tutorial 1

- YouTube Tutorial 2
- **AndroidX Documentation:** https://developer.android.com/jetpack/androidx
- **Firebase Documentation:** https://firebase.google.com/docs
- **Retrofit Documentation:** https://square.github.io/retrofit/
- **Glide Documentation:** https://github.com/bumptech/glide
- **JUnit Documentation:** https://junit.org/junit4/
- **Espresso Documentation:** https://developer.android.com/training/testing/espresso
- **OSMDroid Documentation:** https://osmdroid.github.io/
- https://developer.android.com/codelabs/biometric-login#0
- https://developer.android.com/identity/sign-in/biometric-auth
- https://firebase.google.com/docs/cloud-messaging
- https://github.com/marketplace/actions/automated-build-android-app-with-github-action
- https://github.com/IMAD5112/Github-actions/blob/main/.github/workflows/build.yml
- Save data in a local database using Room | Android Developers
- SharedPreferences | Android Developers
- RecyclerView.Adapter | Android Developers
- Fragments | Android Developers
- Step-by-Step: Setting Up and Implementing Room Database in Android | by Shamsuddoha Ranju | Medium
- https://youtu.be/-LNg-K7SncM?si=9p3K4D240saA8yJn
- Android Studio-Google maps doesn't show map - Stack Overflow