Overview:

Sky Sailor is an inventive flight booking and an itinerary management application designed to simplify the complexities of travel planning. It integrates advanced technologies and user-centric design. Sky Sailor provides a seamless experience for booking flights, managing travel schedules, and staying informed about travel updates. This ultimately ensures travel planning hassle-free for users.

Functionalities:

1. **Four Distinct Screens (More Than Two Distinct Screens):** Sky Sailor features a versatile and intuitive user interface across multiple distinct screens:
    - **Login/Sign Up Screen:** Secure entry point for user registration and login, using Firebase Authentication for robust security.
    - **Home Screen:** Central dashboard that provides quick access to all major functions such as upcoming flights bookings, offers, and ads.
    - **Search Screen:** Allows users to efficiently search for flights using multiple parameters and different flight types, displaying detailed results with options to book directly.
    - **My Trips Screen:** Dedicated area for users to view, manage, and archive their upcoming and past itineraries, enhancing travel organisation and accessibility.
    - **Settings Screen:** Customisable user settings for managing personal preferences, notification settings, display of archived flights and logging out of the user's account.

2. **App Navigation for Different Screens (Navigation Component):**
    - Utilise NavHost and NavController for seamless navigation across Login/ Register, Home, Search, My Trips, and Settings screens.

3. **User Authentication (Firebase Authentication):**
    - Securely register and log in users, ensuring safe and personalised user experiences.

4. **Flight Search and Booking (Room Local Storage, Firebase):**
    - Allow users to search for flights, view flight details such as dates and prices, and book flights. Data is saved locally for offline access and synced with Firebase for cloud storage. Confirmation dialogs ensure user consent before booking or cancelling flights.
    - Firebase structure,
        i. flights (flightID, departureDate, returnDate, destination, origin, passengerCount, price, returnDate, tripType)
        ii. users (UserID) → bookedFlights (archived, deleted, flightID, departureDate, returnDate, destination, origin, passengerCount, price, returnDate, tripType)

5. **Manage Booked Flights (Room Local Storage, Firebase, Swipe Gestures):**
    - Users can view, archive, or delete their flight bookings with intuitive swipe gestures, with changes synchronised across devices via Firebase. Confirmation dialogs prevent accidental deletions or changes to bookings.

6. **View and Manage Offers (Firebase, Preference DataStore):**
    - Retrieve and display flight-related offers before their expiry date, with the capability for users to favorite offers stored locally, keyed to their user ID for personalised access.
    - Firebase structure,
        i. offers (description, discount, expiryDate)

7. **Schedule Notifications (Permissions, Notifications, Broadcast Receiver, Local Storage, Implicit Intent):**
    - Send timely notifications for upcoming flights, manage notification preferences and permissions, and utilise broadcast receivers to trigger notifications aligned with the user's travel schedule. Uses local storage to store the current preference of the user's for enabling notifications in the sky sailor app. If notifications are permanently disabled, the app directs users to the system settings for adjustments.

8. **Share Flight Details (Android Sharesheet):**
    - Enable users to share their flight details with others via Android Sharesheet, enhancing the social functionality of the app.

9. **Intent to External Apps (Implicit Intent):**
    - Direct integration with Google Maps to display airport locations for departures, enhancing user orientation and planning.

10. **App Lifecycle Management (Lifecycle-Aware Components, State Restoration):**
    - Ensure robust handling of app lifecycle events including screen rotations, using lifecycle-aware components like ViewModel to maintain and restore UI state. Enhanced state restoration techniques have been integrated into the main activity file to efficiently manage transitions between states such as onPause, onDestroy and more, further securing user data and app stability during activity lifecycle changes.

11. **Custom Content Provider (ContentProvider, Room):**
    - Develop a custom ContentProvider that interfaces with the Room database for efficient data handling and accessibility, ensuring flight and booking data can be accessed by other apps. Extensive testing has been conducted using Android Studio's experimental test features to ensure robust performance and reliability.

Sky Sailor leverages cutting-edge technology to deliver a robust, intuitive, and comprehensive travel management solution. By addressing key user needs through sophisticated functionalities and ensuring seamless integration across system components, Sky Sailor aims to redefine the travel planning experience. The application enhances safety and user confidence using confirmation dialogs for critical actions such as booking, archiving, or deleting flights, as well as enabling notifications. This ensures users make informed decisions without accidental inputs. Additionally, if a user permanently disables notification permissions, the app intelligently guides them to the system settings, promoting a thoughtful and user-friendly interaction with app permissions.

User Experience and Design Philosophy: Sky Sailor is designed with a focus on simplicity and efficiency, ensuring that the user interface is intuitive even for first time users. The design philosophy emphasises minimalistic layouts, coherent colour schemes, and responsive design to work well across various landscape and portrait orientations. This approach not only enhances user satisfaction but also boosts app usability and accessibility, making it an imperative tool for travellers.

If you want to see populated booked flights, login with the below credentials:
Email: Flightapp123@gmail.com
Password: flightapp

Mock Screens