Trip Hunter - Final Report

Eleni Loizou UC: 1058322

Stylianos Panayiotou UC: 1057970

Chrysanthi Theocharidi UC: 1057961

---

1. Introduction

Trip Hunter is an Android-based travel guide application designed to transform how travelers explore and plan their journeys. By offering curated recommendations, real-time updates, and offline functionality, Trip Hunter simplifies travel and creates a personalized, engaging experience for users. The app leverages technologies like **Firebase** for secure data management, **Yelp API** for up-to-date bars/clubs recommendations, **WeatherStack API** for weather updates and **MAPS API** for the location of restaurants and night life venues .

Motivated by the challenges of planning trips and exploring unfamiliar destinations, Trip Hunter provides solutions through streamlined usability, rich content, and advanced features like weather updates, reviews, and caching for offline access.

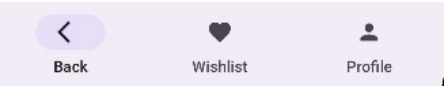
---

2. Design Phase

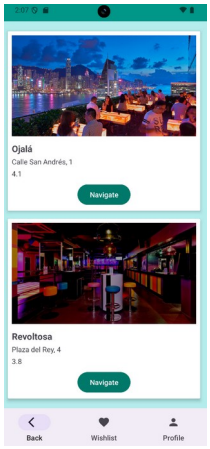
The design of Trip Hunter was guided by three key principles: **usability, engagement, and scalability**. Wireframes and prototypes were created using **Figma** to conceptualize the user interface and user experience before implementation.

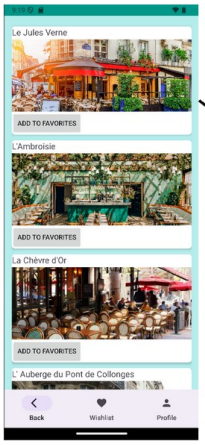
Design Choices

1. **Minimalist Navigation:**
    - A bottom navigation bar was chosen for its simplicity and ease of use.
    - Buttons for Back, Wishlist, and Profile allow seamless movement between app sections.

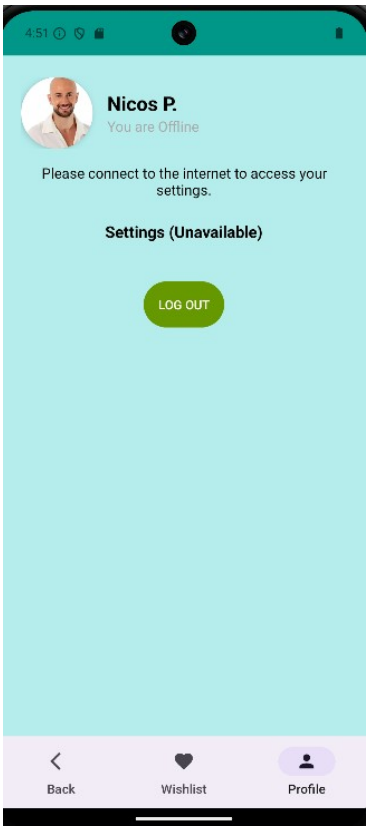


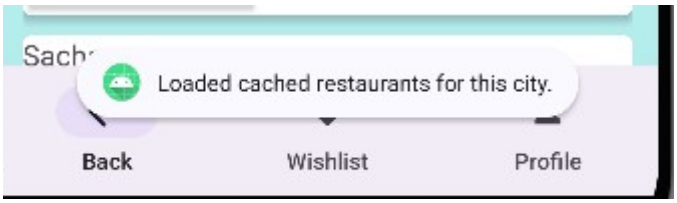
2. **Dynamic Pages:**
    - The app integrates APIs (e.g., Yelp and WeatherStack, Map) to provide real-time, relevant content.



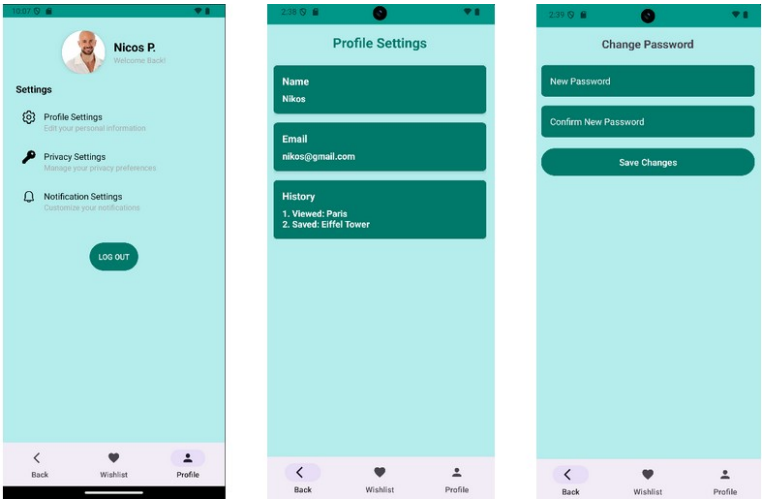


3. **Offline Functionality:**
    - Key information (e.g., restaurants cached data) is saved locally for users with limited connectivity.





4. **Personalization:**
    - Features like the profile page and review submissions foster user engagement and make the app feel tailored to individual preferences.



The **Figma prototype** served as the blueprint for translating these ideas into reality, enabling early visualization of the app's functionality.

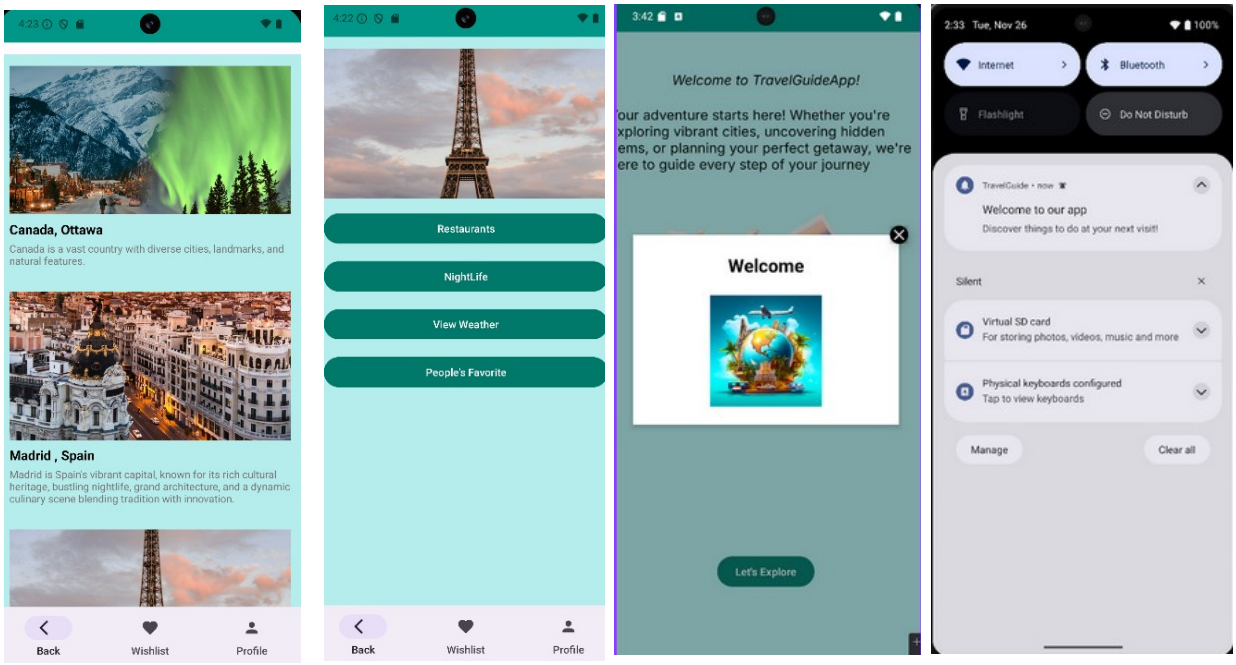
---

3. Project Overview

3.1 Features

Trip Hunter integrates the following key features:

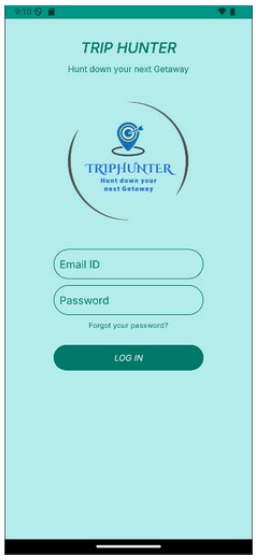
- **Explore Destinations:**
    - Users can browse curated lists of destinations, restaurants, and nightlife spots with detailed descriptions and real-time data.
- **Weather Updates:**
    - Displays live weather updates using the **WeatherStack API**, including dynamically changing icons based on current conditions.
- **Restaurant and Nightlife Listings:**
    - Recommendations fetched from the **Yelp API** include ratings, addresses, and navigation options via Google Maps.
- **User Reviews and Ratings:**
    - Users can share feedback and rate destinations, with all data securely stored in **Firebase**.
- **Favorites Management:**
    - Users can save destinations and venues for quick access, even offline.
- **Push Notifications:**
    - Timely reminders and updates are sent via **Firebase Cloud Messaging**.
- **Offline Mode:**
    - Cached data ensures that users can access to the restaurants that previously viewed without internet access.

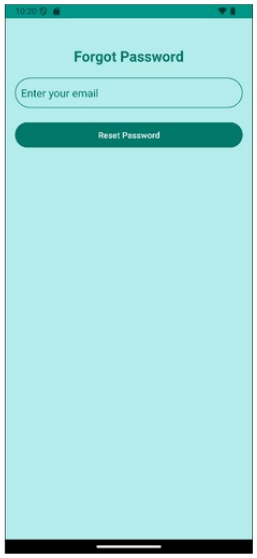


3.2 User Interface

The app consists of multiple intuitive screens:

1. **Splash Screen:** Displays the app's logo and tagline during startup.
2. **Login/Signup Page:** Handles authentication via **Firebase Authentication**.
3. **Home Page:** The central hub for exploring destinations and accessing features.
4. **Destination Details:** Provides comprehensive information about venues, including options for navigation and saving as favorites.
5. **Review Submission:** Allows users to leave reviews with star ratings.
6. **Settings and Profile Pages:** Includes password management, account details, and browsing history.





3.3 Instructions for Use

1. **Login or Signup** to access the app.
2. Navigate through the home page to explore destinations, restaurants, and nightlife spots.
3. Use the **Favorites** feature to save preferred locations for future use.
4. Check real-time **weather updates** for selected destinations.
5. Contribute reviews and ratings to help other travellers.
6. Access **offline mode** to view cached restaurants that previously fetched information.

---

4. Project Planning

The team worked collaboratively to deliver Trip Hunter. Each team member contributed significantly, focusing on specific aspects of the application:

- **Stylianos Panayiotou:**
    - Developed the **Yelp API** integration for nightlife activities.
    - Implemented **Push Notifications** using Firebase Cloud Messaging.
    - Designed and coded the **Splash Activity**.
    - Worked on the **Offline Activity**, ensuring cached data functionality.
    - Also did the **Reviews page.**
- **Eleni Loizou:**
    - Designed and implemented all **XML layouts and styles** for the app's user interface.
    - Developed the **Wishlist Page** and **Data Helper Database** functionalities.
    - Coded the **Signup, Login, and Start Pages**, integrating them with **Firebase Authentication.**
    - Led the development of the **PowerPoint presentation**.
- **Chrysanthi Theocharidi:**

- Implemented the **Restaurants Page**, integrating Yelp API for real-time recommendations.
- Developed the **Weather API integration** and **Weather Activity** for real-time updates.
- Designed the **Maps API Activity** for navigating to destinations.
- Coded the **Destinations Detail Activity** to display curated information for each location.
- Made an authorization in Firebase so when the **Forgotten Password** page appears to reset the code by sending an email

The team utilized **GitHub** for version control, with all contributions regularly pushed to a private repository. Collaborative tools like Figma were used to align on design and development goals.

---

5. Future Improvements

5.1 Limitations

- **Limited Offline Data:** Currently, only restaurants are cached for offline use.
- **Basic Push Notifications:** Push notifications are limited to static reminders and could be made more dynamic.

5.2 Challenges

- **Offline Caching:** Ensuring seamless retrieval of cached data was complex and required efficient local storage solutions.
- **API Integration:** Managing asynchronous data fetches from APIs while maintaining app responsiveness was a significant challenge.

5.3 Planned Features

- **Flight and Hotel Bookings:**
  - Integration with travel booking APIs to allow users to book accommodations and flights directly.
- **AI Recommendations:**
  - Machine learning algorithms to suggest destinations based on user preferences and history.
- **Enhanced Offline Mode:**
  - Expanding offline capabilities to include images and additional data for smoother functionality.

---

6. Conclusions

Trip Hunter successfully delivers a robust, user-focused travel guide application that addresses key traveler needs, including real-time information, user reviews, and offline functionality. Through modern technologies like Firebase, Yelp API, and WeatherStack API, the app provides a comprehensive and engaging experience.

This project provided valuable insights into API integration, user-centered design, and efficient caching mechanisms. Future enhancements will further strengthen Trip Hunter's position as an indispensable travel companion.

---

7. References

Documentation

- Firebase Documentation
- Android Developer Guide

APIs

- Google Maps API → https://console.cloud.google.com/apis/credentials/key/8193a5ff-6546-4172-a525-6cdb2d672476?inv=1&invt=AbihXw&project=travelguideapp-442416
- Yelp API → https://api.yelp.com/
- WeatherStack API → http://api.weatherstack.com/

Design and Collaboration Tools

- Figma Prototyping Tool
- GitHub → https://github.com/lne2003/TravelGuideApp.git