## 1. Introduction

The course project focuses on designing and developing a fresh mobile application. This project will not only demonstrate skills learned in class in software construction development, API integration, and cloud deployment, but also introduce a unique concept of a Restaurant Finder Mobile Application.  
  
It is not like a common app existing in a platform such as Google Maps or Yelp, this app will be lightweight, cross-platform, and cloud-enabled, with a focus on personalized features like saving favorites across devices. By being open-source, it also demonstrates transparency and learning, making it a unique academic and practical project.

## 2. Problem Definition

When visiting unfamiliar areas, users often struggle to find quality restaurants that match their preferences quickly. Existing apps offer a vast array of options, but they can overwhelm users with cluttered interfaces, ads, and features that fail to prioritize simplicity and personalization.  
  
This project addresses that gap by providing:

* Quick geolocation-based suggestions tailored to the user’s location.
* A minimalist UI for browsing restaurants via list and map views.
* Favorites management, synced securely across devices using cloud storage.

## 3. Research Insights

Industry leaders like Google Maps and Yelp dominate restaurant discovery. However:

* Google Maps is feature-rich but often overwhelming for simple, quick searches.
* Yelp focuses heavily on reviews and promotions, which may introduce bias or distract from decision-making.
* Zomato (popular in certain regions) is limited by availability and requires account setup for most features.
* Users increasingly want focused, intuitive apps that give them relevant results quickly, without unnecessary distractions.

## 4. Unique Value Proposition

What makes this project unique compared to industry apps:

* Lightweight & Focused → Prioritizes restaurant discovery with minimal distractions.
* Cross-Platform by Design → Built with React Native & Expo for iOS and Android.
* Cloud-Synced Favorites → Uses Firebase to allow users to save and access favorites across devices.
* Academic + Practical → Designed as learning, they are open-source and customizable.

## 5. Technical Approach & Architecture: For the initial diagram, see: [https://github.com/JPretz/sweng861-2025-FALL-project-proposal.](https://github.com/JPretz/sweng861-2025-FALL-project-proposal)

Architecture Components

* Client (Mobile App): React Native (Expo) for UI, navigation, and geolocation.
* External APIs: Google Maps API & Yelp/Zomato API for restaurant data.
* Cloud Backend: Firebase Authentication & Firestore for user data and favorites.  
    
  Workflow:
  1. User opens app → retrieves current location.
  2. API call fetches nearby restaurants.
  3. Results displayed in list + map view.
  4. User can save favorites → stored in Firebase.

## 6. Tech Stack

* Frontend (Mobile): React Native (Expo)
* Backend / Cloud: Firebase Authentication & Firestore
* APIs: Google Maps API, Google Places / Yelp / Zomato API
* Deployment: Expo Go (development), Play Store (demo APK)

## 7. Project Milestones:

## See: <https://github.com/JPretz/sweng861-2025-FALL-project-proposal>.

## 8. Future Extensions

## Advanced filters (price, cuisine, ratings).

## Push notifications for nearby offers.

## Offline support for cached restaurant data.

## AI-based personalized recommendations.

## Social features: reviews, sharing, and ratings.

## 9. Conclusion

This project will demonstrate mobile development, cloud integration, and API-driven design, while solving a real-world problem: making restaurant discovery faster, simpler, and more personal. The final deliverables will be a working mobile app, hosted source code on GitHub, and documentation showcasing a scalable, user-friendly solution.