App Design

# A - Background Research

## Task 1: Project choice and executive summary of the chosen project

### Concept

Foodradar is an app for rating and reviewing individual dishes at a given restaurant and will help the user decide which dish to order. This specific search is missing within the current app context, with many services providing overall restaurant reviews, but never ranking the individual meals. We aim to fill this niche, giving users more information so that they can more easily decide where and what to eat.

### Required features and functionalities

### Target user group

*Probably needs more detail*

Live in urban are with many food options

Have disposable income

Tech literate & own mobile phone

Age range *(check reference) – refer to demographics using related apps or research study as source*

### User demand and benefits of your app

### Other existing apps

### Explain how your mobile app will fulfil user needs

Foodradar will make it easier for people to decide what to order when at a restaurant. Many people are indecisive when it comes to ordering food. Foodradar tries to make the most of your restaurant visit, enabling the user to choose their optimal dish. As foodradar has all this data on individual dishes it will also implement a search through these dishes. The user will be able to search for reviews on specific meals, E.g. if the user wants a Carbonara dish they can get the top rated Carbonara dishes near them.

## Task 2 – Background research and review of related mobile applications

### Problem Space Presentation and Discussion

* Stakeholders and their challenges and needs
* Reasoning and how the problem space informed your approach

### Mobile Application Review

*For each reviewed app include: the name of the app, its URL (in Appstore) and screenshots of each key UI (user interface) component with a description of how it is used*

*Discuss the positive and negative aspects of each app from the perspectives of: approach, features and functionality, general UI design, page navigation*

*Focus your discussion on what you can learn from the reviewed apps and how it can improve and inform your own app design*

**Oink[[1]](#footnote-1)**

Oink was an app from the developer Milk which had similar core concept to FoodRadar. Unfortunately, it was shut down after just five months[[2]](#footnote-2). The developer abandoned the project to work on other ideas, leaving behind a number of users. This project is an excellent case study, as we will be able to learn from both their successes and failures in terms of UI and other aspects.

Positive:

* It provides a more detailed rating system than other apps.
* Newsfeed on friend’s ratings.

Negative:

* The applications scope was too large, allowing users to rate anything.
* Depends entirely on user input for data.

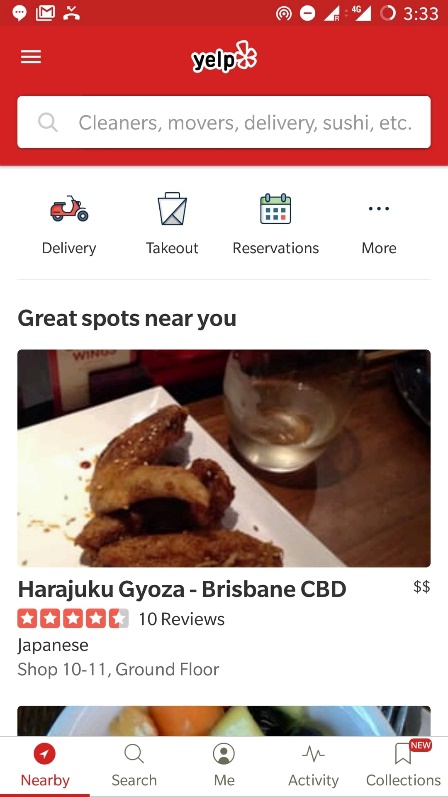
Conclusion:

* Oink had a clear problem with the scope of their application being too large. As all their data comes from user input the result is likely to be too chaotic and inaccurate.

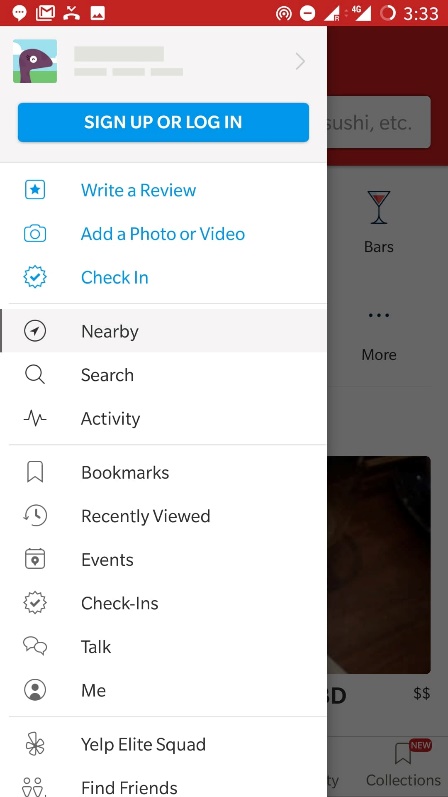
**Yelp**

Android: <https://play.google.com/store/apps/details?id=com.yelp.android&hl=en_AU>  
Apple: <https://itunes.apple.com/au/app/yelp-discover-local-favorites/id284910350?mt=8>

Yelp is one of the most famous websites for rating restaurants and other locations. Yelp presents the general rating, price and some more information so the customer can decide is the location fits their needs.

UserInteface:

**Home page**: The main page has a stack layout with some recommended restaurants/businesses near you. The



Positive:

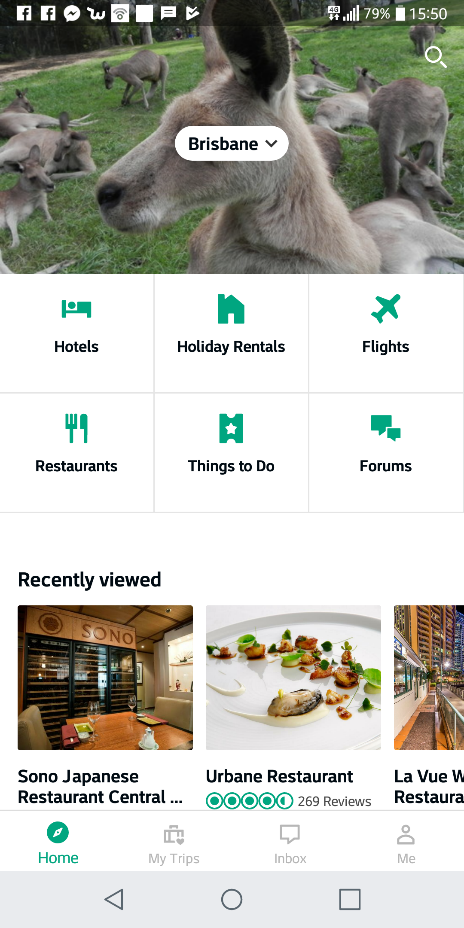
**TripAdvisor**TripAdvisor has a reliable and popular rating system for restaurants. They are well established with a comprehensive database of restaurants. They have a wide range of filters, categories and dietary restrictions.

Figure 1: TripAdvisor Home Page

**User Interface:** The UI is really search driven with most screens revolving around search or recommendations. The navigation is simplistic with a persistent navbar at the bottom of the screen

**Home page**: The homepage prominently displays the six categories of services that TripAdvisor reviews followed by horizontally scrollable lists of restaurants grouped up by categories. Everything presented here is filtered by some given location which defaults to the user’s current location. This is a smart design decision made to minimize the number of clicks required on average as the app is mostly used to find immediate placed to go rather than for longer term planning.

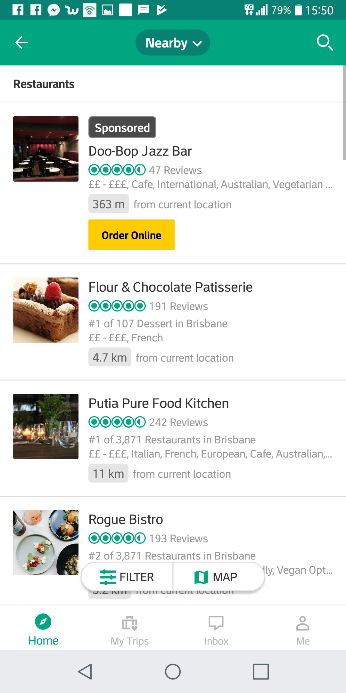
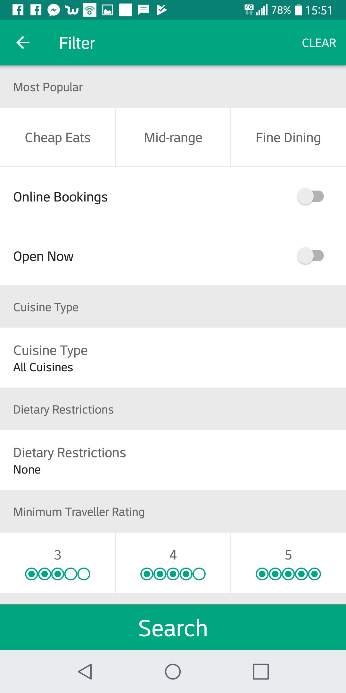
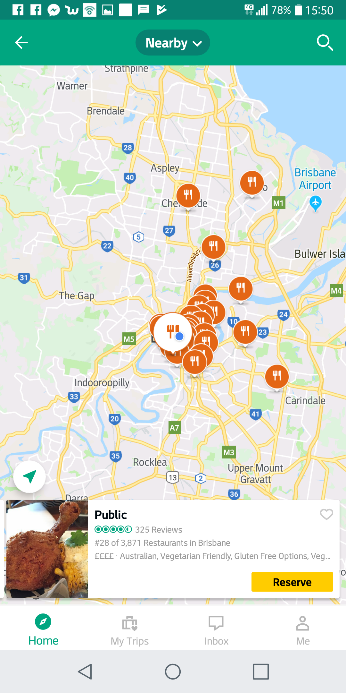
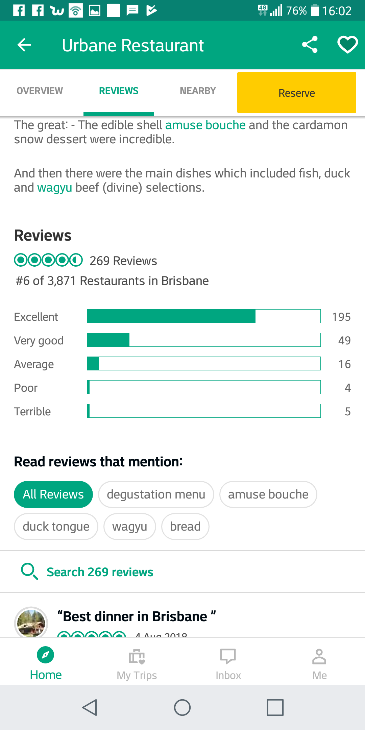
******Search page**: The search is a familiar stack layout with 4 key elements: location, general search, comprehensive filters and a map. Some places have the option to order or reserve a table online that redirect users to the appropriate website. The filter options are extensive and intuitive to use. The map has pins representing each restaurant that fits the search filters. When a pin is selected a small box at the bottom of the screen shows its name and a short summary. The reviews tab gives a total number of reviews as well as a distribution breakdown. A great feature included is finding reviews that mention some keyword allowing for an ad-hoc way to see reviews for individual meals.

Figure 2: Filters

Figure 5: Reviews

Figure 4: Search

Figure 3: Map

In conclusion TripAdvisor allows the user to search through the reviews, which FoodRadar will also implement. What FoodRadar will hold above TripAdvisor is the ability to read reviews and ratings for individual dishes.

**OpenTable**

OpenTable is an Application, which allows the user to reserve a table at restaurants. Additionally, it shows the menu to all the listed places and overall ratings for the restaurant. It does not include ratings for individual dishes on the menu.

**User Interface:**

# B – Mobile App Design

## Task 1: User Interface Design

UI prototype (wireframes)

*Present your prototype of pages and explain the user interface. Elaborate on the UI layout of pages (e.g., user inputs), the applied navigation strategy, applied navigation patterns, applied forms patterns, applied design principles*

### Storyboards

**Scenario 1 – “What to order at this store?”:**

Without FoodRadar:



With FoodRadar:



**Scenario 2 – “Where to find a good burger?”:**

Without FoodRadar:



With FoodRadar:



### Explain the applied UI patterns and UI styling considerations

*Maybe this section should be moved directly under the wireframes section? Not sure*

## Task 2: Software architecture and implementation

### Describe the overall system architecture

*Provide a diagram and description of the various components in the broader system, their role and how they integrate with your application.*

### Describe the overall software architecture

*Describe your data model (e.g., using UML), explain which architectural patterns you are planning to apply and justify why.*

### Flexibility and Maintainability

*Explain how your architecture supports adaptations for changes of requirements (e.g. UI changes), technologies (e.g., migrate to other platforms), application logic (e.g., page navigation) and additional features. Identify and discuss whether the chosen patterns may have drawbacks.*

### Integration

*Explain how your architecture supports the future integration with Cloud solutions, Web Services or back-end systems.*

The application is using the Customers location to display nearby restaurants and filter the result by the user needs. The user gets the ability to filter with the appropriate GUI by distance, price, and items served.

The app is displaying data from our cloud-based database. When the user rate a meal, they are adding data to this database.

To fill the database, we will use the Zomato API[[3]](#footnote-3). This offers 1000 requests per day for free and can deliver lot of data about restaurants all over the world. It also provides the menu for a lot of restaurants. This data will be combined with the data which the community will provide, to add ratings to the list of dishes available at the location.

### Testing

*Explain how your software architecture supports testing.*

# Reference List

1. https://www.youtube.com/watch?v=5xADESocujo [↑](#footnote-ref-1)
2. https://www.theverge.com/2012/3/14/2872172/oink-app-kevin-rose-shut-down [↑](#footnote-ref-2)
3. https://developers.zomato.com/documentation [↑](#footnote-ref-3)