

COMP228 Assignment 2 - Places to Eat and Drink Document

Map View

This screen is simplistic, to allow the user to have full focus on their location and the venues nearby. The map view fills the entire screen similar to an application such as Google Maps, as cutting off the map at the border felt as though you were cutting away information. The map remains centered on the user's location as they walk. There is a tab bar covering the bottom of the screen, to allow the user to switch between the map and table detail view with ease.

Tapping on an annotation on the map will take the user to the Details View, with the relevant information for the venue tapped on.

Table View

The table view contains a list of the venues in order of shortest to furthest distance from the user. I calculate this by creating an array of just the name of a venue and its distance to the user, sorting the array from smallest to largest distance, and then creating a sorted venue array where the indexes of each venue match their corresponding index in the distance array. This is recalculated every time the user switches from the map to the table, instead of just initial load, to allow for updated results as the user moves around.

Each table cell contains the following details:

- The name of the venue
- The building of the venue
- An image to display the user's opinion on the venue (dislike, neutral, like)
- A disclosure indicator to inform more details will be provided

Like the map annotations, tapping on a table cell will take the user to the same details screen for that venue, to allow for a concise experience.

Opinion images are assigned based on user selection on the details screen, and are defaulted to a neutral dot if the user has no opinion on the venue, or has not used the application before.

Details View

The back button on the details view will take the user back to the view they came from, to allow for a smooth navigation. I programmed the user opinions of a value by creating a boolean array the same size and order of the buttons on the view. If a button was selected, it was true in the array and the other two were false. I then used this boolean array to set the fill of the buttons, indicating which has been selected. Selecting a button automatically deselects other buttons. These opinions are stored in User Defaults.

Despite being unable to get Core Data functional without duplicate values when fetched (and, therefore the functions aren't called in my final submission), I still wrote the functions and lay the foundation for Core Data to be used in the application.