

MyPi: A Large Pizza Order Generator



Jackson Hacker
Computer Science



Baru Yogesh
Computer Science



Dr. Badri Vellambi
Faculty Advisor

Problem Overview

When someone orders pizza for a large group, they usually just order cheese and pepperoni, hoping nobody will complain. We seek to go beyond this, and create an application that allows a group of people to order pizza that satisfies everyone's preferences. The underlying problem is that, given a set of user preferences, how can we match all of our users to pizzas that will satisfy everyone's unique preferences?

The result of our calculations is successful if it satisfies the following criteria:

- All users are satisfied, meaning they get a pizza that has only toppings that they like, and no toppings that they dislike.
- The quantity of each selected pizza is proportional to the number of users that would be satisfied by that pizza.
- The total quantity of pizzas is proportional to the total number of users.

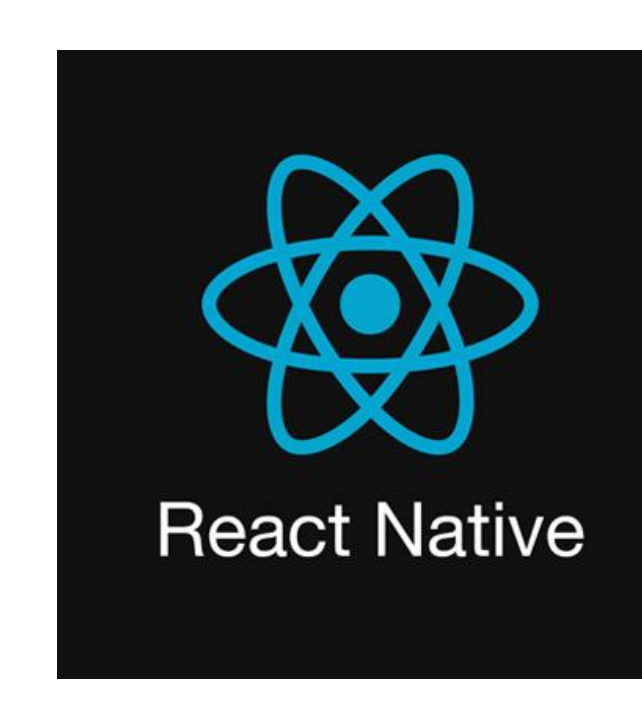
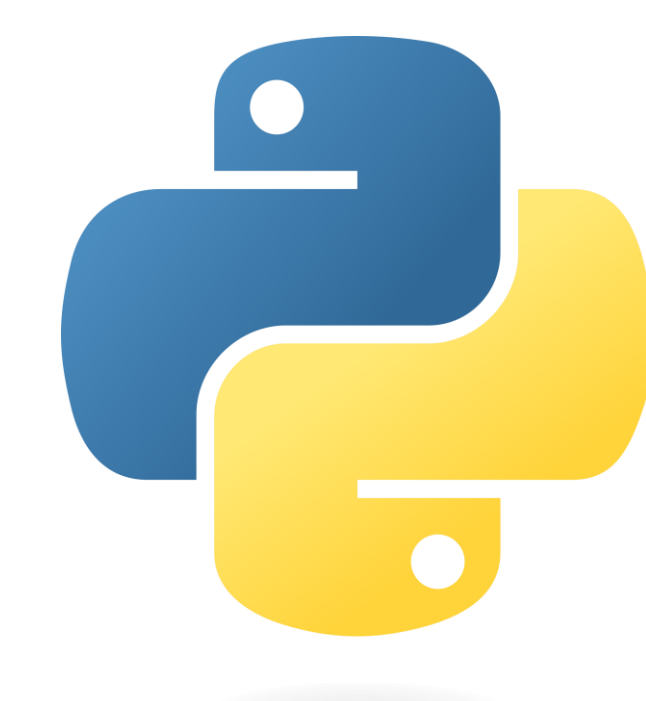
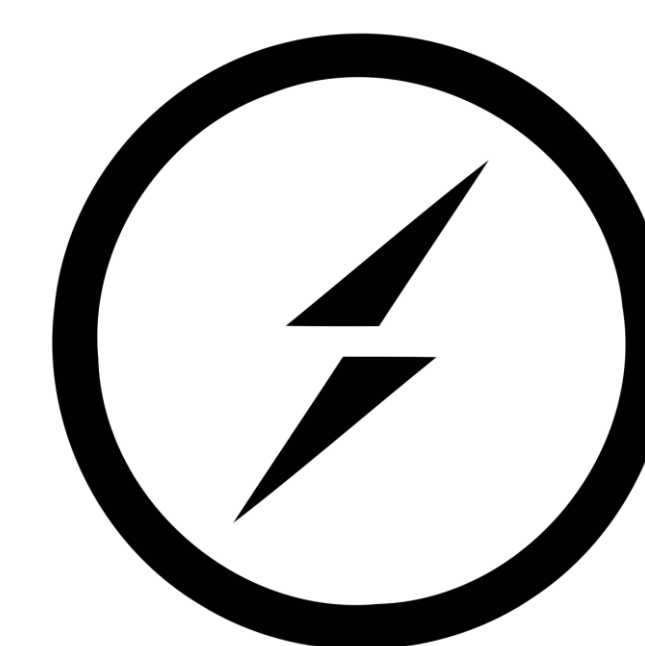


Image 1: Technologies Used-
JavaScript, React Native, Socket.io,
Python

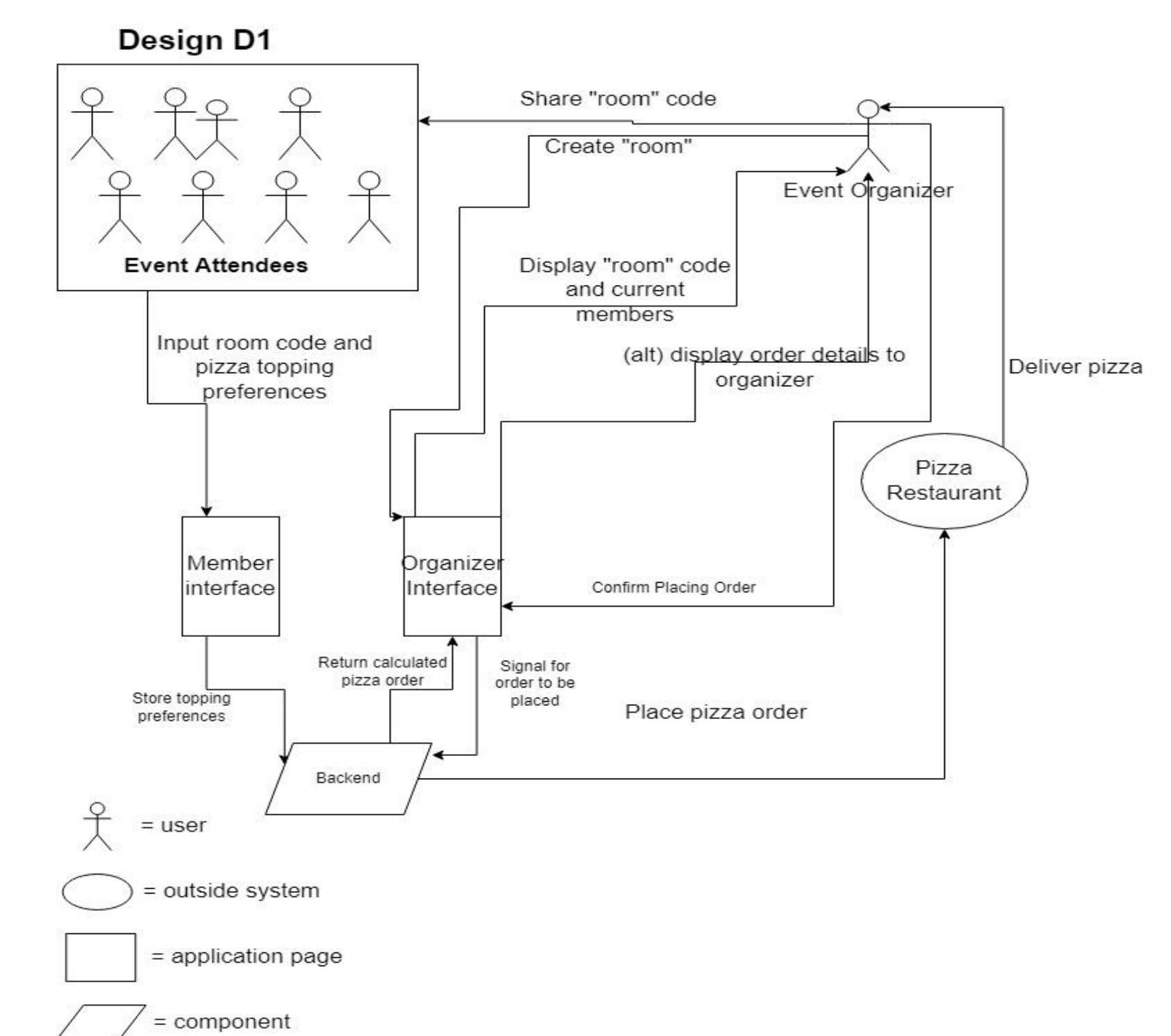


Figure 1: High-Level Design
Diagram

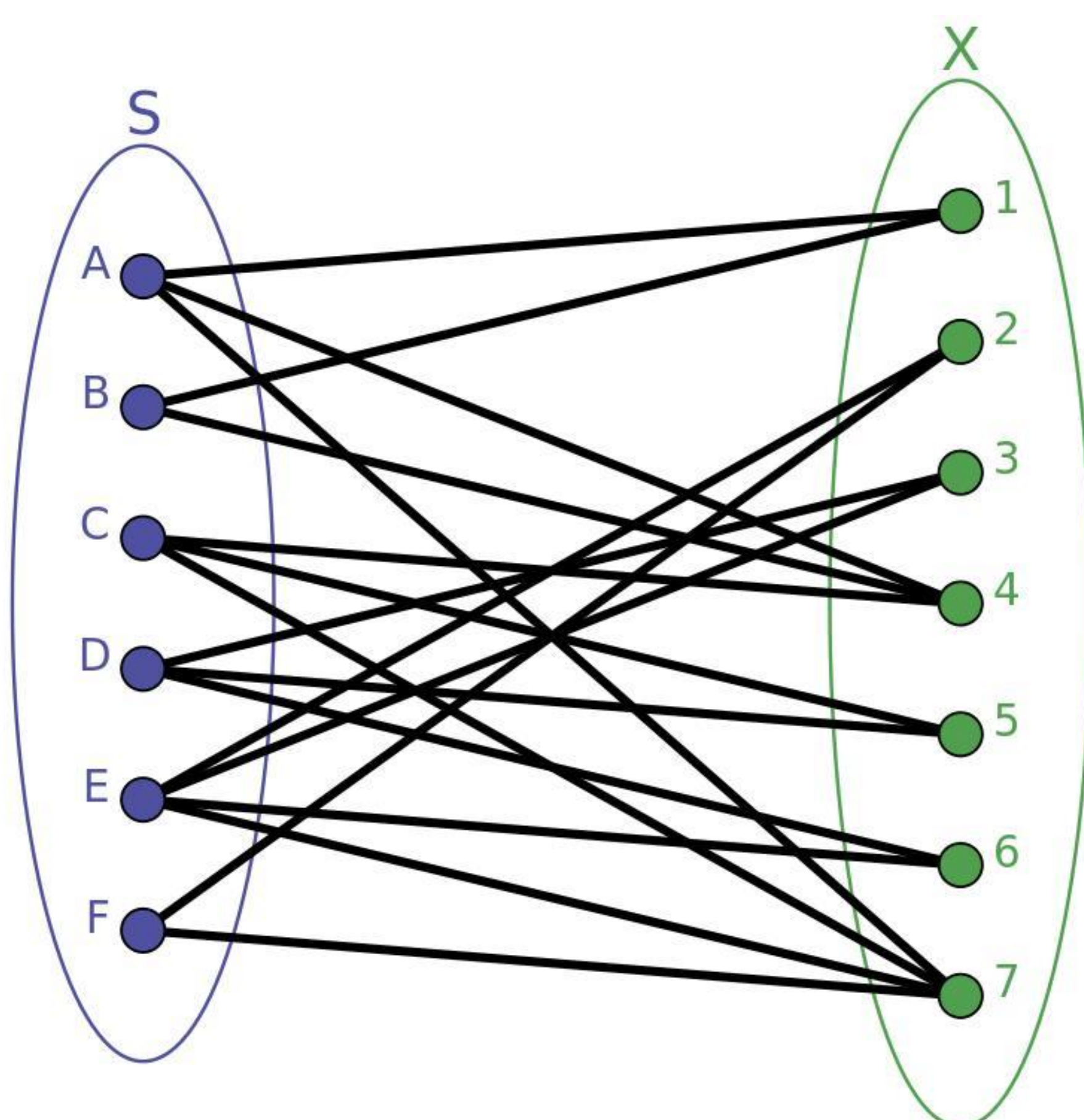


Figure 2: A Sample Bipartite Graph

Solution: Pizza Generation Algorithm

We modeled user preferences as a bipartite graph (see Figure 2). One set of nodes in the graph represents users, and the other represents "candidate" pizzas (i.e., pizzas with different combinations of toppings that people like), and an edge between a user and a node indicates that the user would be satisfied by that pizza. The algorithm applies a variation on a minimum vertex cover in order to find a set of pizzas that satisfies all users.

General Greedy Algorithm Strategy:

- Select the pizza with the most outgoing edges (the most liked)
- Mark all users that like the selection as "satisfied"
- Continue selecting the pizza liked by the greatest number of unsatisfied users until everyone is satisfied
- Pizza quantities scaled by users satisfied

Constraints on Algorithm:

- "Greediness" is altered such that the algorithm prefers pizzas with more toppings (to a point)
- Can set maximum toppings per pizza

Results: Mobile App

There are two classes of app users: Order Administrators (the person organizing the pizza order), and General Users (everyone included in order)

App Flow for Order Admin:

- Create an order using button on home page
- Adjust optional parameters (e.g., maximum toppings per pizza)
- Obtain "order code" and share it with others
- Observe as users submit preferences
- Once everyone has submitted, generate order
- The result will be our algorithm's suggestions for what pizzas to order

App Flow for General Users:

- Obtain "order code" shared by Order Admin
- Enter name and select desired toppings
- Submit your preferences to be used in the order

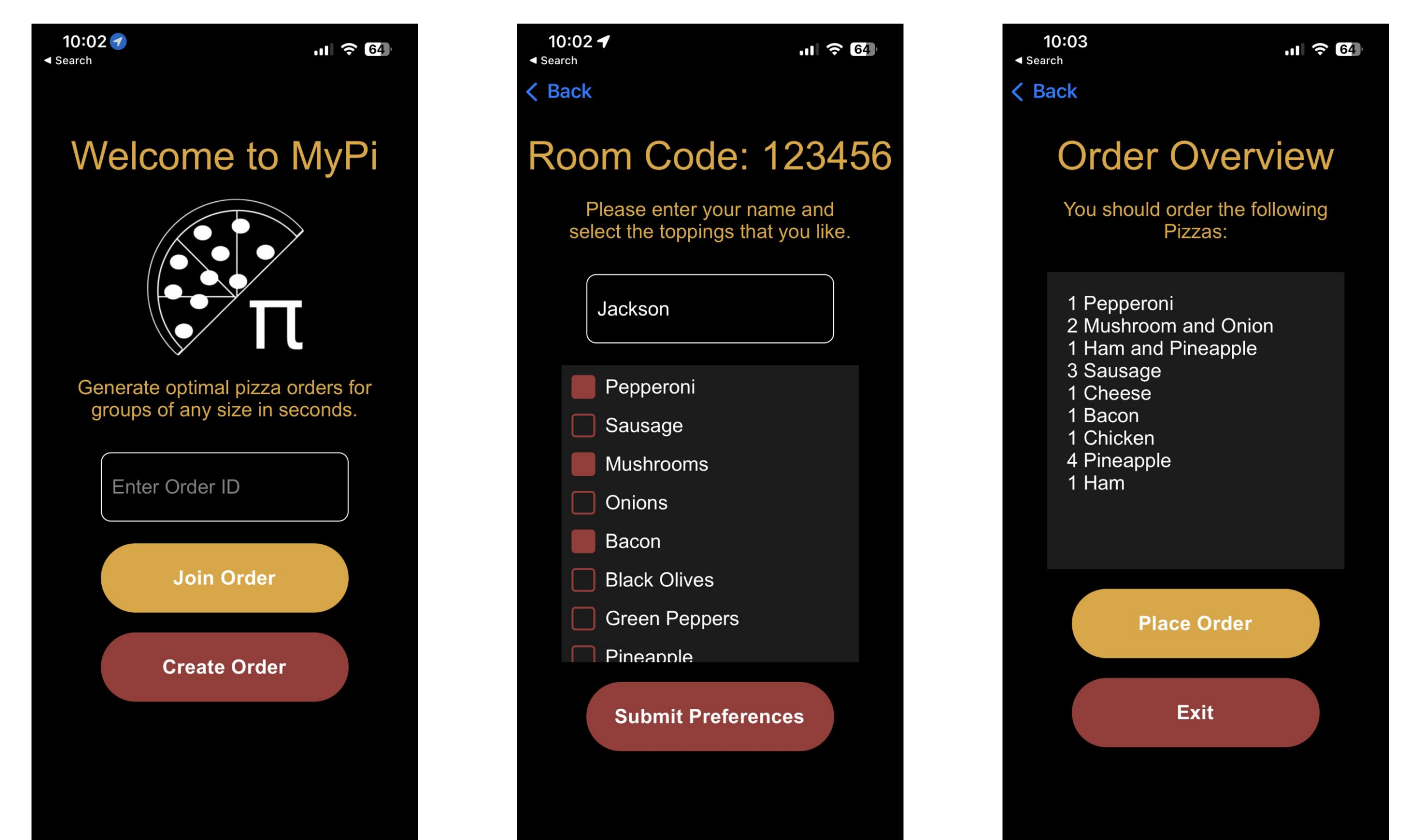


Image 2: Sample Screenshots from
Mobile Application