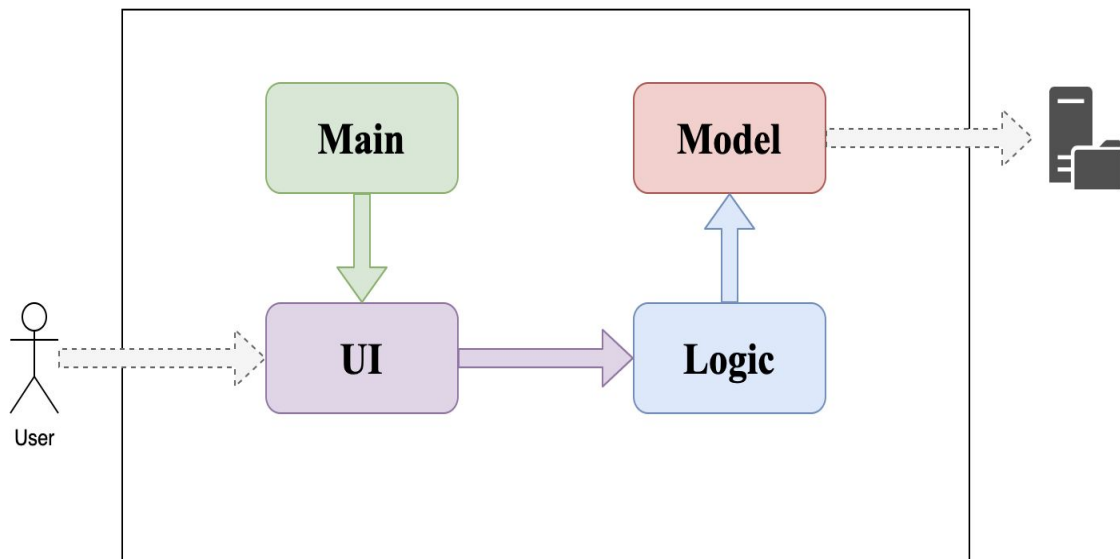


Developer Guide

Design

Architecture



Feature 1. Architecture Diagram

The Architecture Diagram given above explains the high-level design of our App. Below are the main components of our product.

1. **Main:** Our main class has one class called `Main.java`. It is responsible for,

At app launch, the `Main.java` will initialize the **UI** to start waiting for the input from the users.
2. **UI:** The user interface of our App.
3. **Logic:** The logic flow of our whole App.
4. **Model:** Holds the data of our App in memory.

The following section is a more detailed description of these four components.

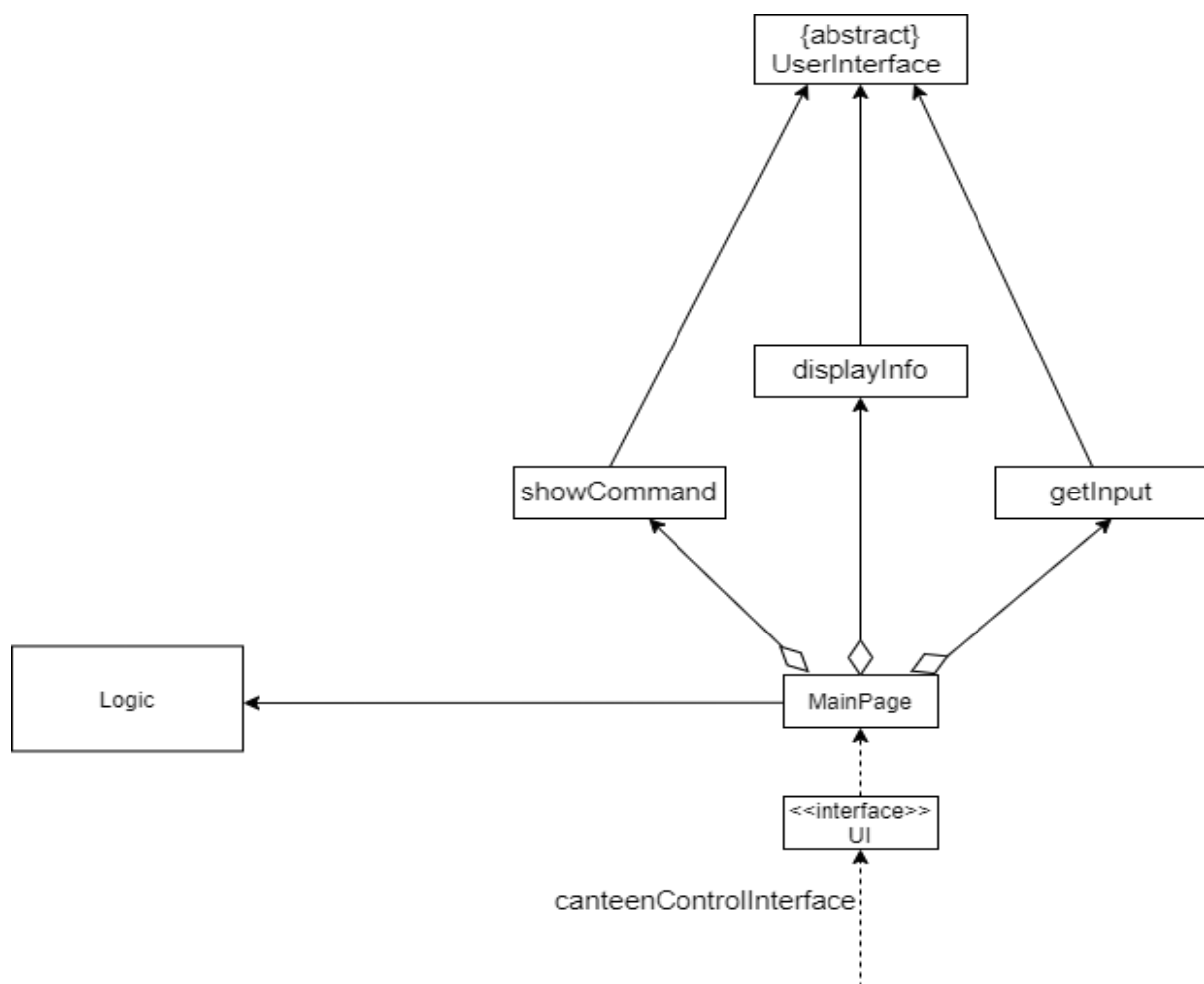
UI component

API: canteenControlInterface.java

The canteenControlInterface is mainly managed in MainPage, which contains several parts: showCommand, displayInfo, getInput. All of these should inherit from an abstract class UserInterface class.

The canteenControlInterface can

1. Show all possible executable commands in numeric order
E.g. 1: Get open canteens
2: Get open stalls
2. Get user input (number) and use the Logic component to execute the corresponding user's command
3. Display the results of user and program interactions

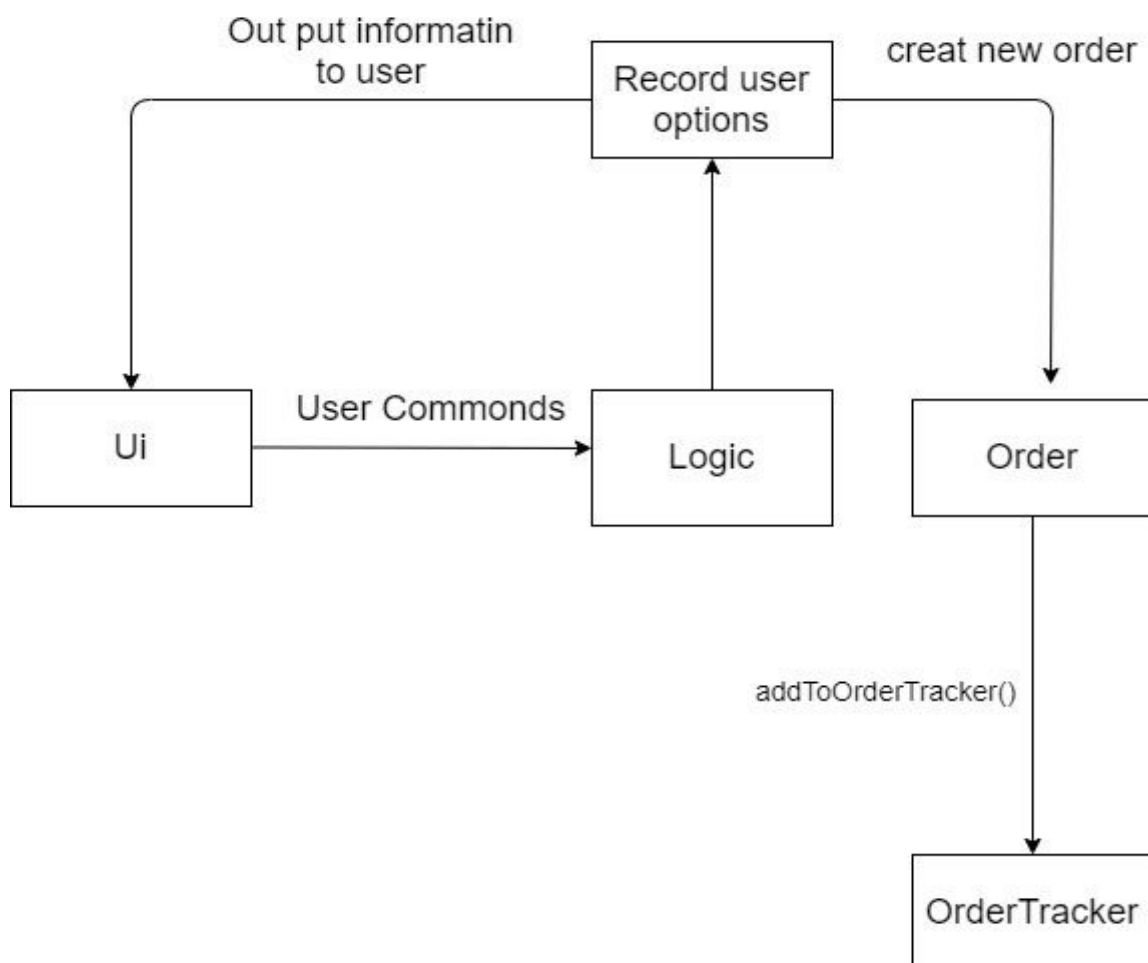


Logic component

API: Logic.java

1. Logic accepts user commands from Ui.
2. This results in an OrderManager object which keeps track of all user inputs.
3. The OrderManager execution can affect the OrderTracker (add a new order).
4. The result of the command execution is encapsulated as an Order object stored in the OrderTracker.
5. In addition, the OrderTracker object can also instruct the Ui to perform certain actions, such as displaying information to the user.

Given below is the Sequence Diagram for interactions within the Logic component.



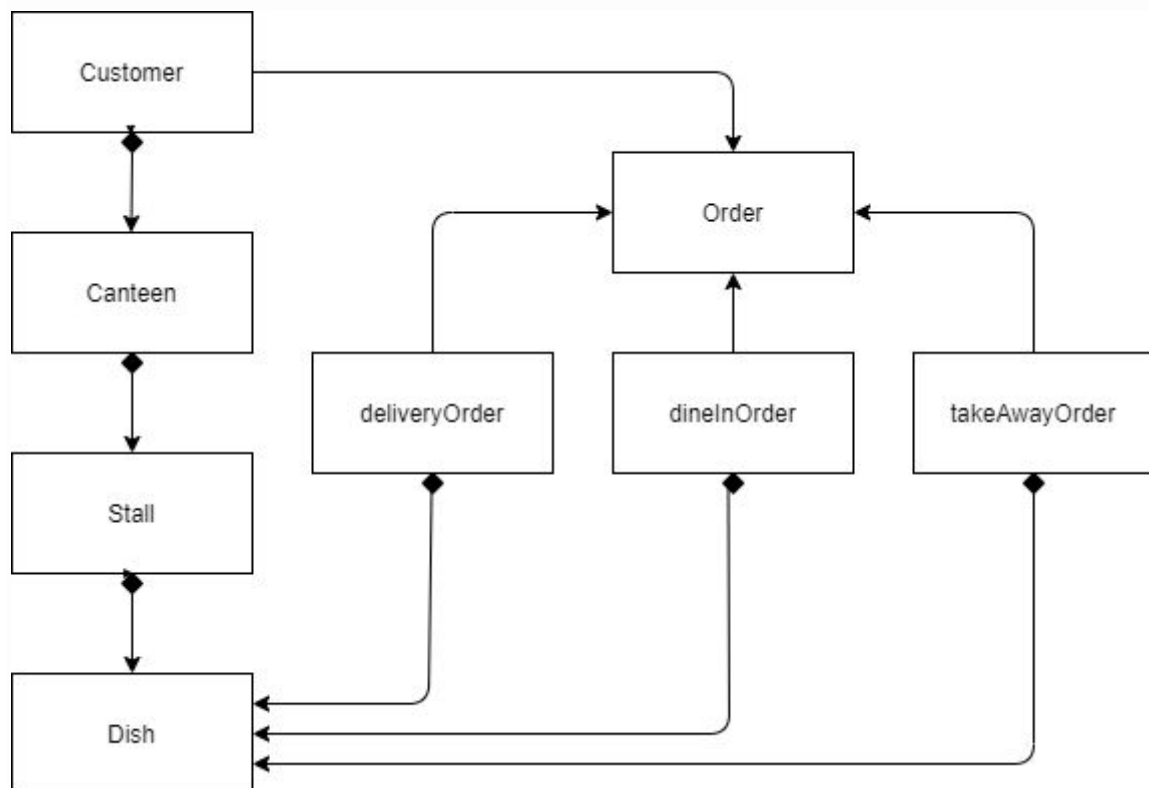
Model component

API: Model.java

The model component stores all information about canteens, customers, dishes, orders, and stalls.

1. Canteen contains the list of stalls
2. Stall contains the list of dishes
3. Customer contains the list of open canteens and initializes Order objects.
4. Order contains the list of dishes that are ordered

Does not depend on any of the other components.



Implementation

Step1: Call `canteenControlInterface.java` to print out all available commands in numeric order. The user can see all available commands and choose the corresponding index.

Step2: Invoke `getName()` to ask the user to input the user name. `canteenControlInterface.java` will collect the user's input of name and send it to the `Logic` part for the step4.

Step3 : Create a `Customer` object. It will initialize a `Customer` object with user name and a given id.

Step4: Invoke `getTime()` to ask the user to input dining time. `canteenControlInterface.java` will collect the user's input of dining time and send it to the `Logic` part for the step4.

Step5: Call `getOpenCanteen()`. The User can choose to check open canteens. The names of open canteens will be printed through `canteenControlInterface`.

Step6: Get the input of which canteen user wants to go and call the corresponding `getStallList()` function in the `Canteen` object.

Step7: Get the input of which stall user wants to go and call `showMenu()` function in the `Stall` Object.

Step8: Get the input of the dish the user chooses from the provided dish list and ask the user whether dinein, takeaway, or delivery.

Step9: Based on the user's choice of three kinds of order, create an `Order` object with canteen name, stall name, a list of dishes, and customer name correspondingly.

Step10: Call `showOrder()` function to show the information of the customer and the order to the user.