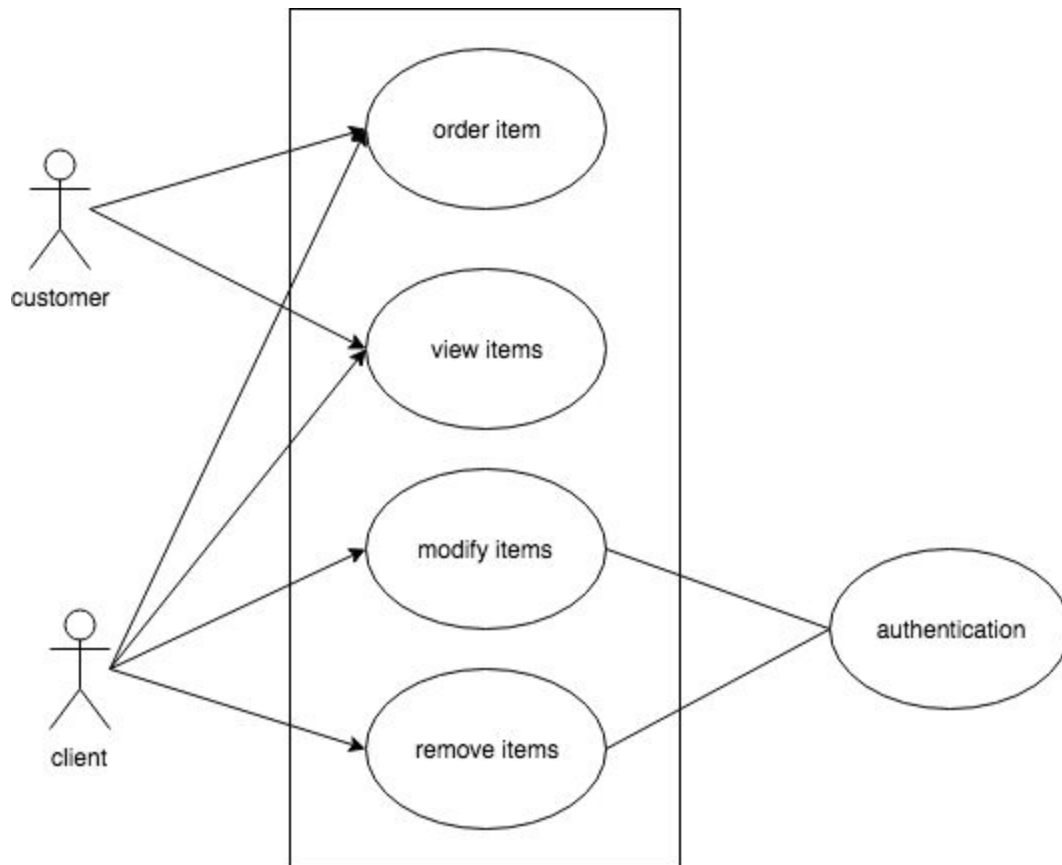


Design overview

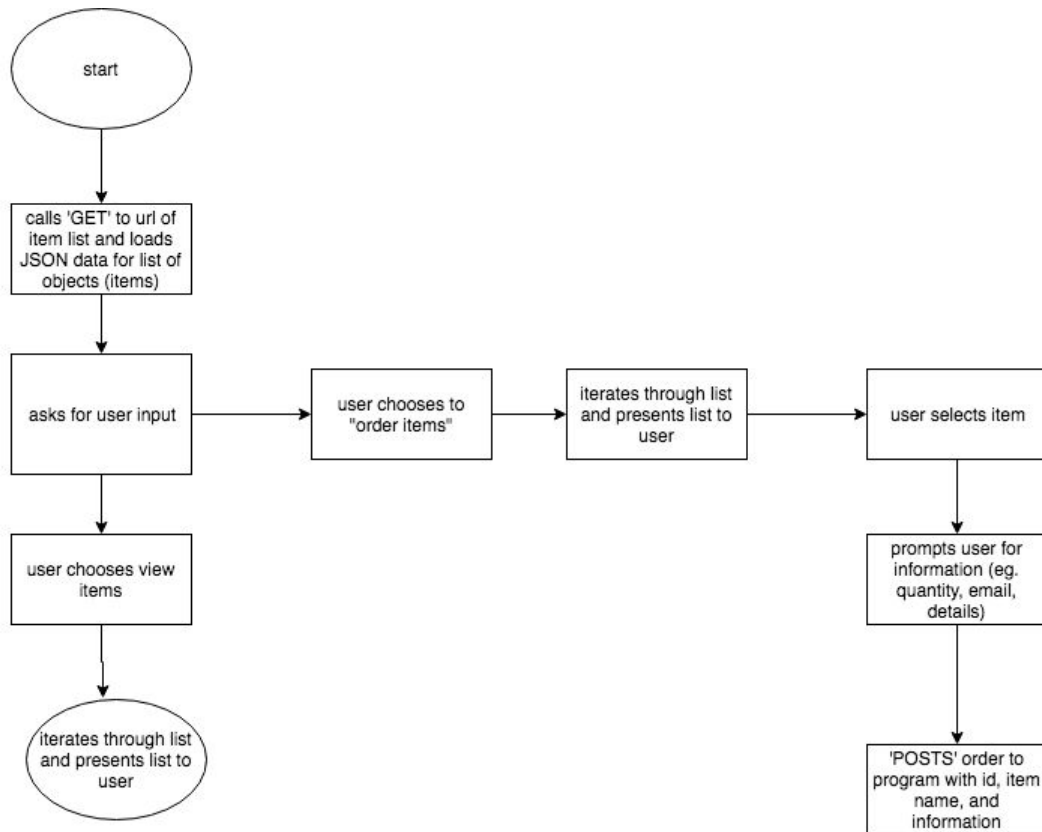
Major components:

- Python back end
- Javascript front end



Comment: UML diagram of the backend design, two users, with the customer having limited access to the program

- Customer inputs to order and view items
- The client is able to also modify and remove items, requiring authentication
- The system operates online, hosted through a web service



Comments: customer order chatbot interface uses this back end design to return output

- Functions will connect each major action of the front end
- Example: function will load and format data so that it can be used in a drop-down menu for the user to select.

Test Plan:

Success criteria	Test method
The program allows the client to modify item data within the program	The program is user-friendly and tells the client the choices they can make to modify item data. By entering the id of the item or a similar method, the client will be able to modify items stored in the program
The program allows the client to access customer order data	The client is able to see data that the customer inputs into the web frontend through commanding the program to show orders.
The program interface will guide the customer on the input to avoid errors(friendly)	The customer that uses the client's program will be told what to input and options by output by the program so that they know what to input.
The program requests relevant information from the customer (product, special requirements etc.)	The program asks customers for the information that the client needs, it will ask for the product name, customer contact information, and additional information.
The program allows customers to input information which is stored to be accessed.	Customers will be able to enter data into the program's frontend through javascript and prompts. The Information is sent through a 'POST' request to be stored in the web service.
The program parses information from customer input and arranges it in a table or similar structure for client	The data is arranged by the program into a data structure and sent to the client.
The program will automatically inform the client of data entry errors	If the customer enters the incorrect format of data that will affect the program, for instance, no email, the program will tell the customer that the error exists
The program warns customer if insufficient information has been provided, potentially because of data entry errors	If the customer fails to input required data such as contact information the program will output that the customer has not achieved this.

