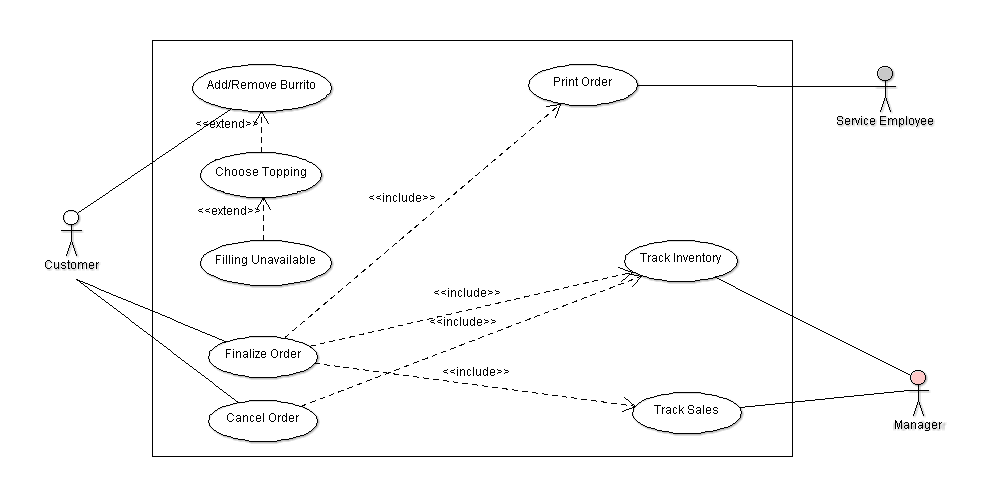
Self-Service Point of Service (POS) Program:

A small city quick service burrito restaurant chain would like to minimize customer wait time for their products and would like to develop a simple self-service point of service (POS) program that would present the customer with the various permutations of currently available fillings to include meat, cheese, rice, beans, salsa, veggies, etc. and place the order into a first in, first out queue to be prepared by the service employees. The customer can select a regular white tortilla or one of 4 flavored tortillas for an extra charge. The customer can select 1 type of rice, beans, meat (or hummus for vegan/vegetarians), and cheese with additional choices of these items will incur extra charges. The customer can select 2 types of salsa with additional choices will incur extra charges. Vegetables and other additions are included in the price of the burrito. The self-service point of service program would need to be able to push the order into the queue and identity what unique id was assigned to the order so the customer can pick up the right order. The confirmation of an order submission would also contain the total calculations of charges for the burrito that was specified in the order. An active view of the inventory of the fillings available would be controlled and when a certain choice is out of stock, the program would reflect that circumstance so no orders come across the system that cannot be filled. Orders that are not paid for, cancelled, or abandoned should be allowed to be deleted from the system and any inventory used in the tabulation of that order would need to be returned to available inventory if applicable. The system is required to track persistent data in a relational database which also needs to be architected with atomicity to ensure transactional integrity. The program must be multithreaded to ensure quick user response and interactivity while processing the order. The program must ensure inventory stock count integrity through use of semaphores or similar technique to prevent deadlocks or race conditions within the software. The first phase of the program design is to take orders, calculate the total cost of the order, and keep track of currently available inventory of filling options. Tracking the currently available inventory would also help trend ingredient orders to keep produce and other preparation stock as fresh as possible. The second phase would be to track weekly and monthly sales of the burrito restaurant chain as well as trend test and special filling items. The third phase would be to integrate self-checkout (via cash or credit) at the POS terminal and to securely transact with the burrito restaurant chain’s bank to electronically deposit all funds for any given day of business and expanded functionality to accept orders via web technologies such as web services with the burrito restaurant web front end, mobile applications, text messages, fax, or email.

Use Case Diagram (Phase 1):



Prioritized Use Cases:

* Customer initiates order
* Customer selects topping
  + Additional charge is totaled (if necessary)
  + Unavailable fillings are shown but disabled
* Customer submits or cancels the order
  + Inventory is subtracted from available and current orders not submitted are updated
* Order is printed for service staff to prepare the order
* Available inventory is tracked for management
* Real-time sales are tracked for management