

CS 744 - Software Project Management

FrontDash

FrontDash is a company that delivers food of customers' choice at their door step. A customer using FrontDash's service will order his/her favorite food from one or more of his/her favorite restaurants, listed at the company's website. An employee of FrontDash will get the food from the selected restaurant and deliver it to the customer. The aim of this project is to develop a web-based software system that will assist on all operations performed by the three parties - FrontDash company, restaurants and customers, and communications among them. Consequently, **the software system must have three independent components that can be concurrently used.**

Restaurant

A restaurant that wishes to sell food through FrontDash must register with the company. While registering, the restaurant should provide

- restaurant's name - must be unique within the application.
- street address - this is important because FrontDash's driver will have to get the food from this address.
- contact phone number(s) - each phone number must be 10 digits long.
- contact person - this person will be contacted for any questions, concerns, inquiries etc.
- email address of the restaurant - required for communication between the restaurant and FrontDash.
- hours of opening - must indicate the duration for each day of the week.
- menu - contains a set of food items. Each food item must include
 - name of the item
 - picture of the item
 - price of the item
 - availability - may be indicated as AVAILABLE AT ALL TIMES, AVAILABLE ONLY on some days and/or at some times, or UNAVAILABLE (may indicate sold out, temporarily unavailable or not available until further notice).

There are two types of users in a restaurant - manager and employees. There is only one manager account; it should be hard-coded in the application. A manager should be able to

- login and logout.
- add an employee account - the details should include (i) employee's full name (must be unique within the restaurant), (ii) username (lastname PLUS two digits) and (iii) auto-generated password.
- delete an employee account
- register the restaurant with *FrontDash*
- withdraw from *FrontDash* - by invoking this operation, the restaurant will no longer be in business with *FrontDash* .
- change contact information

The following minimal set of functionalities must be available to employees of the restaurant:

- Login and logout
- Change the password - must be done right after the first login. After first login, an employee can change the password at anytime.
- Change the menu - This includes any change to the details of any food item.
- Change opening hours

The front page for the restaurant must include a nice picture of the restaurant, a small description of the restaurant, and its operating hours. You may include the street address as well, but in this project, customers will order food through *FrontDash* and so street address need not be displayed. There must be a link to this page from *FrontDash* so that customers can browse through the menu and should be able to order food.

It should be noted that the registration process by a restaurant only makes a request to *FrontDash* . It is up to *FrontDash* to approve or disapprove the request. The restaurant will be included in *FrontDash* only after its registration request is approved. Similarly, when a restaurant withdraws from *FrontDash* , a notification will be sent to *FrontDash* application which will then send a confirmation back to the restaurant.

Customer

A customer orders food from a restaurant through *FrontDash* . Any direct communication between customers and restaurants will not be accounted by *FrontDash* and so this project does not care about such direct communications. A customer who wishes to order food should first browse through the restaurants displayed on *FrontDash* 's home page. Once selecting a restaurant, they can browse through the menu in that restaurant and select the food items they want. After confirmation, a bill should be displayed to the customer along with payment information. Notice that there is no login required for a customer. The following minimal set of functionalities must be supported for a customer:

- Browse through the list of restaurants - The details for each restaurant should include the name of the restaurant, an optional icon/logo for the restaurant and a status indicating whether the restaurant is open at that time. Customer should be able to select a restaurant for the next step.
- Browse through the menu in the selected restaurant - Customer should be able to select one or more food items from the menu. While selecting the food items, customer should be able to indicate quantity as well.
- Confirm the order - This should take to the billing page which should display the following details:
 - Restaurant's name from which the food is ordered.
 - Current date and time of order - This is important and will be used by the application later.
 - Food items ordered - For each food item, display
 - * name of the item
 - * price of the item
 - * quantity of the item
 - * subtotal for that item
 - Total of all subtotals before service charge.
 - Service charge - 6% for this problem.
 - Tips - initially empty; customer should be able to enter the tips - you can use percentage or fixed amount.
 - Grand total, after customer enters the tips.
- Pay for the bill - After verifying the bill, customer should initiate payment process. For simplicity, payments can only be made by credit cards. To pay, customer should enter the following credit card information:
 - Type of card - VISA, MasterCard, Discover etc.
 - Credit card number - must be 16 digits
 - First and last name of the credit card holder
 - Billing address
 - Expiry date - Month and year
 - Security code - three digits

Once customer enters credit card information, it should be verified through a third-party software. You may imitate a third-party software or you can use an actual one. After verification of credit card, customer should be prompted to enter the delivery address. A valid address should include the following:

- Building number
- Street name
- Apartment or unit number (optional)
- City name
- State
- Contact person's name
- Contact person's phone number

'State' is required because sometimes the city might be at the border of two states (e.g., La Crosse and La Crescent). When done, customer should be given an order number and an approximate time of delivery (to be computed by the application; details of this computation will be explained later in the section on *FrontDash*).

FrontDash

There are some tasks to be performed by *FrontDash* application without user intervention. For example, when a restaurant registers itself, the request should be added to a queue. It will be later reviewed by an administrator (to be discussed shortly). If a restaurant withdraws from *FrontDash* , the withdrawal request must be queued. An administrator will review the request later and send a confirmation to the restaurant. Similarly, when a customer places an order, verification of credit card information and generation of order number should be done by the *FrontDash* application. The order should be queued which will be reviewed by a staff member shortly.

There are two types of users in *FrontDash* - staff and administrator. There is only one administrator in *FrontDash* . The minimal functionalities for an administrator are listed below:

- Login and logout - a hard-coded administrator account should exist.
- Approve or disapprove a request from a restaurant.
- Add a new staff account - must include (i) full name of the staff member (must be unique within *FrontDash*), (ii) username of the account (lastname PLUS two digits), and (iii) auto-generated initial password.
- Remove a staff account.

The minimal set of functionalities for staff members in *FrontDash* include the following:

- Login and logout.
- Change password - Notice that first-time login should initiate this functionality automatically.

- Retrieve the first order from the order queue. Enter it into the database.
- Compute the estimated delivery time based on the restaurant's address and the customer's address. The calculation should be done by *FrontDash* application; the staff member simply initiates the computation.
- Send a driver to fulfill an order.
- Record the order status and amount from a driver after delivery to a customer.
- Hire a driver
- Fire a driver

General rules of implementation

The following rules must be implemented throughout the project:

- Passwords should never be displayed, nor should be visible when stored in the database. This requires that passwords must be encrypted/masked at all times.
- Every user input must be validated. For example, credit card numbers must be 16 digits long with the first digit not a zero.
- Phone numbers must be 10 digits long with the first digit not a zero.
- Names (both first and last names) must have a minimum of two letters.
- The outcome of any task should be demonstrated through the user interface. For example, while adding a user account, there must be a confirmation message displayed on the user interface. The user should not be required to open the database to check whether a new account has been created.
- Any action that removes an entry from the database must be preceded by a dialog to confirm with the user. For example, while deleting a user account, the dialog "Are you sure you want to delete the account <username>?" must be displayed on the user interface. Only after getting confirmation from the user, the account should be deleted. This process ensures protection against accidental deletions.
- All three components (restaurant, customer interface and *FrontDash*) must be developed as independent web-applications and therefore must be able to use them concurrently. Though the customer interface is part of *FrontDash* application, it should be possible to use it independent of the operations performed by *FrontDash* administrator and/or staff members. Notice that each restaurant should have its own web application. For ease of demonstration, it is sufficient to create a minimum of three restaurants.