

## COMP S380F Web Applications: Design and Development Group Project (30%)

In this project, you are required to form a group of 4 students (in special case with approval of the course lecturer, a group of 3 students is allowed). It is supposed each of the members share similar workloads in the project.

**Theme:** You are required to implement a web application for a **Fast Food Ordering System**.

### Basic Features (70% of the project):

1. Your web application should fulfil the following basic requirements on web pages and functionalities:
  - a. Using major techniques introduced in the lectures and labs, like **Java EE, HTML5, CSS, Javascript**. You are NOT allowed to use non-Java EE server for the system.
  - b. You are required to use **Spring MVC** and **Spring Security**. Using Spring Boot is NOT allowed.
  - c. Dynamic pages generation based on user's input or request.
  - d. Using the **Apache Derby database** as backend for data storage (but not necessary for uploaded files).
  - e. The web application should be easy to use for normal users.
2. In the basic part, you must implement ALL of the following features to receive full mark:
  - a. Website hierarchy:
    1. The website has an **index page**, which shows the list of menu items for ordering.
    2. Each menu item has an **item page**, which shows
      - its item description,
      - some photos (if any) of the item,
      - price of the item,
      - availability of the item, and
      - a list of comments from registered users.
    3. A registered user can order the item by adding it to the **shopping cart** and check out.
    4. The website administrator can mark an item as available/not available for ordering.
  - b. User registration and login:
    1. Information includes username, password, full name, phone number, and delivery address.
  - c. Unregistered users can read content in the index and item pages except making order and comment.
  - d. Registered user can read content in the index and item pages, and do the followings:
    1. Adding an item to the shopping cart.
    2. Check out the shopping cart.
    3. Update the personal information except the username.
    4. Write comments in an item page.
  - e. The website administrator can do anything a registered user can do, and the followings:
    1. Edit (add, remove, update) the list of registered users and their information.
    2. Change the availability of an item.
    3. Delete an item page or a comment in an item page.

Note that all the features are supposed to be fully functional.

### Additional Features (30% of the project):

You are required to implement N features in the following list, where N is the number of members in your group.

1. Ordering history page of a registered user
2. Adding and showing favorite menu items for a registered user
3. Drag-and-Drop of files for uploading photos
4. Batch uploading of photos
5. Multiple languages (e.g., English, Traditional Chinese)
6. Storing photos to the Apache Derby database
7. Other features proposed by your group are also acceptable, but please consult the course lecturer for approval.

### Submission and Assessment:

#### Source code submission

**Deadline: ~~May 2~~ May 9, 2021 (Sunday) 23:59**

- Select one group member as the group leader to submit the source code of the web application.
- Zip the maven web application project (please *clean* the project in NetBeans first to remove all the compiled files that are big in file size).
- Prepare one single SQL file for inserting initial data to the Apache Derby database.
- Prepare one single text file *readme.txt*, which contains
  - the database name, username & password for your Apache Derby database,
  - the usernames and passwords for an administrator account and a registered user account, and
  - the list of additional features implemented and any necessary details for running them successfully.
- Submit the 3 files (web application zip, SQL and *readme.txt*) to the OLE submission page.

#### Assessment

- Your web application will be tested by the course instructors according to a procedure that can test all the functionality.
- The web application will be built and deployed to a server computer, and other computers will be used as clients for testing the functionality. It is expected that all clients can access the web application, when the web application is deployed alive.
- Note that you should **fill in your database with enough initial data** such that functionality in your web application can be tested properly or understood more easily. For example, your database should already have the administrator account and some user accounts.