## Submission

**Code**: Submit your code through GitHub as usual. Do not submit code through Brightspace.

## You must use the following link to create the repository for your project solution. All commits must be done prior to the deadline. Commits after a deadline will not be accepted. Creating a public repository or sharing any of your code for any of these assessments will result in academic misconduct. For more information on how to use these links please refer to Week 2 Lab.

**See Teams for GitHub link**

**Documentation**: Submit your grading form through D2L Brightspace. (See below.)

## Introduction

You are an expert full-stack web application developer. While living on ichiban ramen noodles, you are attempting to create an innovative web application that will help home owners keep track of their inventory of household items for insurance purposes. When you have successfully created an MVP (minimum viable product), you hope to be contacted by investors to secure funding for your start-up company.

You have chosen the trendy name **HOME nVentory** for the application.

Through a small amount of market analysis and chatting with friends working in the insurance industry, you have determined that there is a need for a web application with the following basic functionality.

## Required Functionality

You have determined that the MVP must have the following functionality.

### Authentication

* Users must be able to login using an email address and password.
* Users must be able to logout.

### Registration

* Users must be able to register (create) an account consisting of an email address, password, first name, and last name.

### Manage User Account

* Users must be able to edit their account information.
* Users cannot delete their own account.

### Account Active/Inactive

* Users must be able to deactivate their account from the account page.
* If an account has been deactivated, the user cannot log in. Only a system administrator can reactive their account.

### Manage Inventory

* A home inventory item consists of category, name, and price.
* Users must be able to view their own items.
* Users must be able to add their own items.
* Users must be able to delete their own items.
* Users must be able to edit their own items.

### Manage Users

* A system administrator is able to manage all aspects of the system.
* System administrators must have the ability to view all accounts.
* System administrators must have the ability to create a new account.
* System administrators must have the ability to delete an account. Deleting an account will delete all of the associated items for that account.
* System administrators must have the ability to edit an account.

### Manage Categories

* System administrators must have the ability to view the list of categories.
* System administrators must have the ability to add categories.
* System administrators must have the ability to edit a category name.
* There is no functionality to delete a category.

## Usability

The website must be user friendly and easy to navigate. At a minimum, each page must display:

* An appropriate title
* A menu to navigate to other pages
* A logout link
* The current logged in user’s name

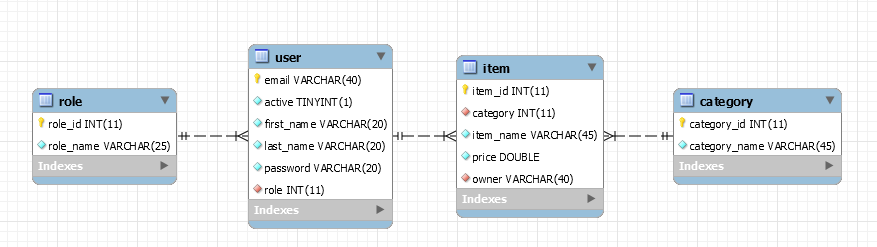
In addition, the website must gracefully handle all 404 file not found requests and all uncaught thrown exceptions.

## Application Design

You must design the web application using Java in NetBeans and MySQL database. You must use an n-tier MVC application architecture. You may use either JPA or JDBC.

## Database Design

You are given an SQL script that will create the ERD below. You may edit the database schema as necessary to adapt it to your liking as long as it still supports the required functionality.



At a minimum, you must include the following accounts:

* cprg352+admin@gmail.com,password
* cprg352+admin2@gmail.com,password
* cprg352+anne@gmail.com,password
* cprg352+barb@gmail.com,password

## Required Documentation

In addition to the NetBeans project, you must also submit a Grading Form that is completed according to the instructions.

## Optional Functionality

If you complete the required functionality perfectly, you will obtain a grade of 70%. To achieve a higher grade, you must complete some of the following optional functionality components. Choose from the options carefully. **Partially completed (or broken) optional functionality components will not receive any marks. Please see the grading form for how your final project grade is calculated.**

1. Companies

* Add a new table called company with two fields: ID and company names. Using inventorydb.sql, populate the table with a few company names. Add a foreign key to the users table to associate each user with a company.
* Implement a company administrator page similar to Manage Users for system administrators, except that company administrators can only manage users of the same company as them.

1. Manage Roles

* System administrators must have the ability to promote a user to a system administrator or a company administrator.
* System administrators must have the ability to demote a user from system administrator or company administrator to a regular user.

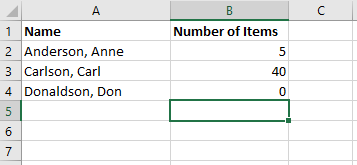
1. Security

* Ensure users are not able to view or make changes for which they are not allowed to.
  + Only users with the correct user name and password may log in.
  + Only authenticated users may access inventory and regular user pages.
  + Only administrators may access manage users and manage categories pages (and manage roles, if implemented).
  + Only the current user and administrators can access their account.
  + Only the current user can view their inventory
  + Only the current user can view, edit, or delete their items.

1. Registration email
   * When a user registers an account, an email is sent to the provided email address. The email contains a link which the user must click in order to activate their account. (This may require a database change.)
   * After the user has activated their account, send them a welcome email.
   * Security is required for this to ensure that users must use the link provided and cannot bypass this step.
2. Reset password
   * If a user forgets their password, they will have the option to reset their password. If they choose to reset a password, they will receive an email containing a link which will send them to a page to enter a new password.
   * Security is required for this to ensure that users must use the link provided and cannot bypass this step.
3. Reporting
   * Administrators would like to produce an Excel report.
   * Create a page only for administrators that will allow them to export a list of all of the users, the number of items each user has in their inventory, and the total value of the items for each user.
4. Undo delete
   * For all delete options (delete users by administrators, delete items by users), provide a way for the user to undo the delete.
   * Tip: This will probably require a database change.
5. Company subdomain branding
   * Provide a way to load a CSS file based on the subdomain of the website.
   * The name of the subdomain is the same as the company name.
   * Create a standard theme that is applied to the base domain.
   * Create a basic theme for each company/subdomain.
   * E.g. company1.home-nventory.com and company2.home-nventory.com would each have different styles
   * Suggestion: consider implementing a custom tag.
6. Stunning UI
   * Create an awesome-looking mobile-first, fully responsive user interface that renders well on mobile devices.
7. Multi-lingual support
   * Allow the user to select a different language and store their selection in a cookie.
   * All pages must be available into at least 2 languages with English being the default.
8. Deployment

* Deploy the application and database to a 3rd party server.
* Provide the URL in the comment field for grading.
* Free credits for AWS: <https://education.github.com/pack>

1. Alternate Authentication
   * In addition to the regular login, users can log in directly with a special link.
   * After a successful registration, users are sent an email containing a link. The link uses a random number (UUID/GUID) as the authentication credential.
   * This option will require a database change to store the random number authentication credential.
2. Item keyword search
   * Add a search box to allow administrators to search through all the items.
   * Display the item name and the owner of the item.
   * Allow partial matches, e.g. “b” finds “blender” and “books”
3. 2FA with SMS or email or Google Authenticator
   * User have the option of enabling two-factor authentication. When enabled, users must complete a two-step authentication to login. The first step requires that the user enter their email and password. If the credentials are valid, they are sent a text message or email with a random number. The second step requires the user to enter the random number to fully authenticate with the system. Make sure you add a telephone number to the user table.
   * <http://www.comparecellular.ca/text-messaging/>
4. For administrators, add a feature that exports an Excel spreadsheet containing a list of active regular (non-administrator) users sorted alphabetically by last name then first name, and the number of items for each user. E.g.



1. Upload picture
   * Allow the user to upload a picture of the item on the Manage Inventory page.
   * Pictures can be changed and are deleted when the user deletes an item.
   * A default picture is shown for items that do not have pictures.
2. Hashed passwords
   * Enforce all passwords to be salted and hashed using a strong one-way encryption algorithm.
3. Profile picture
   * On the user’s Manage User Account page, add the functionality to allow the user to upload a profile picture. The current profile picture is displayed on the Manage User Account page and also displayed beside the logged in user’s name.
4. Got any more ideas for an option? Talk to your instructor to see if it can be added to the project description.

## Submission

Submit your code through GitHub as usual.

Submit your grading form through D2L Brightspace.

## Sample Grading Form

## CPRG-352 Final Project Grading Form

### Required Components

**Instructions:** For each of the required components below, fill in your expected grade. Provide a comment if there is anything important you would like the instructor to know about that component. (e.g. How it works/not works, where to locate it within the browser, etc.)

|  |  |  |  |
| --- | --- | --- | --- |
| **#** | **Required Component** | **Comment** | **Grade** |
| 1 | Authentication |  | 5/5 |
| 2 | Registration | Registration works most of the time, but sometimes produces an error | 8/10 |
| 3 | Manage User Account |  | 10/10 |
| 4 | Account Active/Inactive |  | 10/10 |
| 5 | Manage Inventory |  | 5/5 |
| 6 | Manage Users |  | 5/5 |
| 7 | Manage Categories | Only able to view the categories. | 3/10 |
|  | Usability | Used a jspf and JSTL import to standardize the design. | 5/5 |
|  | Tested all functionality and completed Grading Form | Done! | 5/5 |
|  | Instructions  (adjust as necessary) | 1. Clone the NetBeans repository located at the following URL:   https://github.com/SAIT-ITSD/studentname   1. Run the SQL script located in /database/script.sql 2. Run the NetBeans project. | 5/5 |
|  | **Total (required)** |  | **61/70** |

### Optional Components

**Instructions:** For each of the optional components below, fill in your expected grade. If the component is not completed, partially completed, or broken, enter a grade of 0. If you did not complete any optional components, enter 0 in all boxes. Provide a comment if there is anything important you would like the instructor to know about that component. (e.g. How it works/not works, where to locate it within the browser, etc.) If you select more than 6 optional components, you may adjust the total to a higher number. You may not adjust the total to a number less than 30. This means to get 100% you need to complete a minimum of 6 optional components.

|  |  |  |  |
| --- | --- | --- | --- |
| **#** | **Optional Component** | **Comment** | **Grade** |
| 1 | Companies |  | 0/5 |
| 2 | Manage roles | After a system admin logs in, you will see a link to the manage roles page. | 5/5 |
| 3 | Security |  | 0/5 |
| 4 | Registration email | Tried it, but couldn’t get it to work. Removed code from project. | 0/5 |
| 5 | Reset password | Tried this one too. | 0/5 |
| 6 | Reporting | After an admin logs in, you will see a link to the reporting page. | 5/5 |
| 7 | Undo delete |  | 0/5 |
| 8 | Company subdomain branding |  | 0/5 |
| 9 | Stunning UI |  | 0/5 |
| 10 | Multi-lingual support |  | 0/5 |
| 11 | Deployment | Please find the website located at: <https://home-nventory.com> | 5/5 |
| 12 | Alternate authentication |  | 0/5 |
| 13 | Item keyword search |  | 0/5 |
| 14 | 2FA with SMS |  | 0/5 |
| 15 | Excel Export |  | 0/5 |
| 16 | Upload picture |  | 0/5 |
| 17 | Hashed password |  | 0/5 |
| 18 | Profile picture |  | 0/5 |
|  | **Total (optional)** |  | **15/30** |

### Final Project Grade

**Instructions:** Copy the grades calculated above into the table below.

|  |  |  |
| --- | --- | --- |
| Total (required) | Total (optional) | **Final Project Grade** |
| 61/ 70 | 15 / 30 | **76 / 100** |