CPRG-352 – Final Project Fall 2017

## 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 improve how employees organize notes. 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.

Through a small amount of market analysis and chatting with friends working in large corporations, you have determined that there is a need for employees to quickly create and organize personal notes. These notes are like a small post-it note that is used to jot down to-do lists, reminders, and random thoughts.

You have chosen the trendy name NotesKeepr for the application.

## Required Functionality

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

### Authentication

* Users must be able to login using a user name and password.
* Users must be able to logout.

### Registration

* Users must be able to register an account consisting of a username, password, first name, last name, email address, and company.
* The user must be able to select their company from a list.

### Account

* Users must be able to edit their account information.
* Users cannot delete their own account.
* Users cannot change the company they belong to.

### 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 or company administrator can reactive their account.

### Manage Notes

* A note consists of a title, contents, and date created.
* Users must be able to view their own notes
* Users must be able to add their own notes.
* Users must be able to delete their own notes.
* Users must be able to edit their own notes.

### Manage Users – System Administrators

* 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 notes for that account.
* System administrators must have the ability to edit an account.

### Manage Users – Company Administrators

* A company administrators belong to a specific company and are able to manage accounts of that same company.
* Company administrators must have the ability to view all accounts of the company that they belong to.
* Company administrators must have the ability to create a new account. New accounts must belong to the same company as the administrator.
* Company administrators must have the ability to delete an account of the same company as the administrator.
* Company administrators must have the ability to edit an account of the same company as the administrator.

### Manage Companies

* System administrators must have the ability to view the companies.
* System administrators must have the ability to add companies.
* System administrators must have the ability to edit a company name.

## UI Design

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

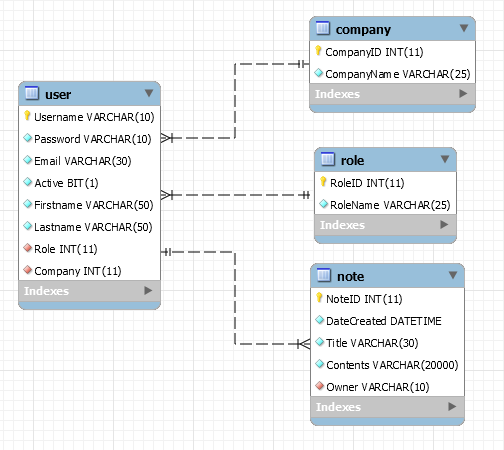
* An appropriate title
* A menu to navigate to other pages
* A logout link
* The current logged in user’s name

## 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:

* Admin accounts: admin, admin2
* Company administrator accounts: comp1, comp2
* Regular user accounts: user1, user2, user3, user4, user5, user6
* All passwords must be *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 (or all!) 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. Collaboration

* Users must be able to add another user to a note as a collaborator. You may choose how this is achieved, e.g. Selecting a user from a drop down list of all other users from the same company
* Users must be able to remove a collaborator from a note.
* Users cannot add themselves as a collaborator.
* Users must be able to view notes for which they are a collaborator.
* Users must be able to edit notes for which they are a collaborator.
* Users must not be able to delete notes for which they are a collaborator.

1. **Manage Roles**

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

1. **Error pages**

* Gracefully handle all 404 file not found requests and all uncaught thrown exceptions

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 notes and users pages.
  + Only administrators may access manage users pages (and manage roles, if implemented).
  + Only the current user and administrators can access their account.
  + Only the current user can view a list of their notes.
  + Only the current user can delete one of their notes.
  + Only the current user and the collaborators for a note may view and edit a note.

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 will 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
   * Company administrators would like to produce an Excel report.
   * Create a page only for company administrators that will allow them to select a start date and end date and then export a list of users and how many notes each user has created.
4. Undo delete
   * For all delete options (delete users by administrators, delete notes by users), provide a way for the user to undo the delete.
   * Tip: This will probably require a database change.
5. Company subdomain theming
   * 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.
6. Stunning UI
   * Create an awesome-looking mobile-first, fully responsive user interface that renders well on mobile and desktop devices.
   * A CSS compiler (e.g. Sass or LESS) is optional
7. Unit Testing
   * Create unit tests for all repository and service layer classes.
   * Dependency injection is optional.
8. Multi-lingual support
   * Allow the user to select a different language.
   * All pages must be available into at least 2 languages with English being the default.
9. Deployment

* Deploy the application and database to a 3rd party server.
* Provide the URL in the comment field for grading.
* For access to a cloud server, please let the instructor know you are interested in completing this option and you will receive instructions.

1. **Public notes**
   * **Authenticated users have the ability to mark a note they own as public. Public notes are accessible to other users of the same company for viewing. Public notes cannot be edited or deleted by the public.**
2. Note keyword searching
   * Add a search box to the view notes page to allow the user to search through all of their accessible notes by keyword or keywords. Ensure the search works through their notes, collaborator notes, and public notes, if implemented.
3. 2FA with SMS or email
   * 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 username and password. If the credentials are valid, they are sent a text message 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/>

## Submission

Create a GitHub repository by navigating to: <https://classroom.github.com/assignment-invitations/db307594c0ebd8e6707400a92c5dd8ba>

Submit your code through GitHub.

Submit your grading form through D2L Brightspace.

## Sample Grading Form

|  |  |  |  |
| --- | --- | --- | --- |
| **#** | **Required Component** | **Comment** | **Grade** |
| 1 | Authentication |  | 5/5 |
| 2 | Registration | Registration works but allows for multiple users with the same username. | 8/10 |
| 3 | Account |  | 5/5 |
| 4 | Account Active/Inactive |  | 5/5 |
| 5 | Manage Notes |  | 10/10 |
| 6 | Manage Users – System Admin. |  | 5/5 |
| 7 | Manage Users – Company Admin | Sorry, I couldn’t get this to work. Only system admin can manage users. | 0/5 |
| 8 | Manage Companies | Able to add and edit companies, but changes not showing up in registration form. | 6/10 |
| 9 | UI/Navigation |  | 5/5 |
| 10 | Completed Grading Form |  | 5/5 |
| 11 | Instructions  (adjust as necessary) | 1. Clone the NetBeans repository located at the following URL:   https://github.com/SAIT-ITSD/...   1. Run the SQL script located in /database/script.sql 2. Run the NetBeans project. | 5/5 |
|  | **Total (required)** |  | **59/70** |

|  |  |  |  |
| --- | --- | --- | --- |
| **#** | **Optional Component** | **Comment** | **Grade** |
| 1 | Collaboration |  | 0/5 |
| 2 | Manage roles | After a system admin logs in, you will see a link to the manage roles page. | 5/5 |
| 3 | Error pages | Try this link (link) and you will see a security error page. | 5/5 |
| 4 | Security |  | 0/5 |
| 5 | Registration email | Tried it, but couldn’t get it to work. Removed code from project. | 0/5 |
| 6 | Reset password | Tried this one too. | 0/5 |
| 7 | Reporting | After a company admin logs in, you will see a link to the reporting page. | 5/5 |
| 8 | Undo delete |  | 0/5 |
| 9 | Company subdomain theming |  | 0/5 |
| 10 | Stunning UI |  | 0/5 |
| 11 | Unit testing |  | 0/5 |
| 12 | Multi-lingual support |  | 0/5 |
| 13 | Deployment | Please find the website located at: <https://www.noteskeepr.io> | 5/5 |
| 14 | Public notes |  | 0/5 |
| 15 | Note keyword searching |  | 0/5 |
| 16 | 2FA with SMS |  | 0/5 |
|  | **Total (optional)** |  | **20/30** |

**Final Project Grade**

|  |  |  |
| --- | --- | --- |
| Total (required) | Total (optional) | **Final Project Grade** |
| 59 / 70 | 20 / 30 | **79 / 100** |