Задача за практичен дел

Испитна задача

Потребно е да развиете апликација за менаџирање на мени на ресторан која ќе овозможи прегледување, додавање, уредување и бришење на ставки во менито. Растораните служат храна од следните типови: пици, колачи и кафе.

Функционални барања

* **(20 поени)** Потребно е на патеките / и /menu да го прикажете целосното мени на ресторанот со користење на темплејтот list.html.
  + Имплементацијата на оваа функционалност може да ја проверите со тестот test1\_list.
* **(20 поени)** Потребно е да се имплементира додавање на ставка со храна во менито на ресторанот. При клик на копчето **Add new menu** од темплејтот list.html, потребно е да се прикаже темплејтот form.html на патеката /menu/add, каде при клик на **Submit** ќе се креира и запише ново мени во базата на податоци. Потоа треба да се прикаже страната /menu.
  + Имплементацијата на оваа функционалност може да ја проверите со тестот test3\_create.
* **(10 поени)** Потребно е да се имплементира бришење на ставка од менито. При клик на копчето **Delete** од темплејтот list.html, потребно е да се избрише мени ставката од базата на податоци. Потоа треба да се прикаже страната /menu.
  + Имплементацијата на оваа функционалност може да ја проверите со тестот test5\_delete.
* **(20 поени)** Потребно е да се имплементира измена на постоечко мени. При клик на копчето **Edit** од темплејтот list.html, потребно е да се прикаже темплејтот form.html на патеката /menu/[id]/edit, при што во <input> елементите ќе се прикажат вредностите за ставката кој се изменува. При клик на **Submit** треба да се запише промената на менито во базата на податоци. Потоа треба да се прикаже страната /menu.
  + Имплементацијата на оваа функционалност може да ја проверите со тестот test4\_edit.
* **(20 поени)** Потребно е да конфигурирате најава на /login и одјава на /logout. Притоа, јавна треба да биде само страницата /, додека сите останати страници треба да се видливи само за ROLE\_ADMIN. Дополнително, кајlist.html копчињата **Edit**, **Delete** и **Add new menu** треба да се видливи само за ROLE\_ADMIN.
  + Имплементацијата на оваа функционалност може да ја проверите со тестовите test6\_security\_urls и test7\_security\_buttons.
* **(20 поени)** Потребно е да овозможите пребарување на ставки во менито според тип на мени и име на ресторан преку формата со id="filter-form" во темплејтот list.html. Резултатите од пребарувањето треба да се прикажат на истата страница, при што ќе се прикажат само менијата кои припаѓаат во селектираниот тип на мени и име на ресторан. Филтрирањето се извршува само според внесените полиња (ако се празни, се игнорира филтрирањето по тој критериум).
  + Имплементацијата на оваа функционалност може да ја проверите со тестот test2\_filter.

**ВАЖНО:** Сите споменати тестови се наоѓаат во класата mk.ukim.finki.wp.kol2021.restaurant.SeleniumScenarioTest.

Поставување на решението

Тестовите мора задолжително да ги извршите, затоа што на тој начин го испраќате вашето решение за прегледување. За проверка дали успешно е поставено вашето решение пристепете на патеката [wp.finki.ukim.mk/submit/[index]](http://wp.finki.ukim.mk/submit/index), каде наместо [index] ќе го наведете вашиот индекс, кој претходно ќе го поставите во TODO вредноста на текстот.

**Вашиот код треба да го архивирате (zip) и да го прикачите и на ispiti.finki.ukim.mk.**

**НЕ Е ДОЗВОЛЕНО МЕНУВАЊЕ НА ТЕСТОВИТЕ, освен внесување на вашиот индекс**

Тие ќе ви помогнат да ја проверите точноста на вашата имплементација, но и ќе испратат информации до нашиот сервер за тоа до каде функционира вашата апликација. Ќе се гледа кодот само на студентите кај кои барем една проверка (ExamAssert) ќе помине успешно.

Техничко упатство:

* Во пакетот mk.ukim.finki.wp.kol2021.restaurant.model веќе се креирани класите кои го репрезентираат моделот. Потребно е да извршите нивно мапирање со соодветните JPA анотации за моделот успешно да се сними во базата на податоци. Притоа важат следните услови:
  + Едно мени може да има повеќе ставки, а една ставка може да припаѓа на повеќе менија.
  + Идентификаторите за Menu и MenuItem треба да бидат генерирани.
* Во пакетот mk.ukim.finki.wp.kol2021.restaurant.service се веќе дефинирани интерфејсите за сервисната логика. За секој од методите имате опис што треба да биде имплементирано. Потребно е да се имплементираат овие интерфејси во соодветните класи во пакетот mk.ukim.finki.wp.kol2021.restaurant.service.impl. Во коментарите на методите се објаснети дополнителни услови (доколку ги има) кои треба да ги исполни методот.
* Класите од пакетот mk.ukim.finki.wp.kol2021.restaurant.repository треба да ги дополните со потребните методи кои ви се потребни за да ја овозможите функционалноста на имплементацијата на сервисниот слој. Тие треба да се изведат од класата JpaRepository од Spring Data.
* Потребно е да ја анотирате класата mk.ukim.finki.wp.kol2021.restaurant.config.DataInitializer и нејзините соодветни методи, така што при стартување на апликацијата ќе се изврши методот initData.
* Во класата mk.ukim.finki.wp.kol2021.restaurant.web.MenuController се дефинирани сите методи кои се потребни за да се имплементира менаџирањето со продуктите. Потребно е овие handler методи да ги мапирате со користење само на HTTP GET и POST барања. Во продолжение е подетален опис за секој од методите:
  + showMenus треба да го искористи list.html темплејтот за приказ на сите менија. Овој handler метод треба да се мапира на патеките / и /menu.
    - Аргументите на овој метод не се задолжителни и може да бидат null.
    - Во случај кога не се испратени аргументите (кога се null) треба да се прикажат сите продукти од базата.
    - Ако некој, или двата, од аргументите е различен од null, тогаш треба да се прикажат продуктите кои ги враќа MenuService.listMenuItemsByRestaurantNameAndMenuType.
  + showAdd треба да го прикаже темплејтот form.html. Овој handler метод треба да се мапира на патеката /menu/add.
  + showEdit треба да го прикаже темплејтот form.html, меѓутоа во секој од input елементите треба да биде пополнета соодветната вредност за менито кој се уредува. Овој handler метод треба да се мапира на патеката /menu/[id]/edit
  + create треба да го креира менито според аргументите на методот (патека /menu). Потоа треба да се прикажат сите менија.
  + update треба да се уреди менито според аргументите на методот (патека /menu/[id]). Потоа треба да се прикажат сите менија.
  + delete треба да се избрише менито со дадениот идентификатор (патека /menu/[id]/delete). Потоа треба да се прикажат сите менија.
* Дополнете ги темплејтите со соодветните **Thymeleaf** атрибути за да се постигнат бараните функционалности. Притоа, ако недостасуваат одредени елементи и атрибути, може да ги додадете, но **НЕ СМЕЕ** да ги менувате id и class својствата на тековните елементи. Во коментари се дадени описи за елементите кои треба да се повторуваат, како и кои методи од контролерот треба да се повикаат.
* Потребно е да конфигурирате најава на /login и одјава на /logout со Spring Security во класата mk.ukim.finki.wp.kol2021.restaurant.config.SecurityConfig. Притоа, јавна треба да биде само страницата /, додека сите останати страници треба да се видливи само за ROLE\_ADMIN. Дополнително, кајlist.html копчињата **Edit**, **Delete** и **Add** треба да се видливи само за ROLE\_ADMIN.
  + Доколку го имплементирате ова барање, потребно е да ја закоментирате означената линија во кодот со TODO: делот.
  + За најава, потребно е да се користат inMemory корисници со следните карактеристики: {username: “user”, password: “user”, role: “ROLE\_USER”} и {username: “admin”, password: “admin”, role: “ROLE\_ADMIN”}.