

ОЛАКШАНИ ПРОЈЕКАТ

Продавница кућних љубимаца

Увод

Потребно је креирати, имплементирати и тестирати сервис који симулара рад продавнице кућних љубицама. Продавница кућних љубимаца ради у две смене и може вршити продају или набавку кућног љубимца.

Ентитети

Како би информациони систем неометано функционисао систем је заштићен **аутентификацијом** пре него се дозволи било која друга акција у систему, а ради додатног нивоа заштите корисници се деле на два типа (улоге): **продавац** и **менаџер**.

Сви ентитети у систему имају обавезан атрибут који их једнозначно идентификује (*ID* или *GUID*).

Корисник се моделује следећим атрибутима: **корисничко име**, **лозинка**, **име и презиме** и **улога (енумерација)**. Корисници су ентитети у систему који се пријављују и изваравају одређене акције које су им дозвољене на основу улоге.

Кућни љубимац се моделује следећим атрибутима: **латински назив**, **име**, **тип** (сисар, гмизавац, глодар), **продајна цена** и **продат** (*boolean*). Максималан број кућних љубимаца који могу бити у продавници је **10**.

Фискални рачун се моделује следећим атрибутима: **име продавца**, **датум и време продаје**, **укупан износ**. Фискални рачуни се издају након успешне продаје кућног љубимца.

Функционалне целине информационог система

Сервис је потребно заштити **аутентификационим механизмом**. **Аутентификација** на сервис се врши уносом **корисничког имена** и **лозинке**, након чега се на основу **типа** (улоге) пријављеног корисника пружају одговарајуће функционалности.

Менаџер након пријаве може да:

- **Дода новог кућног љубимца** (у продавници не може бити више од 10 љубимаца),
- **Прегледа све списак свих кућних љубимаца** (и продат, и у радњи),
- **Прегледа све издате фискалне рачуне**.

Продавац након пријаве може да:

- **Прегледа списак кућних љубимаца који нису продати**,
- **Креира нову продају кућног љубимца и изда фискални рачун**.

Продавница врши продају на основу смене. Прва смена је од **8:00** до **16:00**, а друга смена **16:00** до **22:00**. На основу смене, у сервис се **инјектује другачија имплементација** сервиса продаје.

Дневни сервис продаје када рачуна коначни износ, на укупан износ примењује попуст од **15%** и тај износ примењује као укупан износ на фискалном рачуну.

Ноћни сервис продаје када рачуна коначни износ, на укупан износ примењује додатну вредност пореза, односно на укупан износ дода **10%** и тај износ примењује као укупан износ на фискалном рачуну.

Подацима је потребно приступати помоћу **репозиторијума**, док се подаци чувају у **JSON бази података**. Такође, **сваку активност у сервисима бележити помоћу инјектованог сервиса за евиденцију догађаја**. Сервис треба да подржава **Dependency Injection** односно да за један коресподентни интерфејс може имати више различитих имплементација које се инјектују у друге сервисе.

За чување догађаја користити имплементацију која догађаје уписује у **текстуалну** датотеку са информацијама о **типу евиденције, датуму и времену** и информацији о **евидентираним догађајима**. Тип евиденције може бити (**INFO, ERROR, WARNING**).

Интеракција са корисником

Сервиси искључиво садрже пословну логику и никако не садрже интеракцију са корисником. Интеракција са корисником остварује се посредством **слоја презентације**, где су тематски имплементирани менији и у сваком менију позван одговарајући инјектовани сервис/и.

Тестни подаци

Пројектно решење приликом покретања треба да има већ спремне иницијалне податке за све ентитете који су моделовани.