Rytis Čepulis 3kr. 1gr. 1pgr.

PSP LW2 in progress

Regitra

Displaying page to order driver’s license

Communication between layers to show page for ordering driver’s license is displayed in a sequence diagram (Figure 4.)

Graphical user interface

Description automatically generated

Figure 1. Get page for ordering driver’s license (Sequence diagram)

To display page for ordering a driver’s license UI is going to need some data that describes the main information about the driver’s license that can be ordered, that data is in PersonEntity object which is personal information and categories of vehicles that the user will be allowed to drive (Figure 1.)

Text

Description automatically generated with low confidence

Figure 2. PersonEntity

To authenticate the user and get their personal information UI is going to redirect user to an external identity authenticator which should provide users personal information in PersonDto object (Figure 2.)

Graphical user interface, text, application

Description automatically generated

Figure 3. PersonDto

After the user is authenticated and personal information PersonDto is received UI is going to call GET endpoint in UserController for getting driver’s license order and pass PersonDto as request body. The endpoint is going to return PersonEntity. UserController should stay lean, therefore it is going to get PersonEntity from BusinessLogic layer.

The BusinessLogic layer is calling DataAccess layer which gets and returns LicenseCategory object (Figure 3.) from the database.

Graphical user interface, text, application

Description automatically generated

Figure . LicenseCategories

Having PersonDto and LicenseCategories now BusinessLogic can map those two into UserEntity class and return it to the UI.

Posting order for driver’s license

Communication between layers to post order for driver’s license is displayed in a sequence diagram (Figure 5.)

Diagram

Description automatically generated with medium confidence

Figure . Post order for driver’s license (Sequence diagram)

When the user wants to submit and order the UI is going to call POST method in UserController’s endpoint for getting driver’s license order and pass PersonDto as a request body. The endpoint is going to return OrderEntity (Figure 6.) and status code.

Text

Description automatically generated with low confidence

Figure . OrderEntity

If the status code is correct the UI is going to use OrderEntity to display information that the driver license will contain. UserController should stay lean, therefore it is going to BusinessLayer to get OrderEntity and add order to the database.

The BusinessLogic layer is going to map PersonDto and RegitraInformation (Figure 7.) into an Order class which is going to be the same as OrderEntity and pass it to DataAccess layer which is going to Insert the new order into the database and returns Order object. Finally, the Order is mapped into OrderEntity and passed to UserController which returns it to UI.

Graphical user interface, text, application

Description automatically generated

Figure . RegitraInformation

NOTES

Prisiregistruoja

Uzsisako pazym

Clientas postina orderControlleriui

Controlleris kreipiasi I logikos layeri (paduoda personalIdentificationNumber) ar galima uzsakyti

Logikos layeris kreipiasi I duombazes gateway u

Duombazes gatewayus gauna visas LicenseCategories kurias turi Person

Jeigu is LicenseCatogories yra zmogaus norimas uzsakyti

True ->

False ->

Kokio dydzio sistemas, vartotj skaicius, kas tie vartotojai vartotojai

diagrama kaip isskaidom servisus, komponenetus, modulius

galima pasirinkti viena dali kad per daug neissiplesti (pvz automobl. registracija, vair, pazm isdavimas)