The schedule page is a table with the days of the week and times, where the corresponding activity is indicated at the intersection. For unauthorized users to the right of the schedule, there is an opportunity to choose a form for registering a trial lesson in the studio. And for authorized users, instead, there is a button that is responsible for filtering the schedule to get an individual user's schedule, and below this button is information about the number of remaining classes on the active subscription.

The last page has all the information you need to find and contact the studio with ease. It contains the contacts and addresses of the studio, just below the exterior of the studio and its halls, and even below you can find a map to which the location of the studio is tied.

Chaslau Averchanka
(Fr. Skorina GSU, Gomel)
Scientific advisor **Viktar Liauchuk**, Ph.D. in technics, associate professor

TESTING THE WEB APPLICATION FOR A WORKSHOP

There are scenarios of user interaction with the server in this project: from authorization to processing requests. Several scenarios are considered here. Before writing tests, you need to prepare a data area that interacts in some way in tests. For this a backup database was created, and several SQL-scripts were written. They are shown in figures 1 and 2.

```
delete from user_role;
delete from users;

insert into users(id, active, password, username) values
(1, true, '$2a$08$XUSWXBtHO7N4SLkv6VlXWeVguDlQryMS.u9igSmkc30eZyUYYJ25a', 'Cheslav'),
(2, true, '$2a$08$XUSWXBtHO7N4SLkv6VlXWeVguDlQryMS.u9igSmkc30eZyUYYJ25a', 'test');

insert into user_role(user_id, roles) values
(1, 'USER'), (1,'ADMIN'),
(2,'USER')
```

Figure 1 – SQL- script that updates user data

```
@Test
@Sql(value = {"/create-user-before.sql", "/forms-list-before.sql"}, executionPhase = Sql.ExecutionPhase.BEFORE_TEST_METHOD)
@Sql(value = {"/forms-list-after.sql", "/create-user-after.sql"}, executionPhase = Sql.ExecutionPhase.AFTER_TEST_METHOD)
public void correctLoginTest() throws Exception{
    this.mockMvc.perform(formLogin().user("Cheslav").password("test"))
            .andDo(print())
            .andExpect(status().is3xxRedirection())
            .andExpect(redirectedUrl( expectedUrl: "/"));
}
```

Figure 2 – Test that checks user data for validity for authorization

Doing like this it is possible to cover the whole project with automated tests. A developer can see total results in friendly form (figure 3).

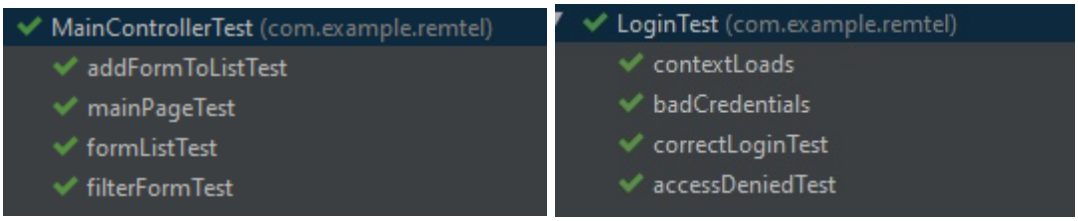

Figure 3 – Test results for interacting with the database

Chaslau Averchanka
(Fr. Skorina GSU, Gomel)
Scientific advisor **Viktar Liauchuk**, Ph.D. in technics, associate professor

AUTOMATION OF ORDERS ACCOUNTING FOR A WORKSHOP

At the moment most of the technical maintenance workshops have their own web services to automate customer interactions. Some of them are discussed in the presentation.

The reason for creating a web service is that this workshop does not have a website as such. Its purpose is to allow dividing and automating the work of the workshop.

There are two roles in the developed application: user and administrator. Each actor has a specific list of available use cases.

Relational database consists of about 4 tables. MySQL is used for storing information.

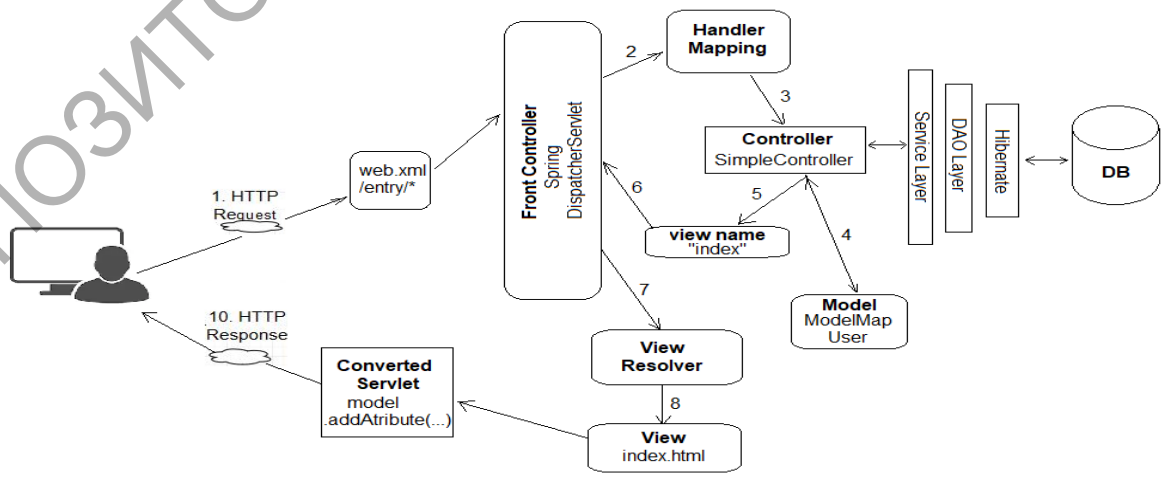

Figure 1 – Spring MVC architectural pattern