

OADTurk

GROUP 6

ALEKSANDAR STOJCIC

STEFAN GAJANOVIC

AMIR MUJACIC

ALEKSA PANDUREVIC

DJORDJE RAJIC

NIKOLA BALAC

Contents

Inception phase	2
Task 1.	2
Task 2	3
OADTurk Introduction	3
Project utilization.....	3
Target audience	3
The most important OADTurk Use Cases (with priority).....	3
Description of Use Cases	4
Use Case Diagram	8
UML analysis class diagram	8
Project plan.....	9
Screenshots.....	10
Resource estimation	15
Risk factors.....	15
Elaboration phase	16
Task 1	16
Task 2	18
Screenshots and added features	21
Use cases implemented	21
Project plan.....	30
Task 3	30
Our team	30

Inception phase

Task 1.

Our first task was to create a UML Diagram in Visual Paradigm, from the text we were provided.

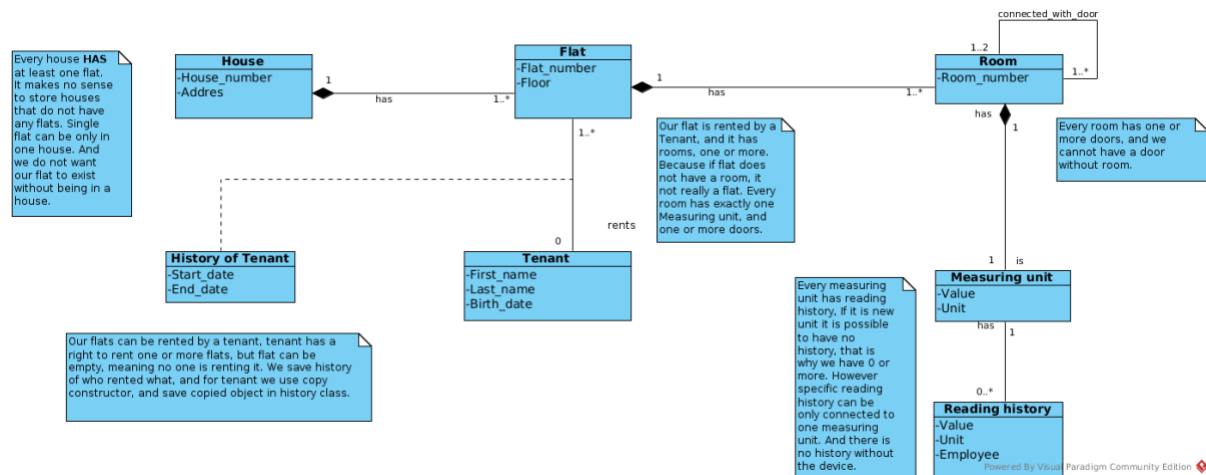


Figure 1. UML Diagram

In our case UML Diagram shows, that every house has at least one flat. We do not have need to store houses that do not have any flats. We also do not want flat to exist without house. These flats can be rented by tenants, however we do save some basic data about tenant and history about which flat was rented when. Flat has one or more rooms, we chose one or more, because if we do not have any rooms that is not really a flat. Each room consists of one measuring unit, and we keep history of every measuring unit. Every room has one or more doors, and we cannot have a door without room. However because it is required to know which room is connected to which we have a `connected_rooms()` method in door class, so we can get this information.

Task 2

OADTurk Introduction

OADTurk is a learning environment based on human computation, that can be used by users. This software is central environment for the administration of learning applications(LAs). Learning applications are central components of OADTurk. Our goal is to make increasing number of learning applications. The two main parts of OADTurk are: “OADTurk User Community” and “OADTurk Environment”.

Project utilization

The customers can use OADTurk for multiple reasons. One of the most important is participating in different learning applications. Every user can choose different learning units. For every different LAs user has possibility to participate in different type of exams. There is also possibility for every user to apply for “creator”.

Target audience

The target audience for our software is actually every person who wants to acquire new knowledge. OADTurk is also very helpful for students during their studies. Human computation offers every person to learn their learning units on efficient way. To narrow out our target audience a little bit, we could say that anyone with a desire to learn new things, and help other do the same would be interested in a project like this. OADTurk give you the opportunity to takes exams, and learn from Learning Applications created by other people, but one can also make their own applications in learning unit, with time, one can become creator.

The most important OADTurk Use Cases (with priority)

Use Cases:	Priority
Registration	High
Sign In	High
Sign Out	High
Applying for Creator	Average

Participating in exams	Average
Users Learning Applications	High
Adding Creators	Low
Managing user's requests	Average
Managing user's account	Average
Creating exams	Average
Creating Learning Application	Low
Changing personal info	High
Defining categories	Low
Creating Learning Units	High
Consuming Learning Units	High
Exploring exam results	High

Description of Use Cases

Use Case: Registration	Priority: High
<p><i>Description</i></p> <p>Performing registration for OADTurk application.</p>	

<p><i>Scenario 1</i></p> <p>A user enters their registration credentials. Then presses button “Register”. Dialog “The authentication E-Mail has been sent to your address. Please verify it and try Signing in” is shown.</p>
<p><i>Scenario 2</i></p> <p>A user enters their registration credentials. Then presses button “Sign in”. He will be then redirected to the welcome tab.</p>

Use Case: Log In	Priority: High
<p><i>Description</i></p> <p>Log in into OADTurk application.</p>	
<p><i>Scenario 1</i></p> <p>A user enters their credentials. Then presses button “Sign in”. Upon successful authentication the user shall be redirected to the OADTurk tab.</p>	
<p><i>Scenario 2</i></p> <p>A user enters their wrong credentials. Then presses button “Sign in”. Because of the wrong authentication the user can not be redirected to the OADTurk tab.</p>	

Use Case: Log Out	Priority: High
<p><i>Description</i></p> <p>Sign out from OADTurk application.</p>	

<p><i>Scenario 1</i></p> <p>A user presses button “log out” and will be redirected to welcome tab.</p>
--

Use Case: Users Learning Applications	Priority: High
<p><i>Description</i></p> <p>A user is enabled to switch over different Learning Applications.</p>	
<p><i>Scenario 1</i></p> <p>A user can choose different Learning Application. For every Learning Application users got different questions to solve. After choosing appropriate answer, the user will be click on the button “Submit”.</p>	

Use Case: Changing personal info	Priority: High
<p><i>Description</i></p> <p>Changing personal info of the user.</p>	
<p><i>Scenario 1</i></p> <p>A user enters their registration credentials additionally with new password. Then presses button “Save”. If the old password was correct, new password will be successfully created.</p>	
<p><i>Scenario 2</i></p> <p>A user enters their registration credentials with wrong old password. Then presses button “Save”. Because of the wrong old password, the user can not be enabled to change password.</p>	

Use Case: Applying for Creator	Priority: Average
<p><i>Description</i></p> <p>Every user is enabled to apply for creator.</p>	
<p><i>Scenario 1</i></p> <p>A user is still creator and is enabled to create new Learning Applications, which must be confirmed by admin.</p>	
<p><i>Scenario 2</i></p> <p>A user applies for creator, and wait for admins to evaluate his request.</p>	

Use Case: Participating in exams	Priority: Average
<p><i>Description</i></p> <p>Every user has a possibility to participate in exams.</p>	
<p><i>Scenario 1</i></p> <p>A user wants to participate in an exam, but he is unregistered. By a click on a button “Register”, he can be registered and then to participate in an exam.</p>	
<p><i>Scenario 2</i></p> <p>A user is registered and by a click on a button “Start”, he will be redirected to new tab with questions.</p>	

Use Case: Consuming Learning Units	Priority: High
<p><i>Description</i></p> <p>A user is enabled to create new Learning Units.</p>	
<p><i>Scenario 1</i></p> <p>A user is enabled to ask on different Learning Units. He can check his answers on button “Verify”, or change the Learning Unit with button “Next”.</p>	
<p><i>Scenario 2</i></p> <p>A user is also enabled to add new Learning Units. If he presses the button “Add Lu”, new window will be shown. In this window user is enabled to choose appropriate properties for Learning Unit, which he wants to add.</p>	

Use Case Diagram

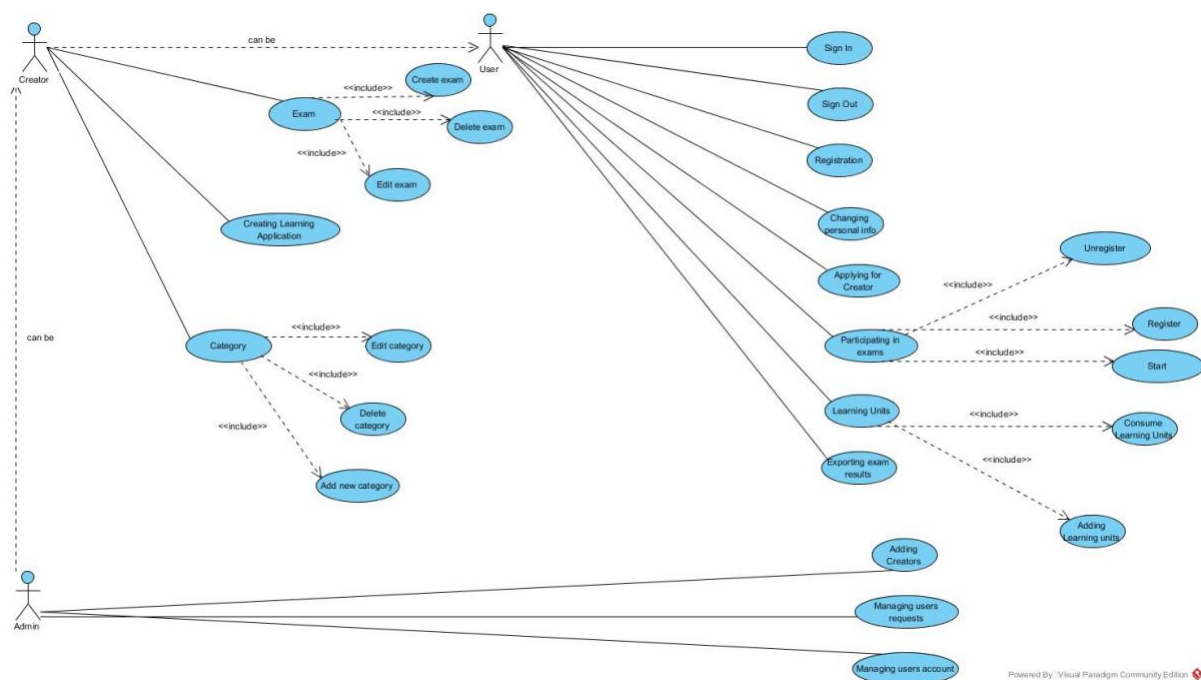


Figure 2. Use Case Diagram in Visual Paradigm

UML analysis class diagram

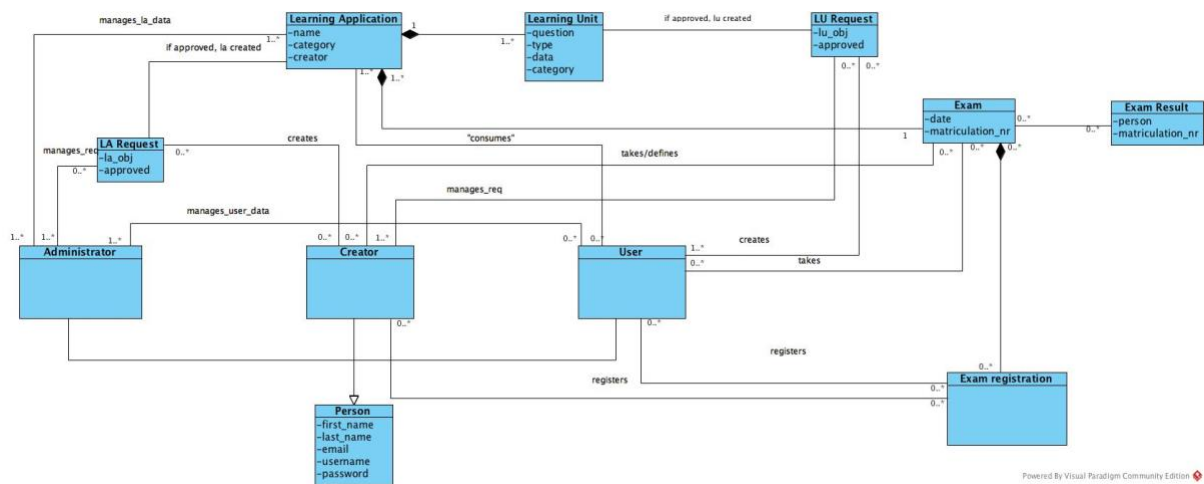
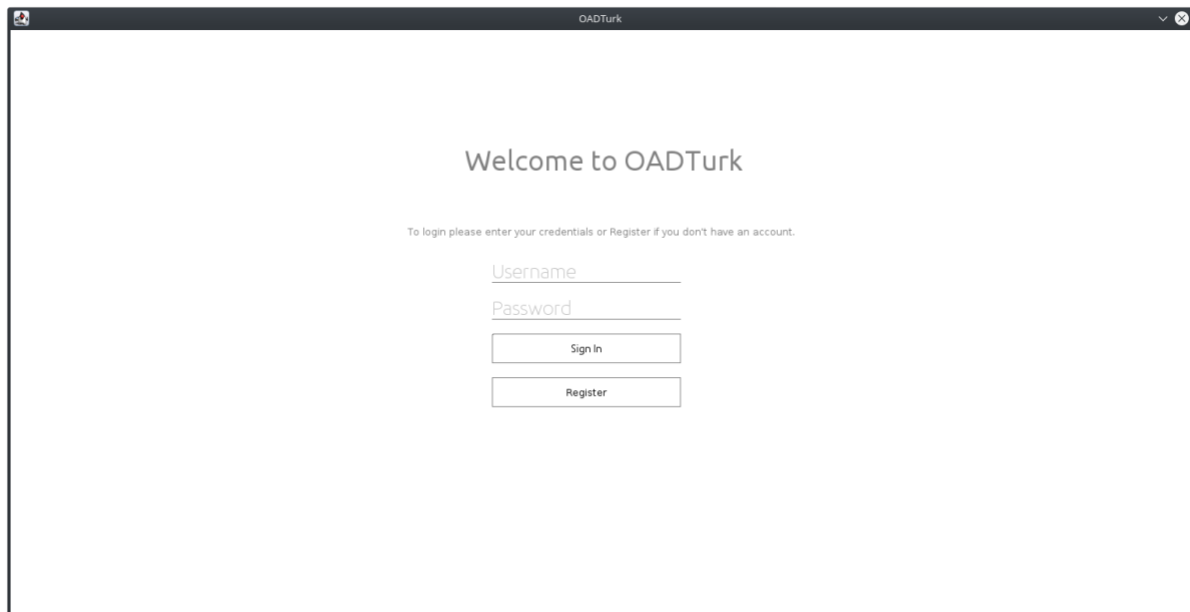


Figure 3. UML Analysis class diagram in Visual Paradigm

Project plan

What	Who
Project Plan	Aleksandar
Infrastructure	Stefan, Amir
Houses Model	Djordje, Aleksa
Testing	Nikola
Submission document	Aleksandar
GUI	Stefan, Amir
Use Case Diagram	Aleksandar
UML analysis class diagram	Djordje, Aleksa
Screenshots	Nikola

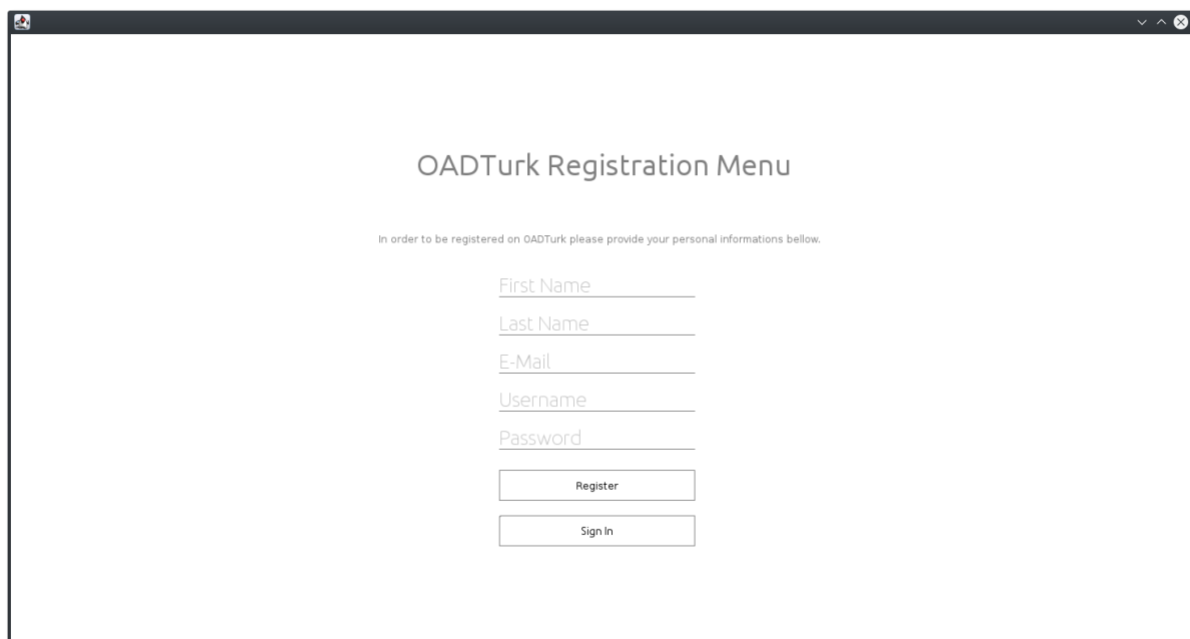
Screenshots



A screenshot of a web browser window titled "OADTurk". The page has a white background and a dark grey header bar. The main content area is centered and contains the following elements: a heading "Welcome to OADTurk", a line of text "To login please enter your credentials or Register if you don't have an account.", two input fields labeled "Username" and "Password", a "Sign In" button, and a "Register" button.

Figure 4. Login page

In the screenshot (Figure 4.) you can see login page for our system. Users cannot consume Learning Applications, or Learning Units before having an account. In this page user is asked to enter login data, username and password. This same page is used for administration login and creators login, based on username which has to be unique system knows which permission logged in user has.



A screenshot of a web browser window titled "OADTurk". The page has a white background and a dark grey header bar. The main content area is centered and contains the following elements: a heading "OADTurk Registration Menu", a line of text "In order to be registered on OADTurk please provide your personal informations below.", five input fields labeled "First Name", "Last Name", "E-Mail", "Username", and "Password", a "Register" button, and a "Sign In" button.

Figure 5. Registration page

For users to be able to obtain login data, registration is needed. Figure 5. shows how registration page looks. Anyone wanting to register need to enter same basic data. First name,

last name, email, username and password. Username is unique for every user, so we can differentiate between permissions, in order to know who has the ability to delete something, or change something.

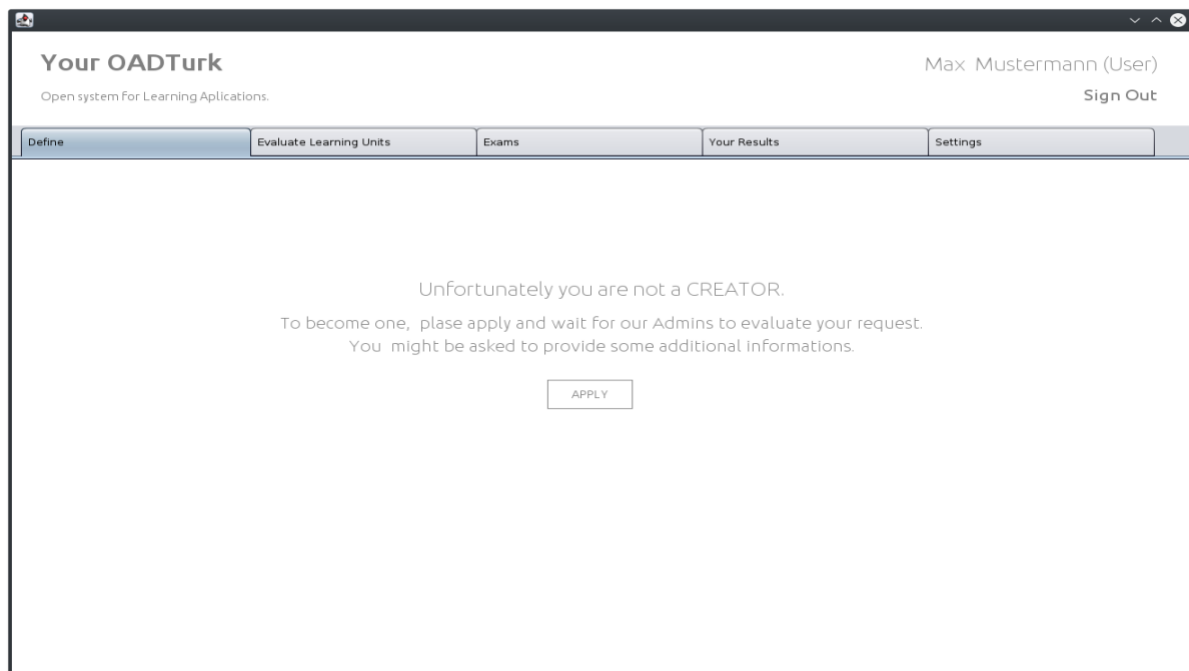


Figure 6. Define page

In example above(Figure 6.) we can see our define page, here creators can create Learning Units, and make requests when creating new Learning Applications. In the future during the implementation this page will not be visible for regular users, however they will have the

possibility to make requests when creating new Learning Units.

The screenshot shows a web application window titled "Your OADTurk" with the subtitle "Open system for Learning Applications." The user is logged in as "Max Mustermann (User)" and can click "Sign Out". The navigation bar includes tabs for "Define", "Evaluate Learning Units" (which is active), "Exams", "Your Results", and "Settings". On the left, a sidebar lists "OAD" and "SWP". The main content area displays a question: "Question question question question question question question question question question question question?". Below the question are four answer boxes, each containing a checkbox and the text "Answer answer answer answer answer". A "Submit" button is positioned at the bottom right of the main area.

Figure 7. Evaluate learning units tab

Here we can see evaluation look (Figure 7.), on the left hand side we can see that we can choose between different Learning Applications, and for that Learning Application we are presented with its learning unit (question) after answering the question we press submit button to proceed. If we want we can press second button that says create LU, we can create new Learning Unit, and send request for creator to approve.

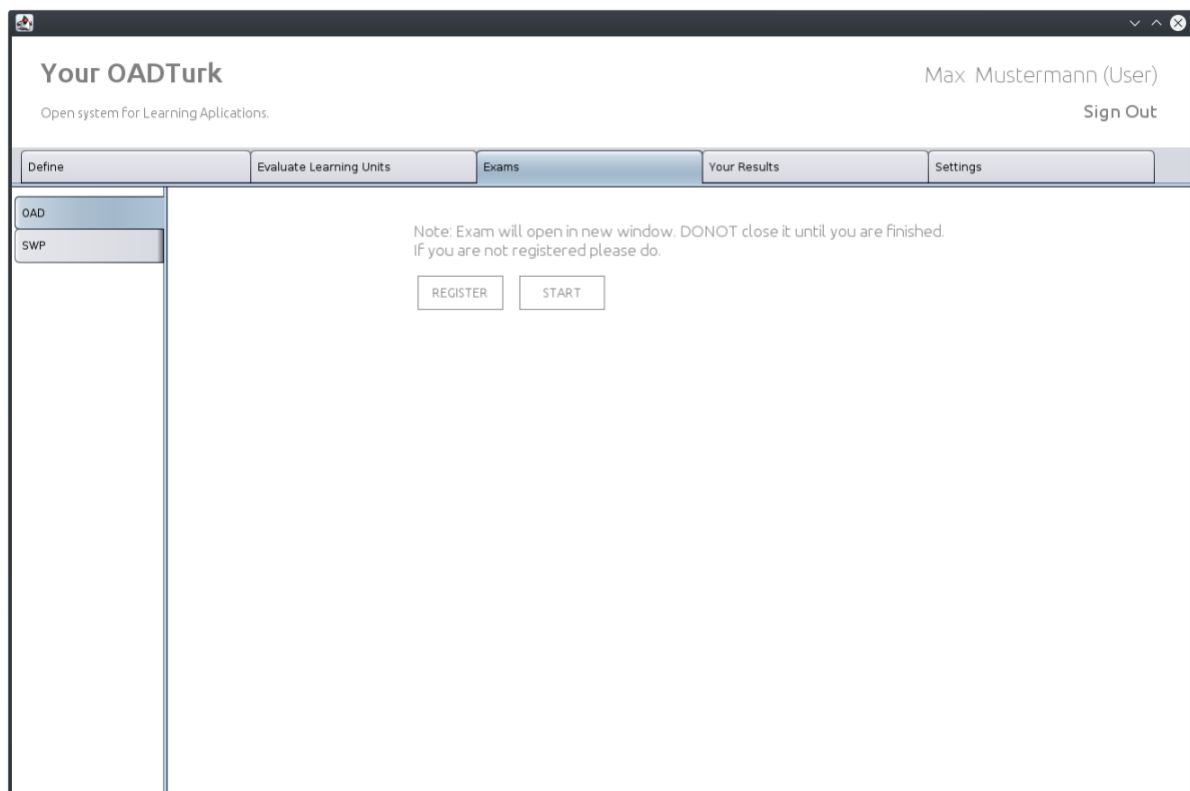


Figure 8. Exams page

Exams page (Figure 8.) allows us to take exams, so we can verify our knowledge. Our exam results are displayed in the results page (Figure 9.).

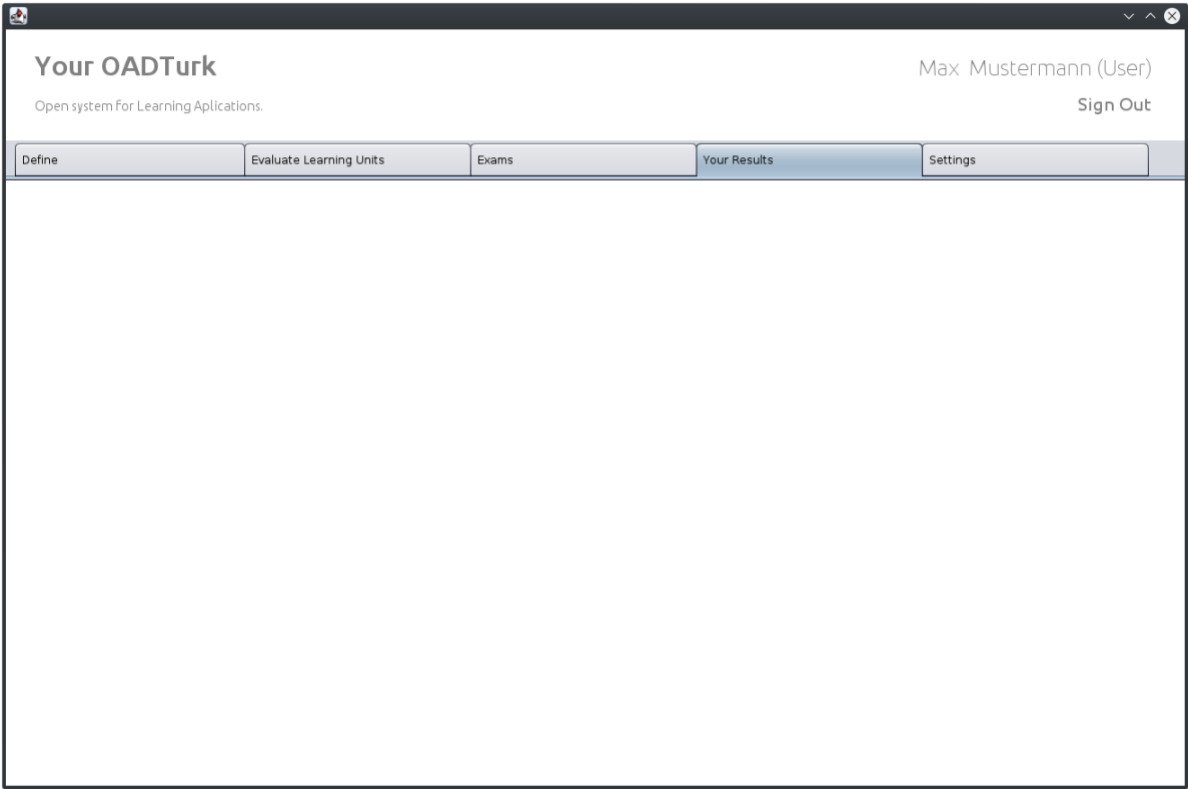


Figure 9. Exam result page

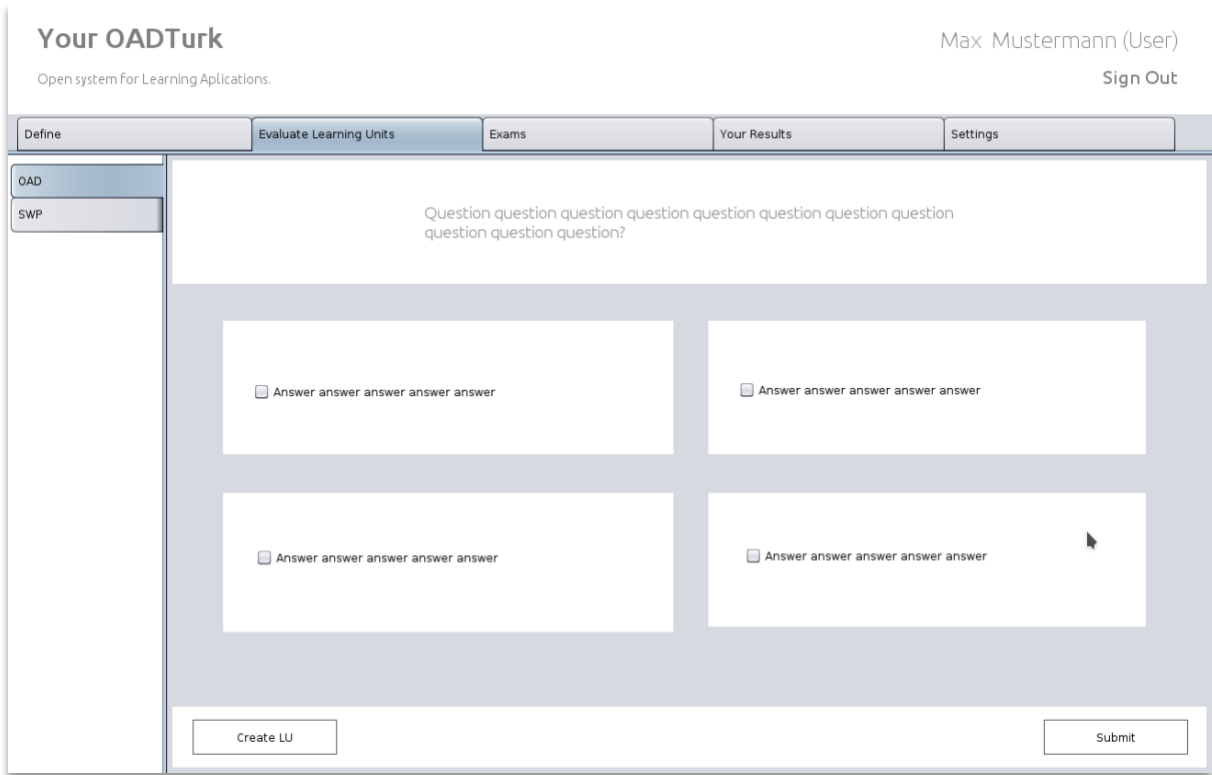


Figure 10. Single exam example

Figure 11. Settings page

In Settings page user is able to change its personal info, such as First Name, Last Name, username, email, and change password. Admin has the ability to change this data too.

Resource estimation

Iteration	Start	End/Delivery
Inception	10.10.2017	7.11.2017
Review	7.11.2017	21.11.2017
Elaboration	7.11.2017	5.12.2017
Review	5.12.2017	12.12.2017
Construction	5.1.2017	23.1.2018
Review	23.1.2017	30.1.2018

Risk factors

We have identified following risk factors that could affect our project:

- Activities are missing from scope

- Estimates are inaccurate
- Scope is not well defined
- Team conflict over proposed changes
- Under communication
- Low team motivation
- Project team lack authority to complete work
- Requirements are incomplete
- Decision delays impact project
- Lack of management or control

Elaboration phase

Task 1

Ovdje die prvi zadatak iz ove zadace. MVC I ti dijagrami.

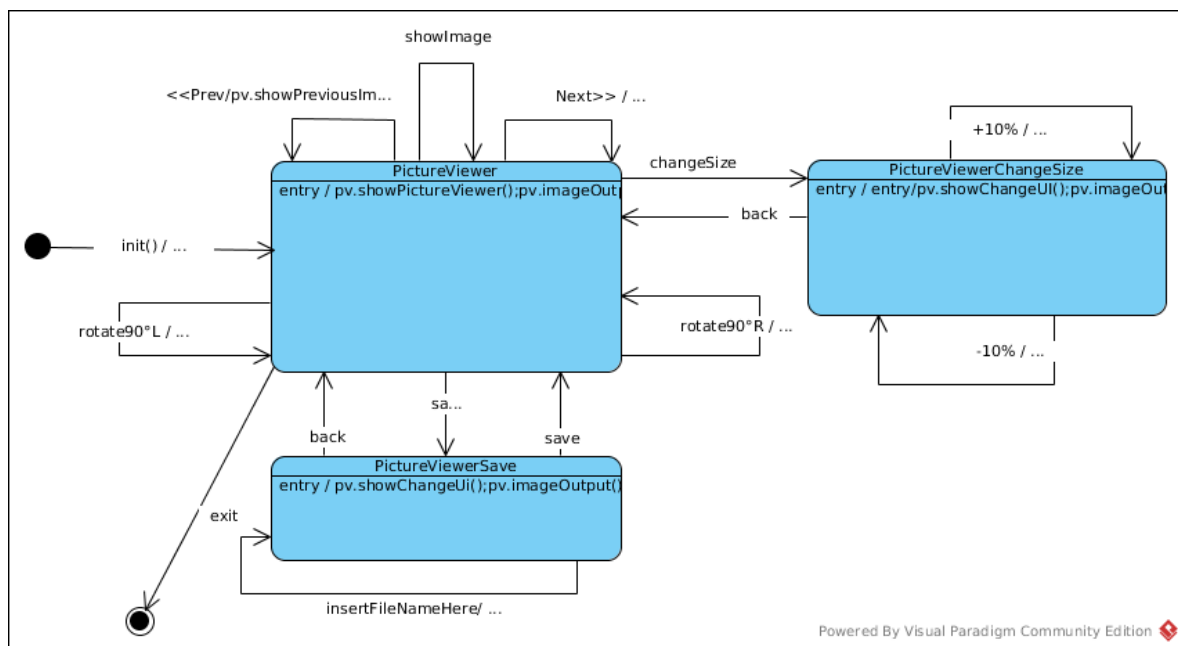


Figure 12. Picture viewer state chart



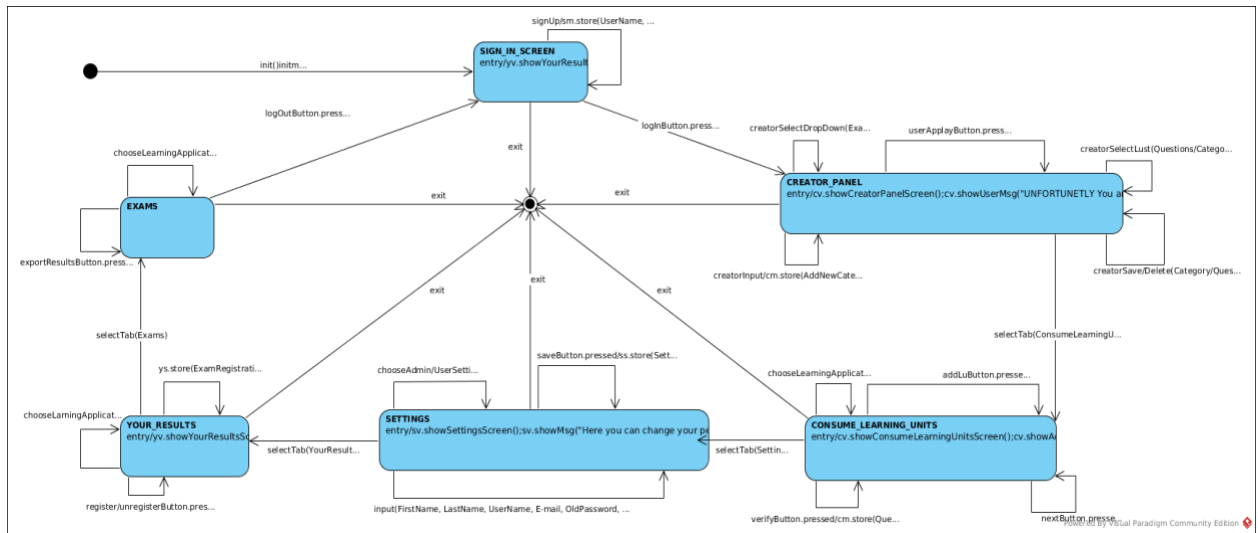


Figure 16. State chart diagram

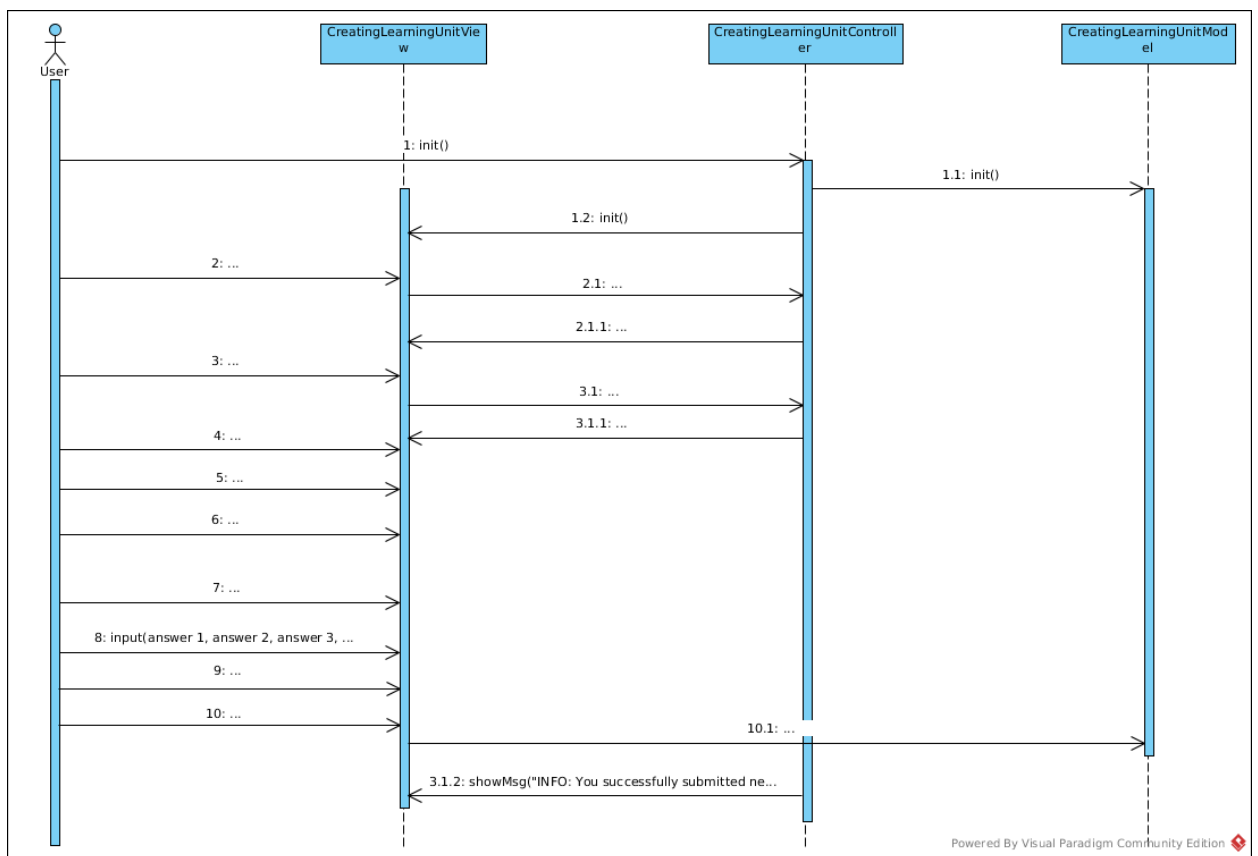


Figure 17. Sequence diagram for Learning Unit

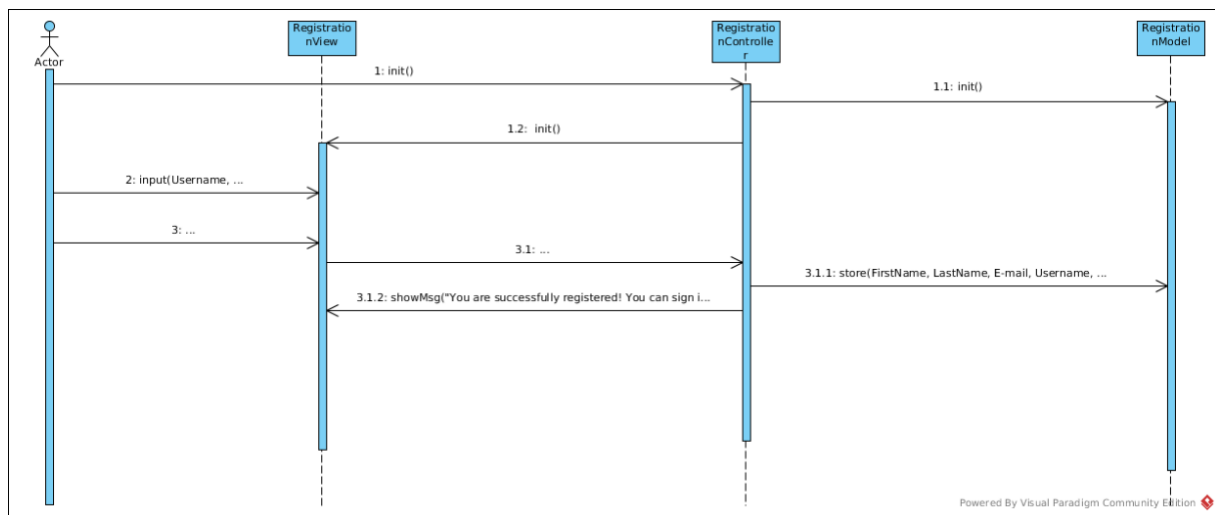


Figure 18. Sequence diagram for registration

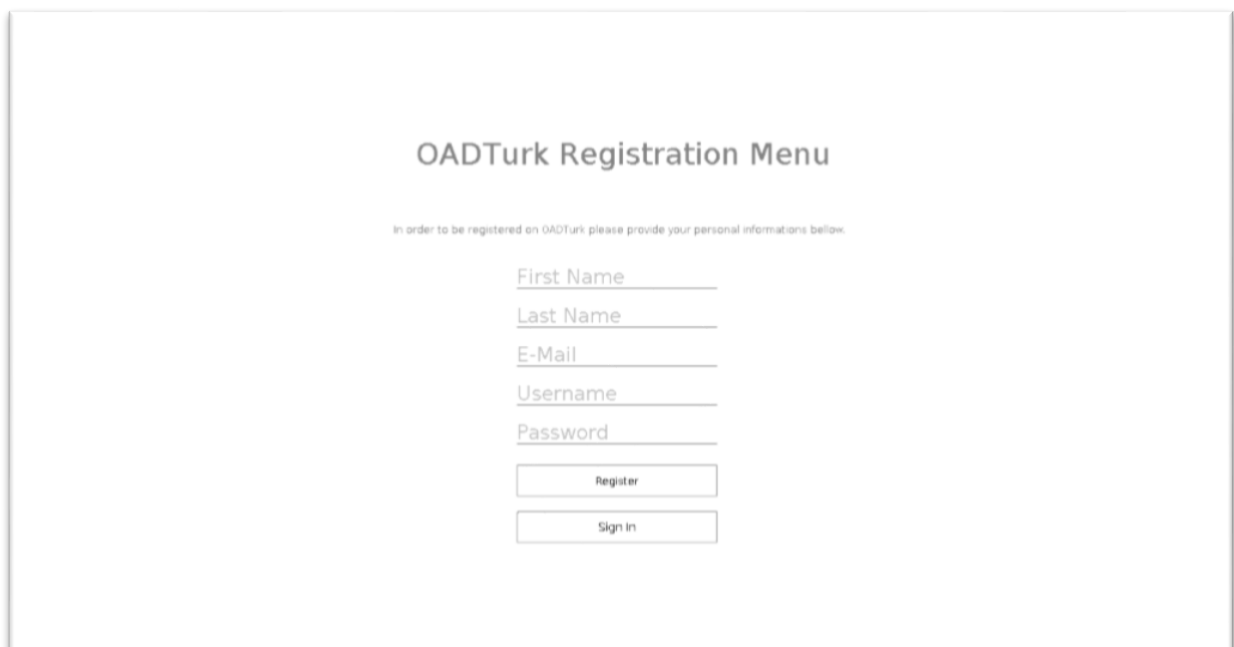
Screenshots and added features

In this phase we've added a lot of new functionality and 80% or more of our use cases are fulfilled. Great majority of buttons and tabs works, and there are also some hidden features that will be demonstrated. In the following part of the document screenshots and description of what has been done can be found.

Use cases implemented

In the following part you can see the screenshots of the use cases that we have implemented, and brief explanation of each.

Use case: Registration



The screenshot displays a web form titled "OADTurk Registration Menu". Below the title, a small instruction reads: "In order to be registered on OADTurk please provide your personal informations below." The form contains six input fields stacked vertically: "First Name", "Last Name", "E-Mail", "Username", and "Password". Each field is a simple text box with a light gray border. Below these fields are two buttons: "Register" and "Sign in", each enclosed in a rectangular box with a light gray border. The entire form is centered on a white background.

Figure 19. Registration page

OADTurk Registration Menu

You are successfully registered! You can sign in now!

Djordje

Rajic

djordje.rajjc@student.tugraz.at

DjordjeR

raja

Register

Sign In

Figure 20. Successful registration

OADTurk Registration Menu

Email address is not valid!

First Name

Last Name

E-Mail

Username

Password

Register


Sign In

Figure 21. Error in registration

In the examples above you can see our registration page (*Figure 19. Registration page, Figure 20. Successful registration, Figure 21. Error in registration*). As you can see our program reacts differently to the user input. In case user has made no mistake and entered all data correctly, user is provided with feedback in a form of green text that inform the user that registration was successful and user is now able to sign in without any problems. In the other case if user

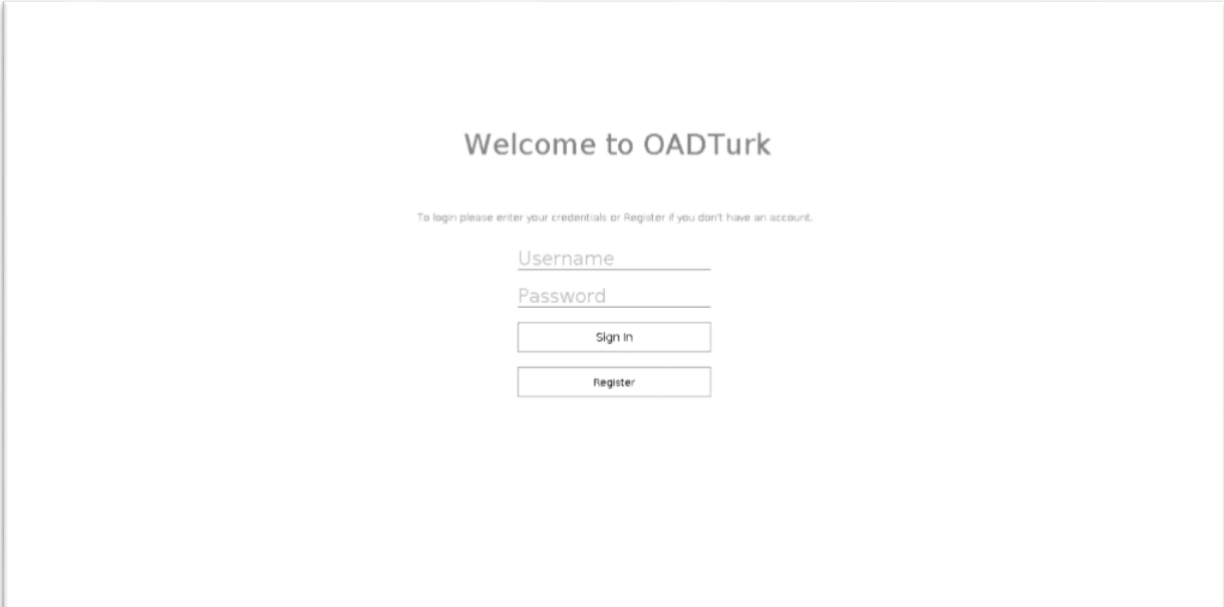
enters no data, or some data that is invalid, same kind of feedback is provided, however in this case our text changes and it is in red colour so the user knows mistake has been made, or we are unable to process users request at the time. Registration is required step, for our application to function as well as possible, and it was high priority to get this part of the program running. After implementing this part of the program next step was to implement sign in panel, so access to our application is indeed limited only to users that have registered.

Use case: Sign in



The screenshot shows a web page titled "Welcome to OADTurk". Below the title, there is a red error message: "ERROR: Invalid Username or Password!". Underneath the error message, there are two input fields labeled "Username" and "Password". Below these fields are two buttons: "Sign In" and "Register".

Figure 23. Sign in with error



The screenshot shows a web page titled "Welcome to OADTurk". Below the title, there is a line of text: "To login please enter your credentials or Register if you don't have an account.". Underneath this text, there are two input fields labeled "Username" and "Password". Below these fields are two buttons: "Sign In" and "Register".

Figure 22. Sign in

Above (Figure 16. Sign in with error, Figure 15. Sign in) we can see our login page, not much has changed since the last time in terms of how the page looks. In this case only the functionality was added now we can login with different data, while creating session. This way only registered users can access this system. If the login info is incorrect user gets the message with the information of what happened. The same login page is used for users, creators and admins. Our system can distinguish between different users, based on their login we show different pages once they have the access to the system. User, creator and admin have very different pages, and different access to certain areas. At every page within our document there is ability in upper right corner to sign out if you do not wish to use system anymore. This makes it so no one else can use your account when you are away.

Use case: Applying for creator

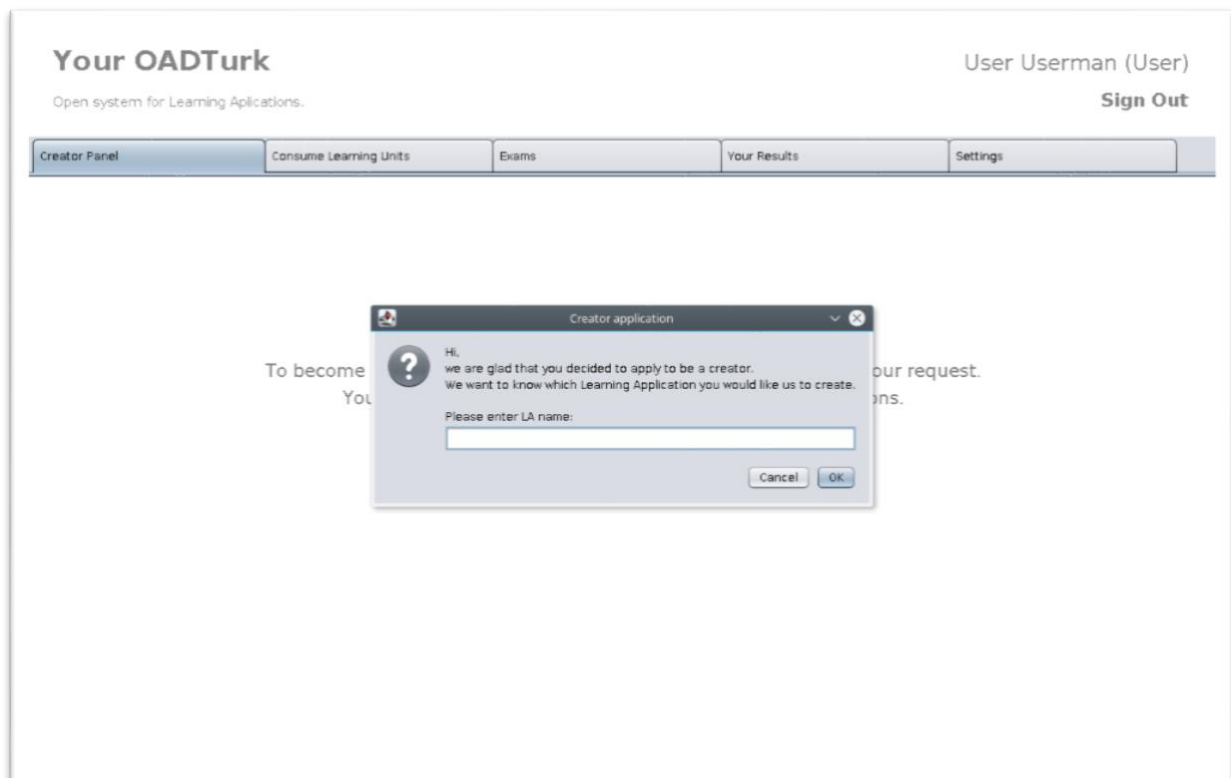


Figure 24. Applying for creator

Ability for users to apply to be creators was also added. Every user can now simply click button and apply to be creators. This functionality also extends to the creator panel where creator can approve or disapprove certain applications from users.

Your OADTurk

Open system for Learning Applications.

Creator Creatori (Creator)

[Sign Out](#)

Creator Panel

Consume Learning Units

Exams

Your Results

Settings

Learning Units:

ID	Question	Created by	Approved	Category	Save	Delete
0	Which of the following instantiations corresponds to the UML model?	creator	1	Category 1	Save	Delete
1	Which of the following texts describes the UML model?		1	Category 1	Save	Delete
2	Which of the following UML models reflects the domain description?		1	Category 1	Save	Delete
3	What is a „conceptual class diagram“?		1	Category 2	Save	Delete

Edit Categories:

ID	Category	Save	Delete
0	Category 1	Save	Delete
1	Category 2	Save	Delete

New Category:

[Add Category](#)

Exams:

1 - Exam2

[Edit](#)
[Delete](#)
[New Exam](#)

Message

You successfully saved this LUI

[OK](#)

Figure 25. Creator panel when viewed as creator

Use case: Participating in exams

Your OADTurk

Open system for Learning Applications.

User Userman (User)

[Sign Out](#)

Creator Panel

Consume Learning Units

Exams

Your Results

Settings

	ID	Exam	Date	Until	Questions	Points	Action
OAD	0	Exam1	25.11.2017 23:59:00	31.12.2017 23:59:00	4	4	Start
SWP	1	Exam2	25.12.2017 23:59:00	31.12.2017 23:59:00	4	4	Register

Figure 26. Exam tab

To take exam, User, Creator or Administrator first need to enter Exam tab as represented in Figure 15. There we have selection of exams which we can take, in case they are active, or for which we can register. When exam is active, on the left side of corresponding exam we will have Start button, when pressed will lead us to next figure which we will discuss in next paragraph. If exam is still not active (registration period), instead of Start button we will have Register button. When Register button is pressed, we get feedback in form of pop up window which ask us for confirmation. After successful registration, button label changes from Register to Unregister. When Unregister is pressed, similar procedure follows as for Register. We get pop up window which ask us for confirmation.

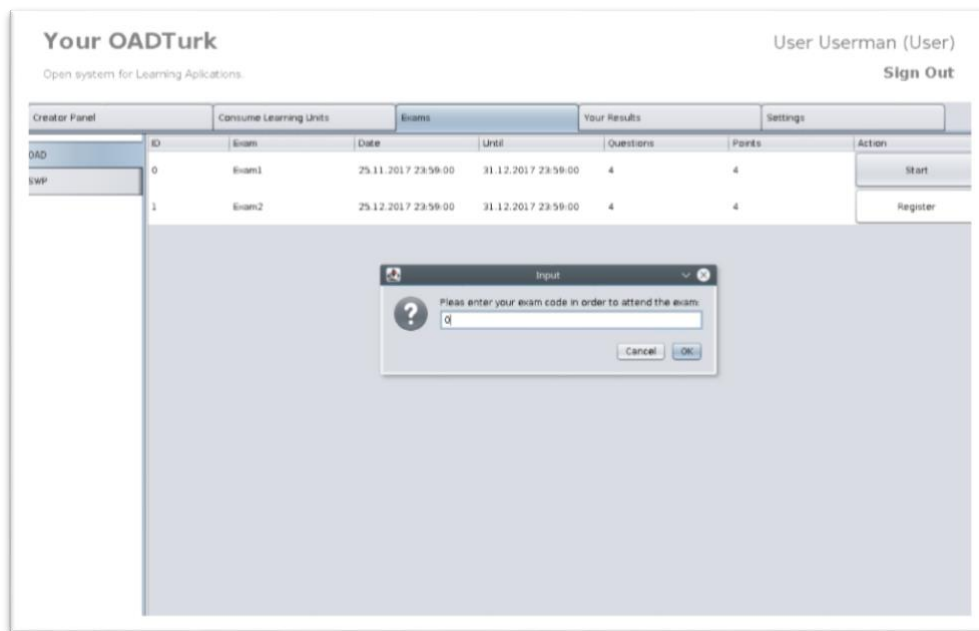


Figure 27. Starting Exam

When Start exam button is pressed, we get feedback as represented above in Figure 18. We are asked to enter exam code to precede to the exam. After entering exam code and pressing OK button in pop up window, window with our exam is opened as shown in figure below.

Which of the following instantiations corresponds to the UML model?

Employee

Project

ssnr
name

[1] — [1..2]
works for

pname

☒

ssnr=1_name=Huber — pname=p1
ssnr=2_name=Maier — pname=p2

☐

ssnr=1_name=Huber — pname=p1
ssnr=2_name=Maier — pname=p2

Figure 28. Exam sample

In this figure, we can see an example of one exam question. Design is simple and intuitive, on top of the window we have question and right beneath that we have offered answers which we pick with clicking the tick box next to it. When clicked, box changes state to ticked. We also have possibility to select multiple answers. After picking all answers, we precede by clicking Submit button which takes us to another question. When we get to the last question, instead of Submit button, we have button labelled as Finish Exam. After clicking that button, pop up window appears with points earned on exam.

Use case: Users Learning Applications

Your OADTurk

Open system for Learning Applications.

User Userman (User)

Sign Out

Creator Panel

Consume Learning Units

Exams

Your Results

Settings

OAD

SWP

Which of the following texts describes the UML model?

```

classDiagram
    class Employee {
        ssnr
        name
    }
    class Project {
        pname
    }
    Employee "1" -- "[1..2]" Project : works for
        
```

☐ Each project has at least two assigned employees.

☐ The maximum number of projects that can be assigned...

Add LU

Verify

Next

Figure 29. Sample of Learning Application

In figure 22. we can see implementation of Learning Applications and some changes in comparison to previous phase, which are: added Add LU and Verify button. Same as in previous iteration, we have on the left side of window list of all learning applications, which when selected, display learning units. Clicking Add LU button, user can apply for adding new LU. Next button leads us to the next question and Verify button gives us solution for given question, as we will see in following figure.

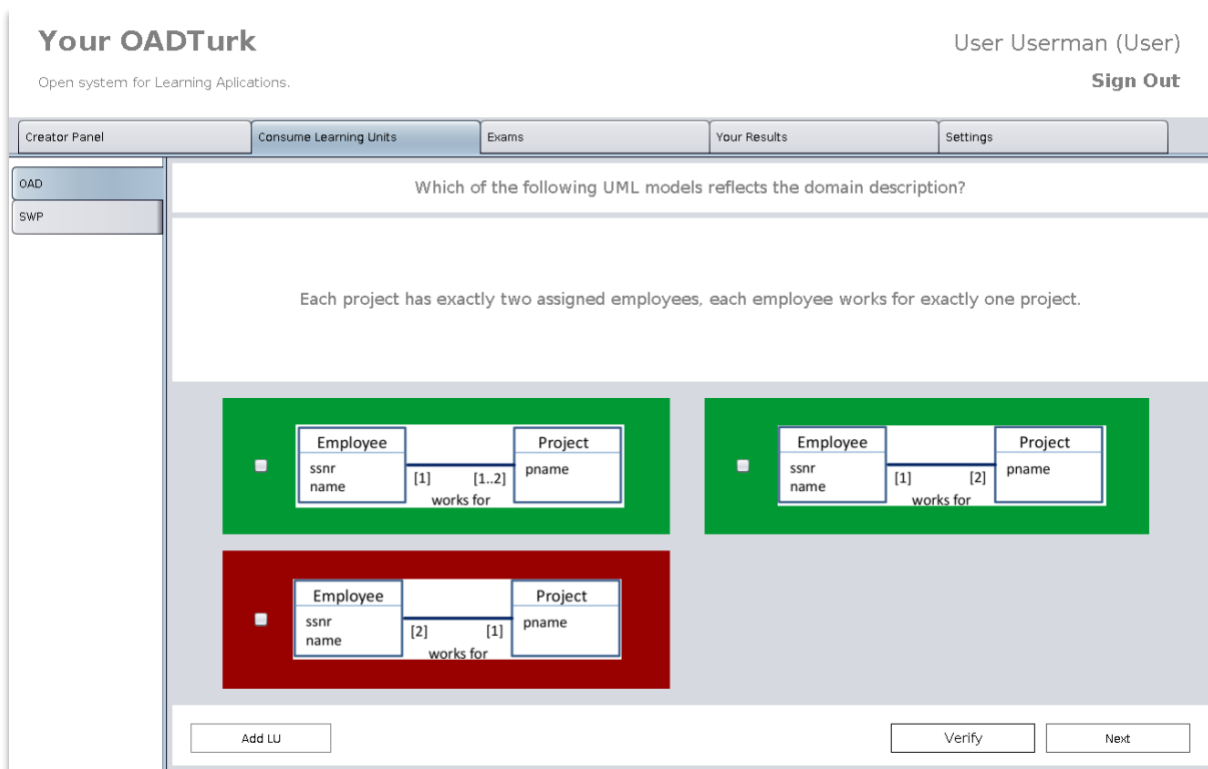


Figure 30. Verify button clicked

This is feedback that we get when clicking on Verify button, green marking correct and red marking false answers.

Use case: Managing user's requests

In Figure 18. we have representation of Creator panel. In central part of window, we can see all Learning Units, their creators, categories and if they are approved by creator or not. Also, we have buttons Save and Delete, with which we can either save changes or delete learning unit. With changing Approved state from 0 to 1, learning unit will be approved and therefore it will appear in Learning Application.

Use case: Changing personal info

Settings tab in which we can edit our personal information remained same as in previous (figure 11.), except now we have functionalities implemented. We can change first name, last name, username, email and password.

Use case: Consuming Learning Units

Figure 22. and 23. are one of the examples of consuming learning units. For each question, we have provided answers which we can select by ticking box next to each one. By clicking Next button, we are provided with new question and in case when we want to know solution of given question, clicking Verify we will get evaluation of our answer.

Project plan

For the continuation of our project we've left admin panel unfinished, and we need to do much more testing. We need to be sure everything is done according to specifications we decided that the admin panel was not the biggest priority right now, we have a working prototype that from users point of side, and creators does almost everything it should. Meaning users and creators are left with very little to be desired. As we go further with the development and testing, we are hoping to work out some minor bugs, finish the development of admin panel and complete the project according to the specifications. However, in this stage, there is a pretty good picture of how program functions and looks. Also, we've left connecting to the database for the end, because we feel connection to the database it pretty much done project, this way we have opportunity to add more things and options if needed. For now, our program works with arrays, that are used everywhere to populate data in our program. This means when we come to a phase where we are using database, not much has to be changed, we only need to populate the arrays from the database, and everything in our program will be up to date. Above in the text, there are many screenshots and explanations of exactly what and how it's been done, so we can be sure every requirement has been satisfied.

Task 3

Ovdje ide treci zadatak iz ove nove zadace.

Our team

Name	Responsibilities
Stefan Gajanovic	Development
Amir Mujacic	Development
Aleksa Pandurevic	Testing
Djordje Rajic	Usability

Nikola Balac	Analysis
Aleksandar Stojicic	Manager