<Hello World/>

Advanced Software Engineering – Project

Jessica Frabotta – 1758527

Alessio Ferrone – 1751859

1. Introduction

"Hello World" is a web application developed using WebRatio, which serves as a platform for users to learn programming languages through various courses. The application offers a comprehensive learning experience, providing a series of lessons accompanied by video tutorials and exercises to assess the user's understanding of the material.

Each course is taught and followed by a tutor to whom users can write to clarify any doubts about the lessons or exercises. The platform, to encourage collaboration among its users, includes forums where users can discuss and exchange ideas on how to solve the assigned exercises.

The system consists of 4 views: home, user, tutor, administration.

Let's start looking at the implementation details, starting from the Domain Model.
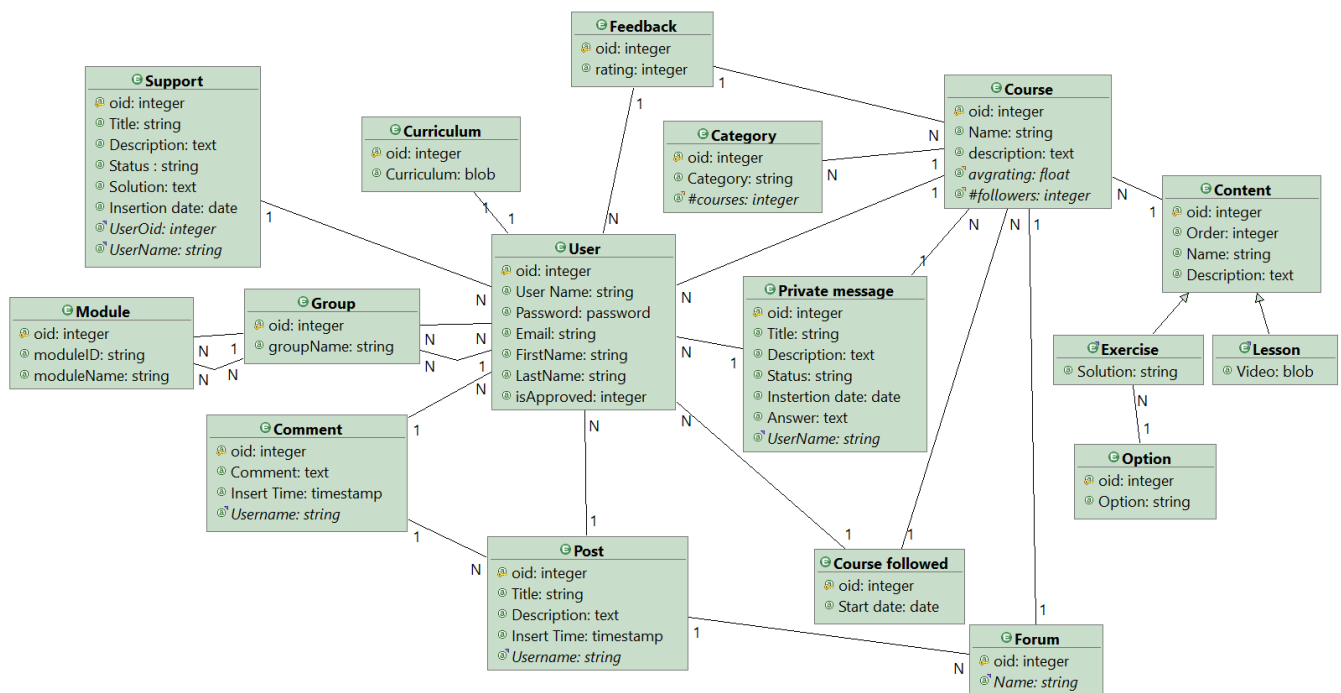
2. Domain Model

Figure 1: Domain Model

Using the Entity-Relationship notation, we have defined the data structure used by the Web application. The domain model is mapped onto an Apache Derby database.

The User Model is the portion of the Domain Model considered by the Login Operation to check the user credentials. It is composed of the **User**, **Group** and **Module** entities and the relationships between them. These three entities are automatically added to the Domain Model when you create a new Web Project and, since they are fundamental for user authentication, they cannot be deleted.

We added 14 new entities:

- **Course:** it stores all the information about a course including some statistics obtained through calculated attributes.
- **Content**: this entity stores all the course materials, which can be classified into two types:
  - **Lesson**: in addition to the content information, it includes a video lesson or tutorial.
  - **Exercise**: exercises can be modeled as either open-ended questions or quizzes with multiple options.
  - The options for a quiz are stored in the **Option** entity.
- **Course followed**: this entity tracks the courses that a specific user has followed.
- **Forum**: each course has a dedicated forum where users and tutors can post and engage in discussions.
- **Post**: this entity records the content of posts, the timestamp of when they were posted, and the username of the poster.
- **Comment**: users and tutors can comment on posts. This entity stores the content of comments, the timestamp of their insertion, and the username of the profile that made the comment.
- **Private message**: users can also send private messages to the tutor. For each course, there is a dedicated messaging system where users can write to seek clarifications or ask questions privately.
- **Category**: each course is part of a category. The categories are typically named after the programming language that the course teaches. The existence of these categories also greatly helps users in searching for courses related to the programming language they are interested in.
- **Feedback**: this entity tracks the ratings that users give to a specific course. The feedback is expressed as an integer value ranging from 1 to 5.
- **Curriculum**: if the user is a tutor, they are required to provide their personal information along with a curriculum vitae (CV). The curriculum vitae is then evaluated by the administration, and if it meets the requirements, the tutor is granted access to the platform.
- **Support**: tutors and users can send support requests to the administrator regarding various aspects or issues on the platform. This entity stores the corresponding information.

3. Home view

The home view is the first page shown, it is public and accessible by everyone.

On the platform there are three groups of users:

- Administrators
- Tutors
- Regular users

To perform actions on the site, users are required to log in. If a user does not have credentials, they must register on the same page. After logging in, the user will be redirected to the corresponding site view.

The user registration does not need any confirmation. The tutor needs the approval of the administrator.

This is how this view is modelled in WebRatio:
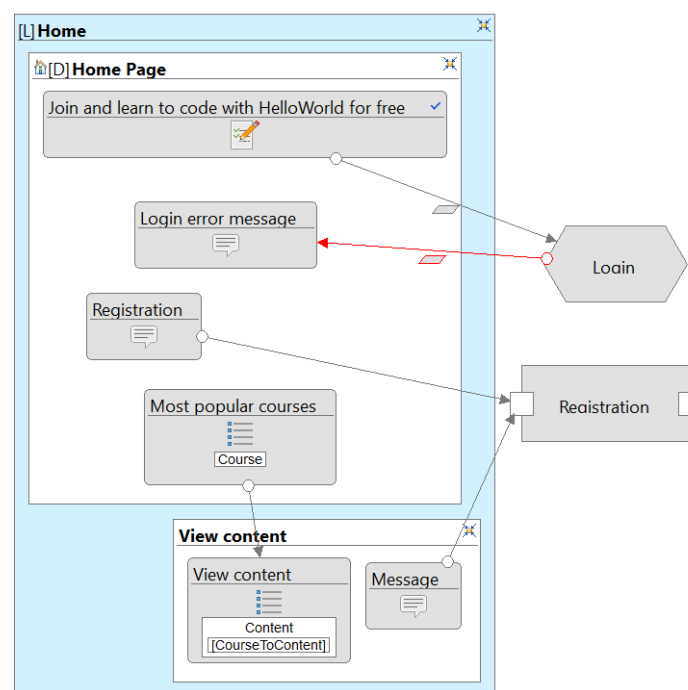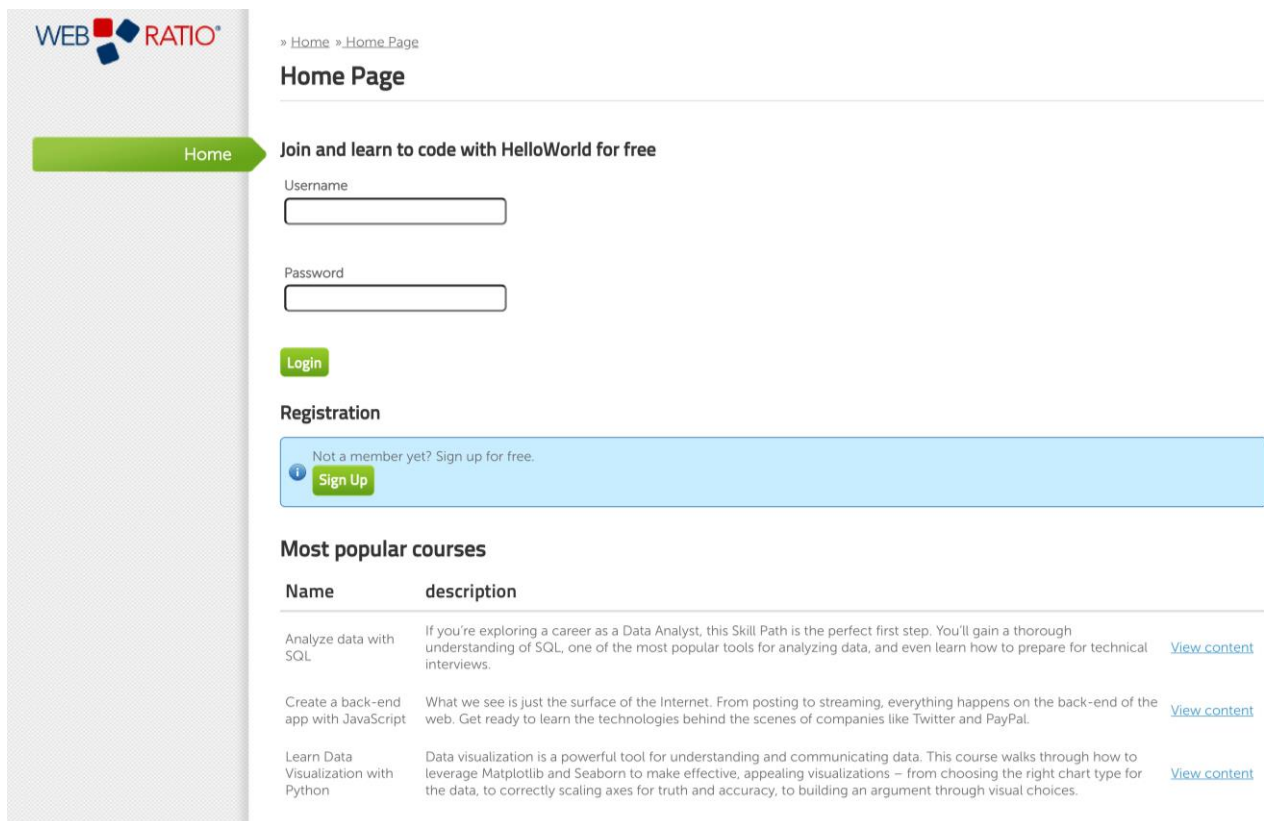
Figure 2: Home view

At the top section of this homepage, there is a form for entering authentication credentials.

A message prompts users to sign up for the platform in order to access the course material.

In the lower section of the homepage, new users who are unfamiliar with the platform can get a little overview of the available courses by browsing through a list of the three most popular courses. The popularity of these courses is determined by the number of followers. This feature is useful for new users as it provides an idea of the platform's organization and the kind of topics covered.

This is what the generated page looks like:

Figure 3: Home page

The registration module is modeled as follows:
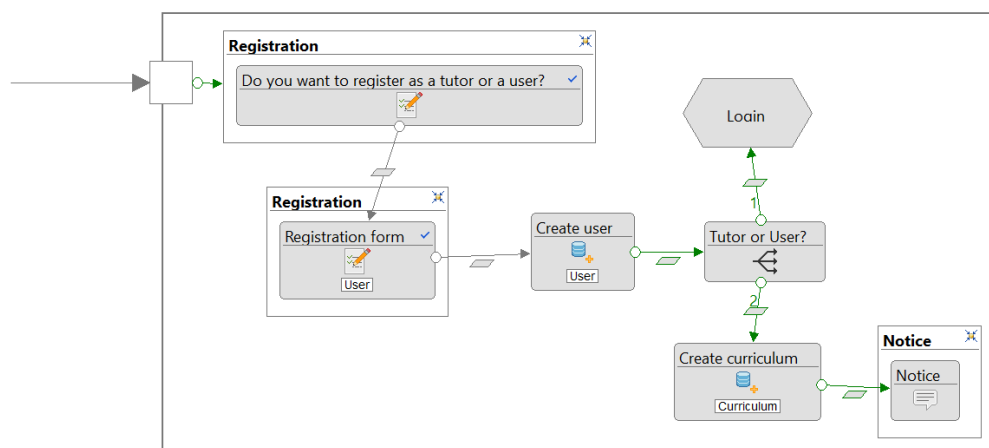
Figure 4: Registration module

The form presented to users during the registration process differs based on whether they want to register as a tutor or a regular user.

Tutors have an additional information field to fill in: their curriculum vitae.

4

Registration for regular users does not require any confirmation. Therefore, after creating a user instance in the database, regular users are automatically logged in. On the other hand, tutors need to wait for their curriculum vitae to be evaluated by the administration. If their curriculum meets the requirements, tutors are granted access to the platform.
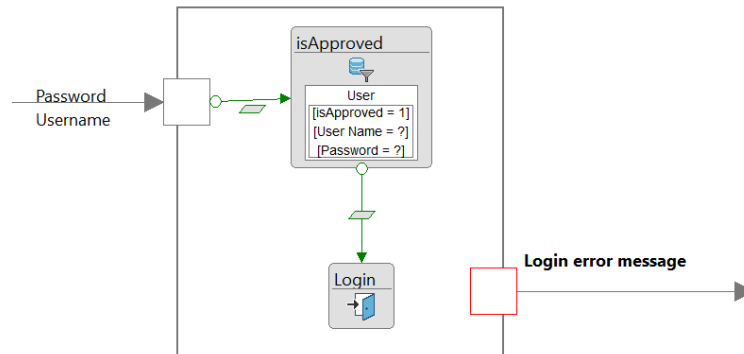
Figure 5: Login action

The figure above depicts the login action. Regular users can always access the platform after registration, while tutors can only access it if they have already been approved.

This is the generated registration form for tutors. For users, the curriculum field is "hidden" using a visibility condition.

Figure 6: Registration form

4. Administration view

This is how the administration view is modelled in WebRatio:
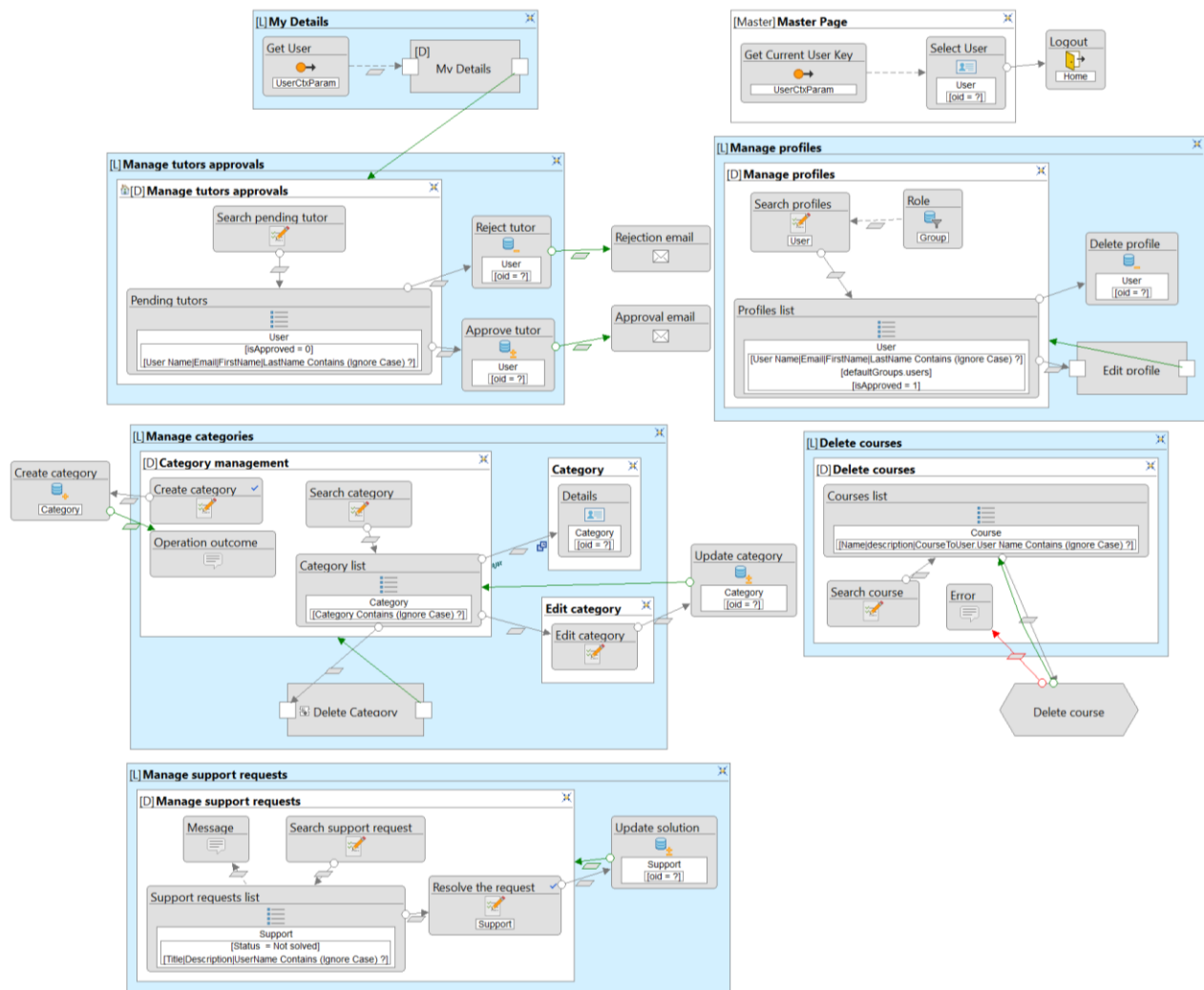
Figure 7: Administration view

After logging in, every view includes a master page that allows users to log out from any page at any time. We chose to put the logout button in upper right part of the page, as shown in the next picture.

The home page of the administration panel is dedicated to managing tutor approvals. The administrator is shown a list of all the pending tutors, and based on their curriculum, they can approve or reject them.

If a tutor is approved, the attribute "isApproved" related to the user with that oid is set to 1. An email is then sent to the tutor, notifying them that they can now log in to the platform and start uploading courses.

On the other hand, if a tutor is rejected, they are removed from the database. Additionally, a rejection email is sent to the tutor, informing them of the decision.

6

Figure 8: Manage tutor approvals section

A search function to search through the tutor approvals is implemented through a form component.

The administrator can also edit and delete profiles:

Figure 9: Manage profiles section

The page includes a search function that allows the administrator to perform searches based on user groups too. This functionality is provided through a preloaded selection field in a form component.

This is the "Profile Details" module, which is frequently used in the project. In the "My Details" section, each user can edit their profile data using this module. Additionally, the administrator has the ability to edit the profile of any user on the platform through this module.
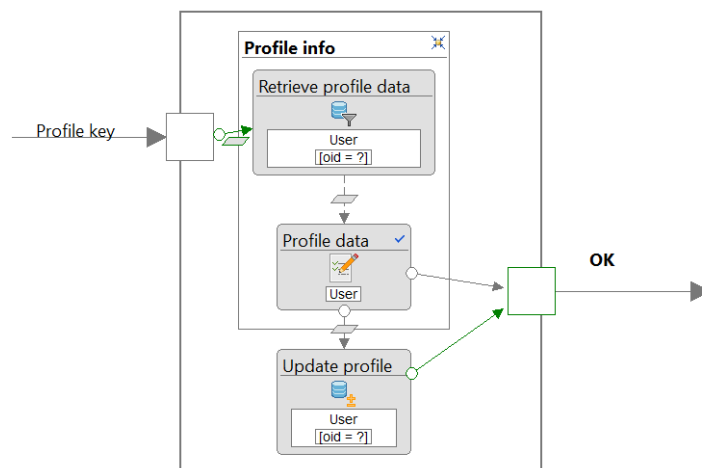
Figure 10: Profile details module

This is how the module works:

- The information about a profile is retrieved from the database using a specific user oid.
- The fields of a form component are preloaded with the retrieved data.
- Th fields are editable, when the modifications are made, just save the new information.
- The "Save" navigation flow sets the parameter binding required to update the user instance in the database.

Figure 11: Profile info page

As mentioned earlier, each course is associated with a category. The presence of categories assists users in finding courses that are relevant to the programming language they are interested in.

One of the tasks of the administrator is to manage categories. They can perform the following actions:

- Create new categories by filling out a form.
- Edit existing categories using a preloaded form, similar to the one used for profile editing.
- Search for a specific category using another form.
- Check the number of courses associated with a particular category.

Figure 12: Category management section

The most complex function of this section is the deletion of a category, which is handled by the "Delete category" module.

Within this module, a switch component is utilized to determine the number of courses associated with a specific category. If this number is 0, the category is promptly deleted. However, if there are courses associated with the category, they must be reassigned to another category before the deletion can be completed.
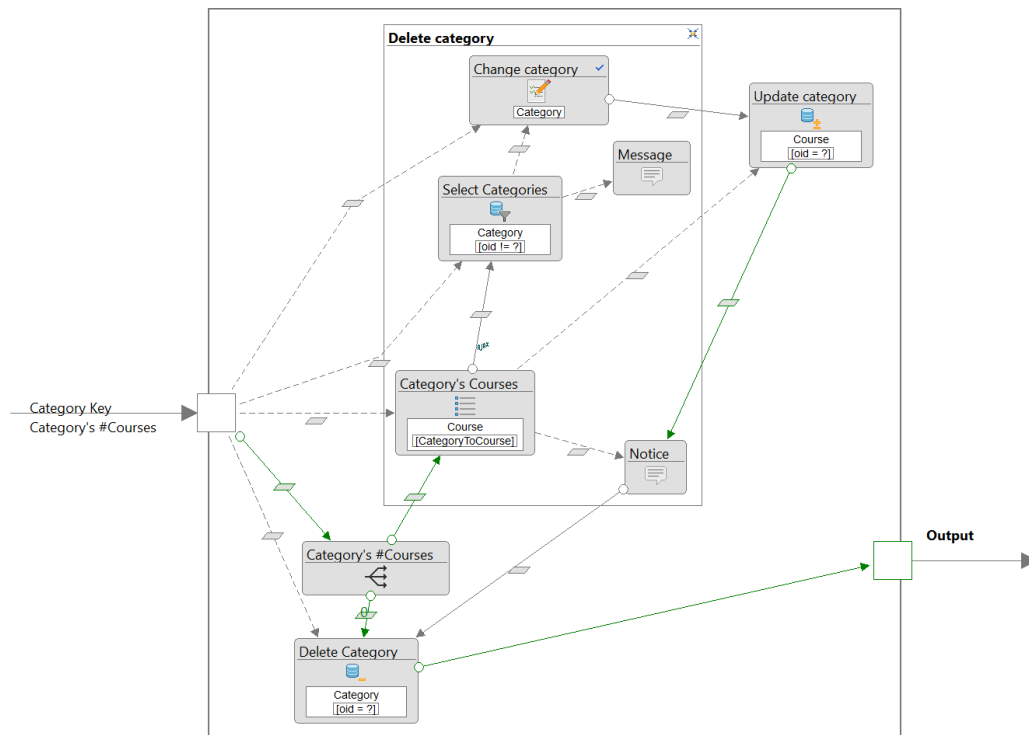
Figure 12: Delete category module

This is what the generated page looks like when there are courses in the category:

Figure 12: Delete category page

After all the courses have been reassigned, a button labeled 'Delete category' appears and the category can finally be deleted.

In the "Delete courses" section, the administrator can view, search, and delete any course available on the platform.

Figure 12: Delete courses section

The last function of the administrator is to respond to support requests sent by tutors and users.

This panel includes a list of all pending support requests, indicated by the status "Not solved". The administrator can search through the list and click on the "Resolve" button to provide an answer to the support request.

Once the administrator resolves the support request, its status is changed to "Solved". The user or tutor who submitted the support request can then view the solution proposed by the administrator in the corresponding support request section.

Further details about this aspect will be provided in the description of the other views.

The support requests are sorted in chronological order, from the oldest to the newest. This ensures that the administrator prioritizes answering the questions that were received first.

Manage support requests

Logout

Search support request

Search

Search

Support requests list

Insertion date	UserOid	UserName	Title	Description	
6/11/23	4	ciro	Error accessing course content	I enrolled in a course, but I'm unable to access the course materials. When I click on the lessons, it redirects me to an error page. Can you assist me in accessing the course content?	Resolve
6/11/23	6	LeonardoDV	Profile removal	How can I remove my profile?	Resolve
6/11/23	7	jess	Video management	How can I change the name of an already created video?	Resolve
6/11/23	5	alessio	Request for support to modify the lessons of a course.	Hello, I would like to edit the lessons of a video but I don't know how to do it. Thank you in advance for your help.	Resolve
6/13/23	20	fab	Error submitting answer to quiz	Hello, I completed the quiz for the 'Learn C' course, but when I tried to submit my answer, I received an error message. It seems like my quiz submission is not going through. Could you look into this and help me submit my quiz successfully?	Resolve

Generated by Web Ratio®

Figure 13: Manage support request section

5. Tutor view

This is how the tutor view is modelled in WebRatio:
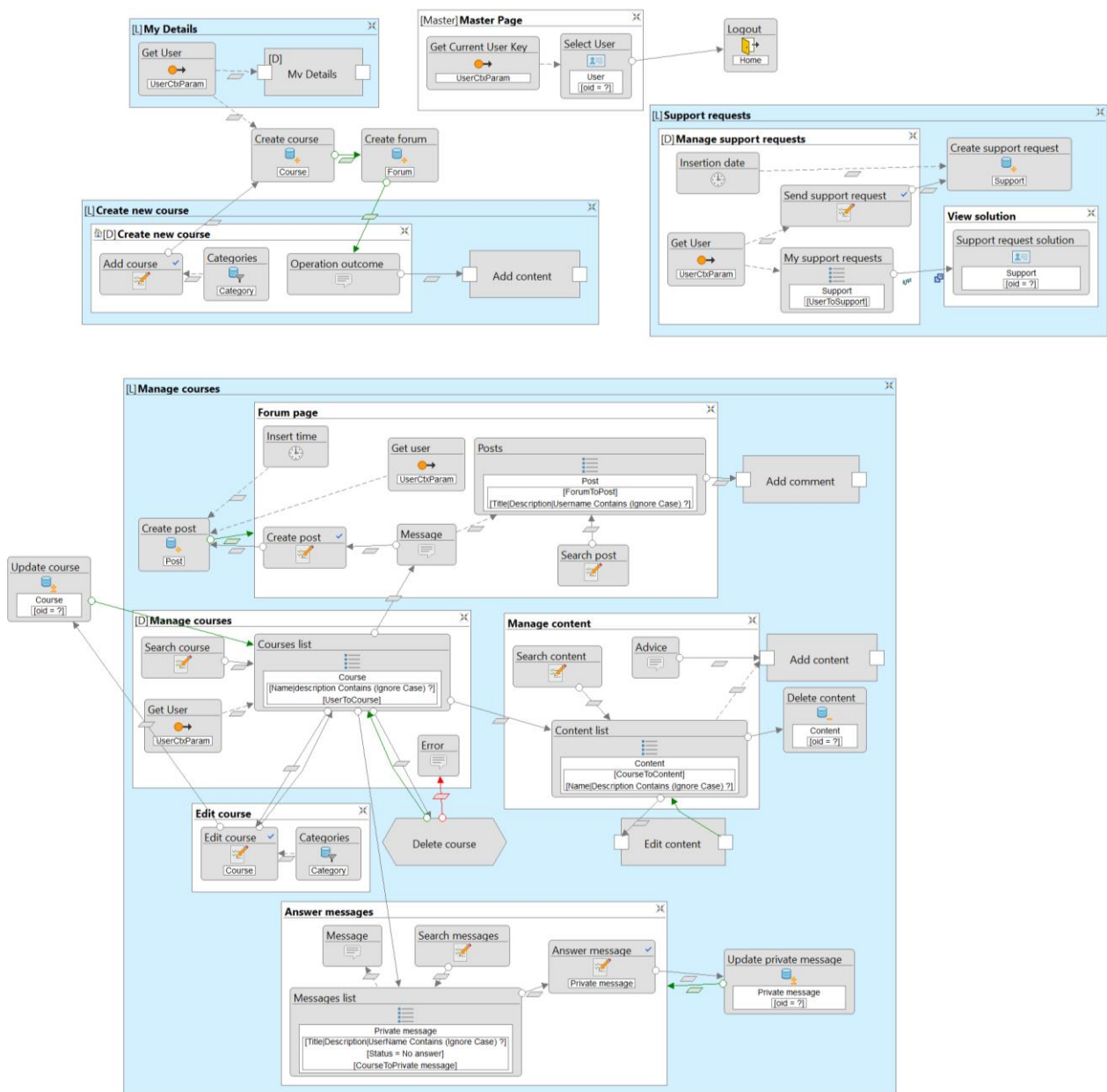
Figure 14: Tutor view

The master page for the logout and the "My Details" section to modify the profile are almost identical to the ones we saw for the Administration view.

The first section shown to the tutor is the one that allows them to create new courses. In order to create a new course, the tutor must fill out a form that asks for the name of the course, a description of it, and a category selected from the options stored in the database.

Submitting the form allows us to gather the necessary information for creating a course instance in the database. Simultaneously, an associated forum for the course is instantiated.

13

After creating a course, the tutor has the option to add content immediately or at a later time in the 'Manage courses' section.

If the tutor chooses the first option, they can easily add content by following these steps:

- Click on the button labeled 'Add Content,' which will be displayed once the course has been created.
- Two new forms will appear, providing the tutor with the necessary fields to insert all the required information for creating lessons or exercises.

This picture displays how the page appears after creating a new course:

Figure 15: Create course section

These are the additional forms shown when the tutor clicks on "Add content":

Figure 16: Add content page

After clicking "Save," a new window will appear, presenting an additional form where the tutor can add options and the solution:

Figure 17: Add content page (options and solution window)

The process of adding content is modelled through 2 modules, "Add content" and "Add options and solution". As previously mentioned, tutors have the flexibility to add content either immediately or at a later time within the 'Manage Courses' section. In both cases the content addition is implemented using these modules.
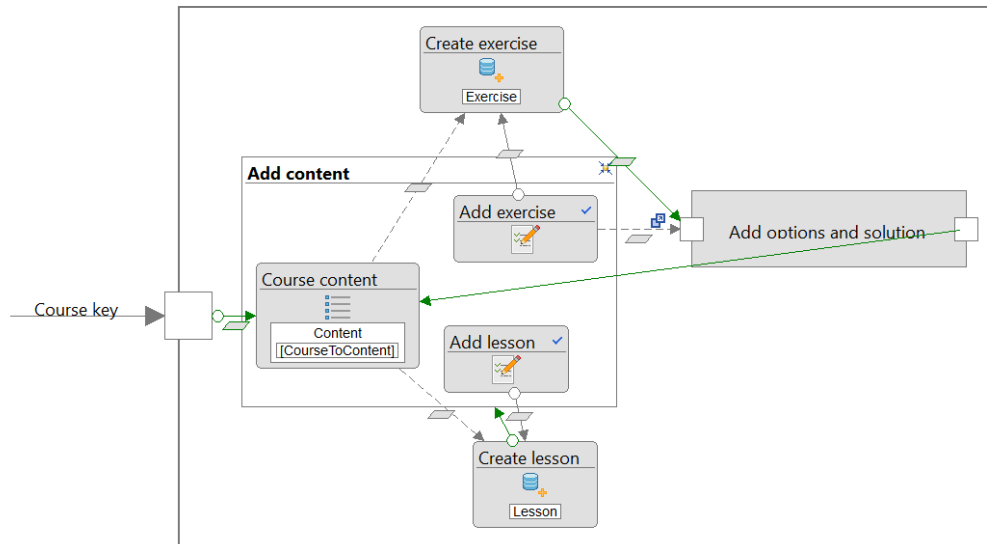
Figure 18: Add content module
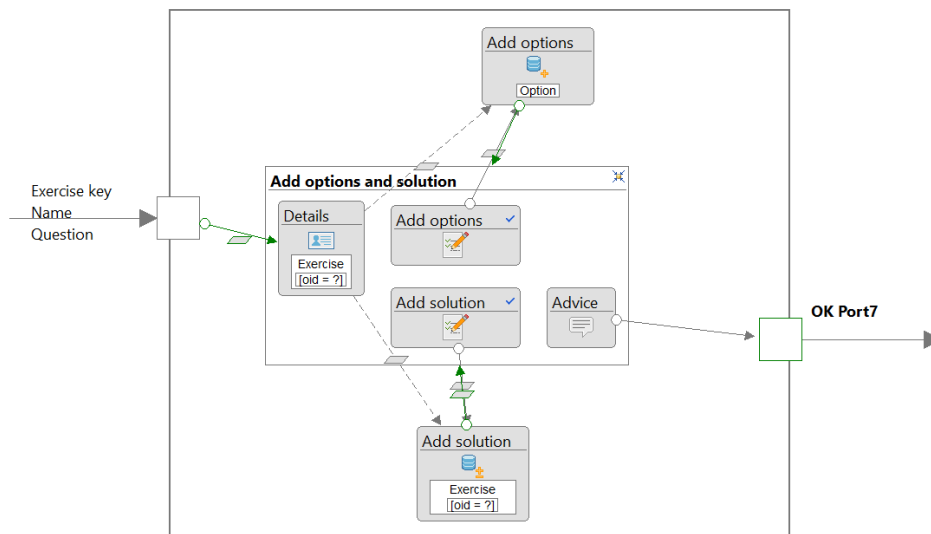
Figure 19: Add options and solution

Let's now talk about the section which has the highest number of tasks in this view: "Manage courses".

This section allows tutors to monitor the courses they have created. They can access a comprehensive list of their courses, search for specific ones, and view various statistics such as the average rating provided by followers and the number of followers for each course.
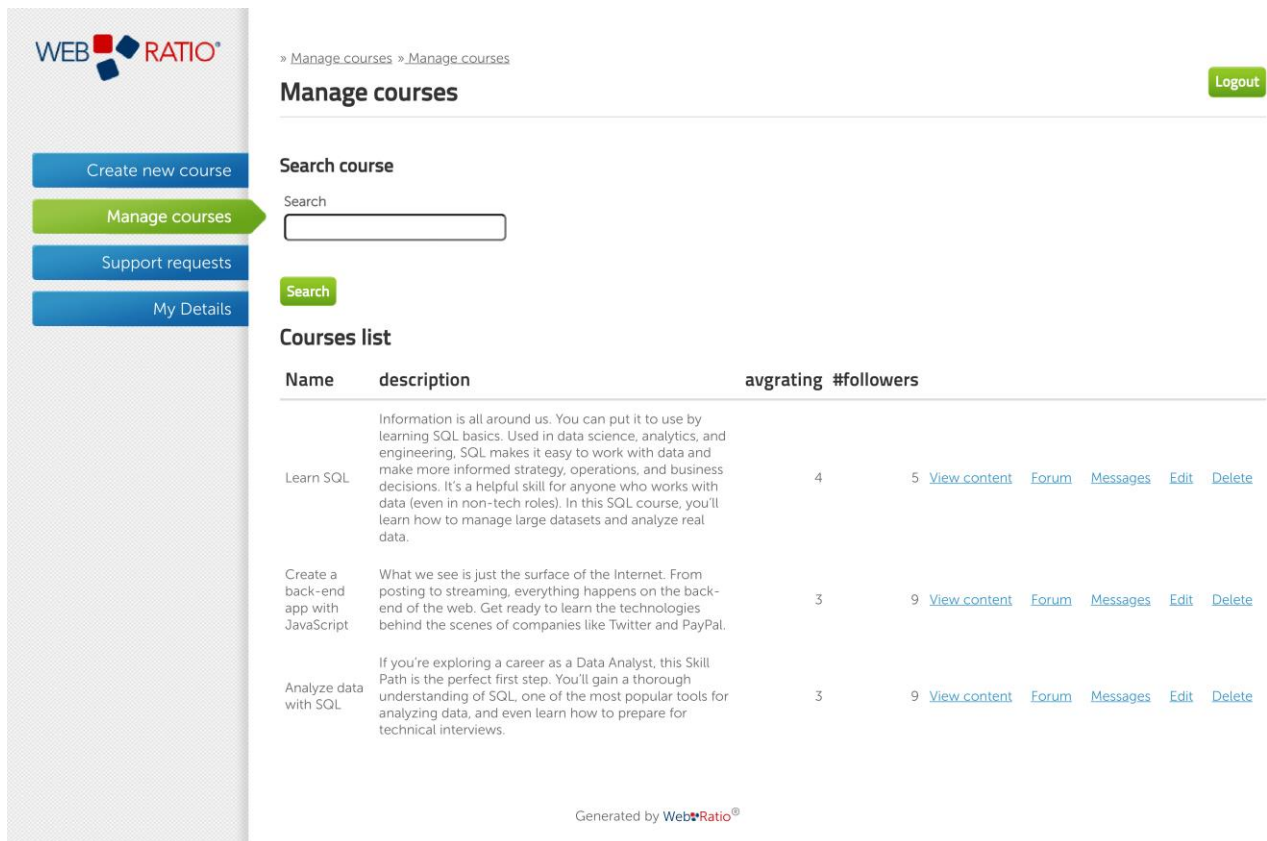
Figure 19: Manage courses section

When the tutor clicks on the corresponding button, it triggers a series of events:
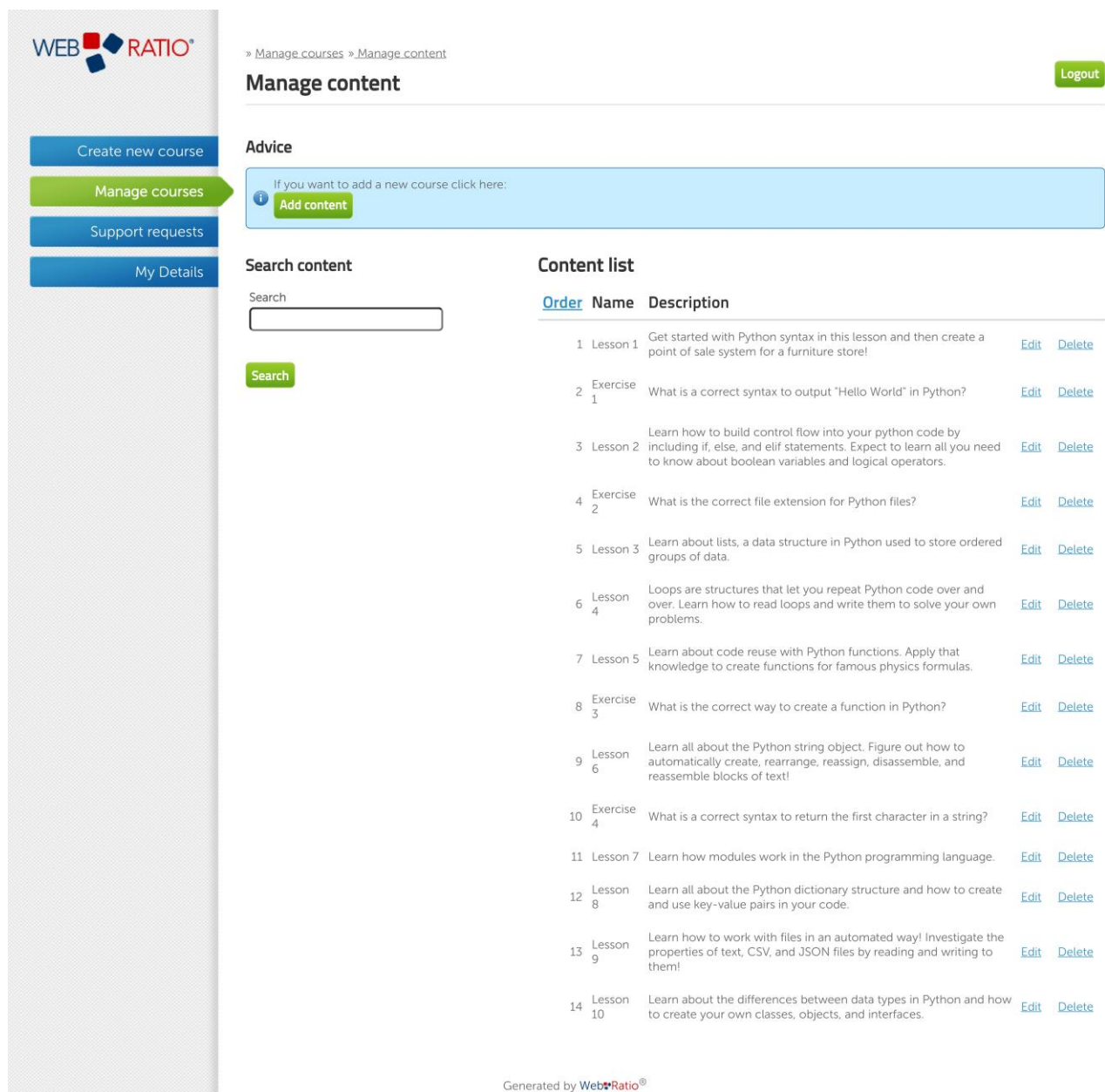
- "View content": this action opens another page that enables the tutor to manage the course contents. The page provides options to add, edit, and delete course materials.
- "Forum": this action grants the tutor access to the course forum, where they can participate in discussions and interact with students.
- "Messages": by clicking on this button, the tutor can check for any private messages specifically related to the course. This feature allows them to address questions or concerns from students.
- "Edit": this action allows the tutor to modify the course information, including its name, description, and category.
- "Delete": by selecting this option, the tutor can delete the course along with all its associated content.

Let's explore how each of them works and how they were modelled.

By clicking on "View content," the page "Manage content" will open. In this section, the tutor can access the content uploaded to the course and perform various actions such as adding additional lessons or exercises, editing existing content, or deleting them. Similarly to other sections of the application, the tutor can search in the list content by using keywords.

The operation of content addition is implemented through with the "Add content" module seen before.

Deleting content is a straightforward process implemented through a delete operation component. When the tutor selects the content they wish to remove, the component removes the instance with the corresponding oid.

Figure 20: Manage content page

The "Edit" function is a little bit more articulated.

It is modelled through the "Edit content" module:
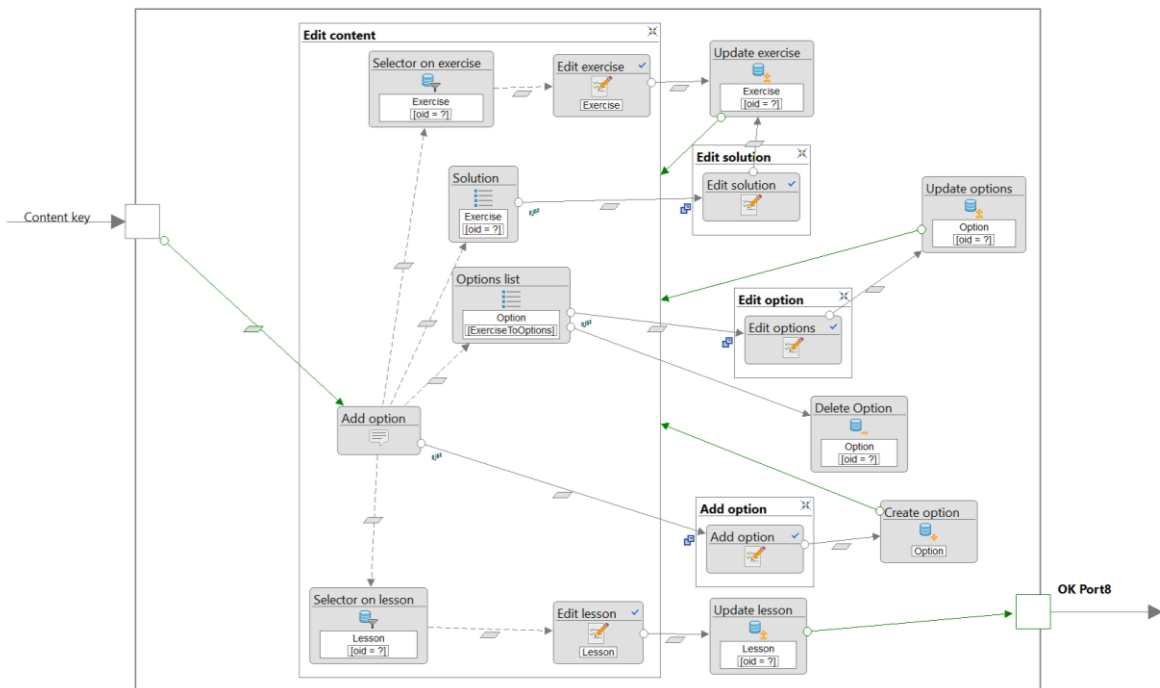
Figure 21: Edit content module

Based on the fact that the content is a lesson or an exercise, a different modifiable and precompiled form is shown.

To achieve this, we implemented a visibility condition that determines which form to display.

If the content is an exercise, the tutor is provided with additional operations such as adding new options, editing existing options, or deleting them.

Furthermore, the solution for the exercise can also be edited.

Figure 22: Manage content page

When the tutor clicks on "Add option" or the "Edit" button in the options list or next to the solution, a window, that resembles the one seen earlier on the "Add content" page, will open.

In this window, the tutor can enter the desired modifications.

Figure 23: Edit option window

By clicking on the button "Forum" in the "Menage courses" main list, the tutor can access to the course forum.

On the forum page, there is a list of all the posts, sorted by the time they were inserted (with the newest posts shown first in descending order). Each post also displays the username of the person who posted it. At the top of the page, there is a button to create a new post and a search form.
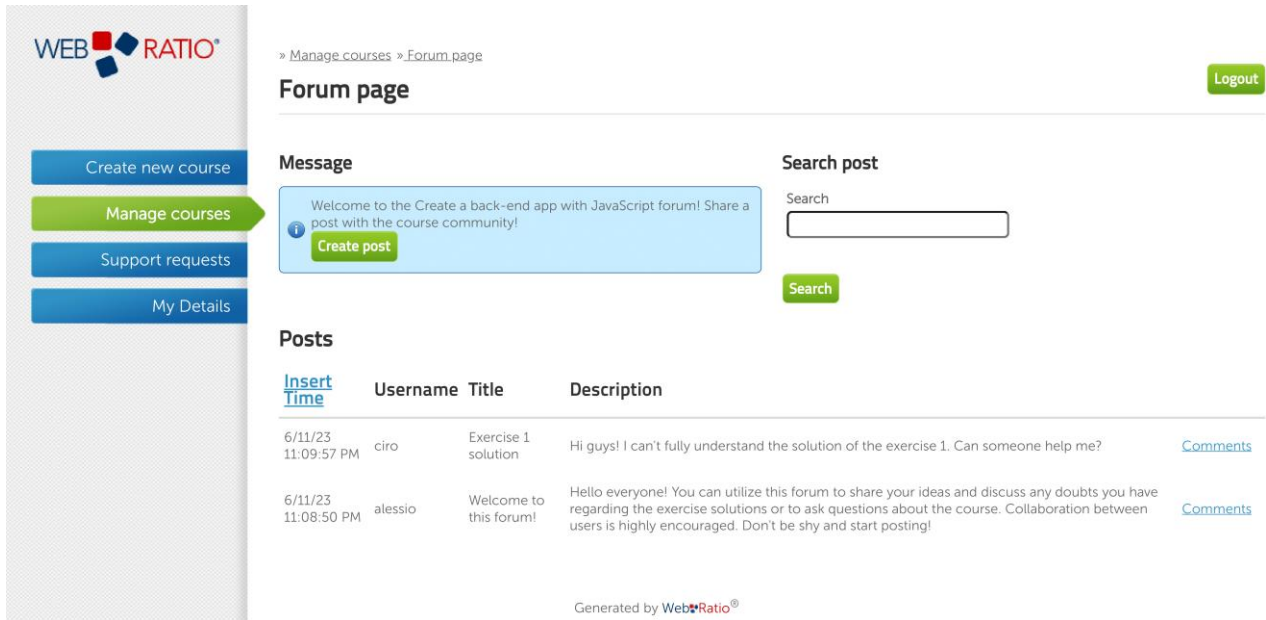
Figure 24: Forum page

To add a comment on a post the tutor has to click on the "Comments" button.

The module "Add comments" is used to implement the comments page and its functions:
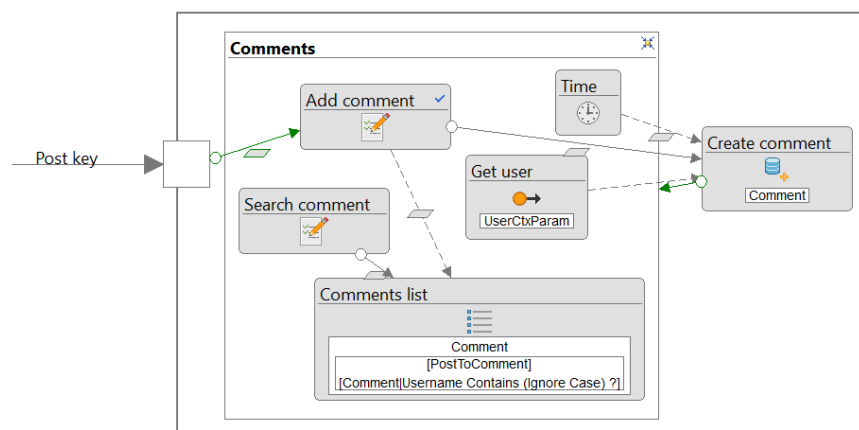
Figure 25: Add comment module

This is how the generated page looks:

Figure 26: Comments page

The user who follows a course can ask questions to the tutor and the community publicly, or privately contact the tutor. Each course has a sort of "mailbox" to help tutors organize the various messages they receive.

Private messages are organized similarly to support requests, but in this case, the recipient is the tutor instead of the administrator. The properties of a message include:

- Insertion date: this field is used to order the list of messages from oldest to newest.
- Username: this indicates the username of the user who is asking for clarification.
- Title: similar to the subject line in an email, the title helps understand the topic of the message.
- Description: this field contains the actual question or clarification being requested.
- Status: the status can be "No answer" or "Answered," depending on whether the tutor has already replied or not.

Figure 27: Answer messages page

By clicking on "Answer," appears a form component where the tutor can enter their response to the user's message.

Continuing with the "Manage courses" functions, there are two remaining options: "Edit" and "Delete":

- "Delete" allows the tutor to remove the selected course from the database.
- "Edit" opens a new page displaying a summary of the previous information about the course, along with the possibility of modifying and updating the database accordingly.

Figure 28: Edit course page

Now, let's analyze the last section of the tutor view: "Support requests."

We have already discussed how support requests work for the administrator, but let's see how they function for tutors and users.

Figure 28: Support requests section

At the top of the page, there is a form for submitting support request information.

In the lower part of the page, there is a list of previously submitted support requests, ordered by insertion date. The list displays the title, description, and status of each request.

By clicking on the "Details" button, a window opens showing the solution (if available) and a summary of the support request.

Figure 29: Support request details

6. User view

This is how the tutor view is modelled in WebRatio:
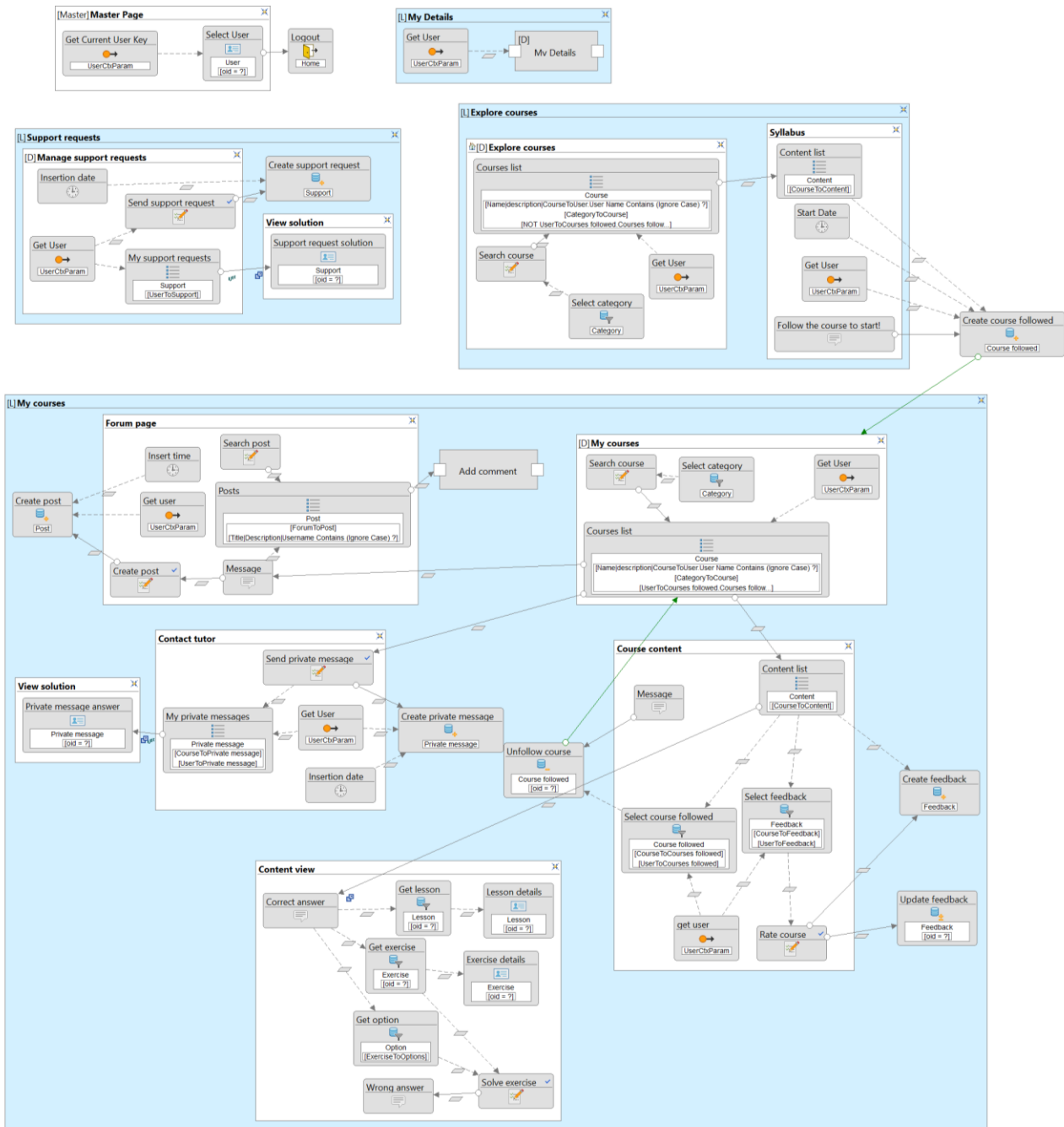
Figure 30: User view

The master page for the logout, the "My Details" and "Support requests" sections are almost identical to the ones we saw for the Tutor view.

The "Explore courses" section serves as the home for this view. Here, users can access all the courses available on the platform, search for specific courses (including by category), and view the content of each course by clicking on "View content."

By default, the course list is ordered based on the number of followers each course has, with the most popular courses displayed first. However, users also have the option to sort the list based on the average rating given by users who follow the course.

Figure 31: Explore courses section

Figure 32: Syllabus page

When users click on "View content," a list of all the lessons and exercises for that course will appear. This list serves as the syllabus for the course. However, to actually watch the videos, solve the exercises, and participate in the forum, the user needs to follow the course. To do this, the user simply has to click on the "Follow" button located in the top half of the page.

When a user starts following a course, they are redirected to the "My courses" section, which displays all the courses they follow. These courses no longer appear in the "Explore courses" section after being followed.

The "My courses" section has a list with all the followed courses and a search form similar to the one in the "Explore courses" section.

The "My courses" section features a comprehensive list of all the courses the user is currently following, accompanied by a search form that closely resembles the one in the "Explore courses" section. Each course within the list offers three distinct options:

- "View content": allows users to see the list of exercises and lessons of the course.
- "Forum": this option grants access to the dedicated forum for the specific course. Users can engage in discussions, seek assistance, or contribute to the community by sharing insights, questions, or thoughts.
- "Contact tutor": users can utilize this option to communicate with the tutor directly. Whether seeking clarification or having specific questions about the course content, users

can write and submit their inquiries to the tutor through this feature. The tutor will then provide necessary guidance or answers to help users progress effectively.

Figure 33: My courses section

Once users click on "View content," the list of all the contents for that course will appear.

At the top of the page, there is a form with a selection field that allows users to rate the course from 1 to 5. The feedback can be updated over time, as every time a user selects a new rating, the feedback for that user is updated in the database.

Users also have the option to unfollow the course by clicking on the "Unfollow course" button.

Next to each content, there is a "View" button. When clicked, a window opens displaying the content based on its type. If the content is a lesson, the window will show the video lesson along with a brief summary of its information. On the other hand, if the content is an exercise, users will find a form to solve the quiz. The visibility of these elements is controlled through visibility conditions.

Figure 34: My courses content

When the content is a lesson, the window shown below will open. To watch the video, users have to click on the underlined video name, and a player will open up.
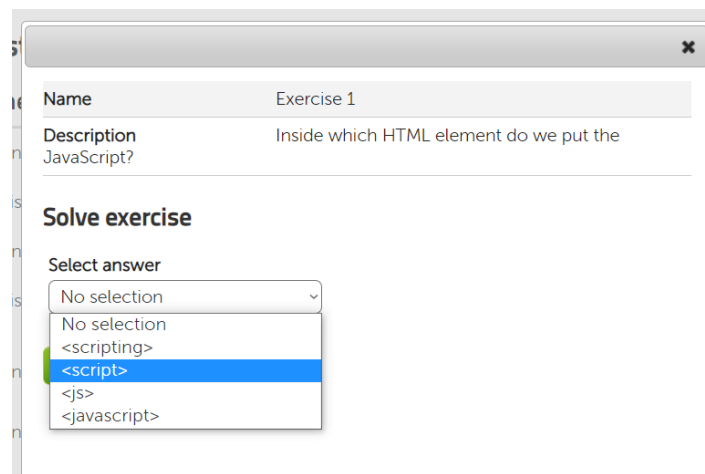

Figure 35: Lesson window

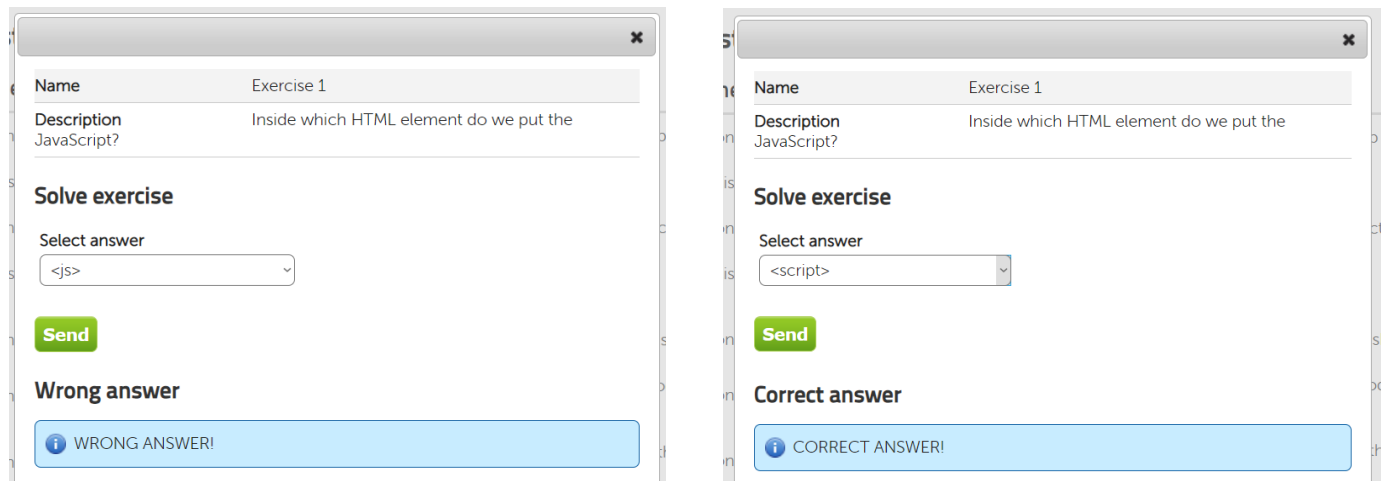When the content is an exercise, users can answer the quiz question by selecting from the available options.

Figure 36: Exercise window

When the user sends their answer, the application will compare it with the solution stored in the database at the time the exercise was created by the tutor.

Depending on whether the answer is correct or incorrect, a different message will be displayed to the user. This is achieved by using visibility conditions.

Figure 37: Feedback provided to the user regarding their answer

The forum of the course can also be accessed from the "My courses" list. The functionalities and their implementation are the same as those shown in the "Tutor view."

From the "My courses" list, users can also write to the tutor to clarify doubts clicking on the "Contact tutor" button.

Figure 38: Contact tutor section

At the top of the page, there is a form where the user can submit the title and description of a private message.

In the lower part of the page, there is a list of previously submitted private messages, sorted by their insertion date. The list includes the title, description, and status of each message, indicating whether it has been answered or not.

Clicking on the "Details" button opens a new window that displays the answer, if available, along with a summary of the private message that was sent.
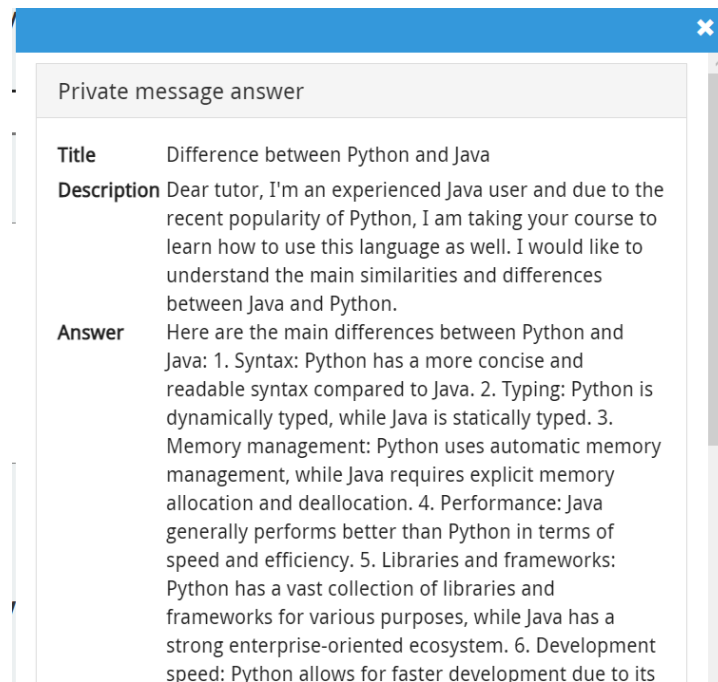
Figure 39: Private message answer window

7. Module definitions view

A Module Definition View is a container of reusable portions of an interaction flow model and action definition.

In this application were defined 2 action and 7 hybrid modules.
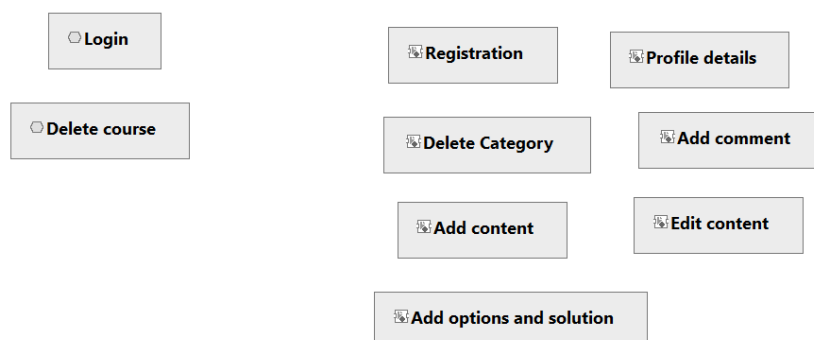
We already analyzed most of them in the previous paragraphs.

Figure 40: Module definitions view