Software Engineering

2022./2023.

Learning Together Platform

Documentation, Rev. *1*

Group: *Team Eder*

Leader: *José Matos*

Date: *18. 11. 2022*

Teacher: *Nikolina Frid*

Contents

1. Documentation change log

Continuous updating

Rev.	**Change description**	**Authors**	**Date**
0.1	Functional and non-functional requirements. Project description. All done in a team meeting.	Every member	23/10/2022
0.2	Added use cases and description for each one.	Every member	4/11/2022
0.3	Added the use case diagram after team meeting where it was constructed.	José Matos	15/11/2022
0.4	Added the 4 sequence diagrams.	Lourenço Carvalho, Maria Costa, Francisco Antunes, Rui Guimarães	16/11/2022
0.5	Added Class and Database diagrams.	Lourenço Carvalho	18/11/2022
0.6	Described system architecture and design.	José Matos	18/11/2022
0.7	Final refinement before 1st revision.	Every member	18/11/2022
1.0			

Continued on next page

Continued from previous page

Rev.	**Change description**	**Authors**	**Date**
1.1			
1.2			
1.3			
1.5			
1.5.1			
2.0			

2. Project assignment description

Learning Together is a new web platform for knowledge exchange. The idea of this platform is that anyone can create their own course on any topic and upload written, video and audio materials. In addition, this platform enables interactive communication with the person holding the course through comments and live chats. An unregistered public user can view the courses that are available, but to access the course, it is necessary to register with an email address. Registered users can browse available courses and enroll in any available course. Some courses may require payment. To create a new course, the following information is required: name, category (choose one of the existing ones in the system), an estimate of how much time the average user needs to master all the materials, price (optional) and the course content. The content of the course is divided into points (teaching units). Each unit, along with a mandatory short description, can contain written, audio and video materials that registered users can download (the materials are not available to users who are not course participants). The course owner can add or remove materials at any time. The owner of the course can enable the option of online chat with participants and indicate the dates and times when it will be available and the maximum number of participants. Interested course participants can register for the offered dates1. Enrolled course participants can comment and rate the course. Comments and ratings are publicly visible (also available to non-registered users). Registered users will have a private and a public profile. Private profile will contain users' personal information, payment details and information about courses taken and/or created. Also, on their private profile, users can select one or more categories of courses that they are interested to get recommendations and updates. A public profile is required only for users who create and offer courses to other users. The application is maintained by system administrators who can add or delete any user, change the category of one of the existing courses and delete any course.

3. Software specification

3.1 Functional requirements

Stakeholders:

1. Students
2. Teachers
3. Team Leaders
4. Developers
5. Administrators
6. Google Calendar API Provider
7. Google Pay API Provider
8. Tawk.to Integration Provider

Actors and their functional requirements:

1. Unregistered users can:
   (a) View available courses
   (b) Register in the website
   (c) See public users profile

1. Registered users can:
   (a) View available courses
   (b) Edit profile
   (c) Interact with courses
      i. Enroll into a course
      ii. Comment and rate a specific course
      iii. Communicate through a live chat with the creator of the course
      iv. Access course materials
   (d) Create and edit their own courses
      i. Add or remove materials at any time
      ii. Set live chat properties

        A. Schedule
        B. Maximum number of users

2. System administrators can:
    (a) Add or delete any user
    (b) Accept courses waiting for approval
    (c) Change category of any existing course
    (d) Delete any existing course

3.1.1 Use cases

UC1 - View Available Courses

- **Main participant:** Unregistered and Registered Users
- **Goal:** Search for a specific course the user wants to apply or just see the available courses in the website
- **Participants:** Database
- **Prerequisites:** None
- **Description of the basic course:**
  1. In the main page, the website shows to the user some course recommendations
  2. The user can filter the courses by categories
  3. If the user wants to search for a specific course, he can do it directly by writing the course's name
  4. System presents a list of courses with the given input
- **Description of possible deviations:**
  1. The user cannot access to the course
     - The user needs to be registered.
  2. The user tries to search for a specific course that doesn't exist in the website

UC2 - See public profile

- **Main participant:** Unregistered and Registered Users
- **Goal:** Users can search for specific public user to find available courses(UC1-View Available Courses)
- **Participants:** System administrators
- **Prerequisites:** None
- **Description of the basic course:**
  1. Registered and unregistered users can search for public profiles, searching for the public user name, to see specific courses
  2. System presents a list of public profiles according to the input
  3. Registered users can enroll public user's courses (UC6)
  4. System administrator can ban/delete public user
- **Description of possible deviation:**
  1. Main actor search for a non-existent public user

UC3 - Register

- **Main participant:** Unregistered User
- **Goal:** Having access to a course or create one
- **Participants:** Database
- **Prerequisites:** None
- **Description of the basic course:**
  1. The future user needs to input their name, surname, email, username, password and an avatar.
  2. After input all the data, he needs to submit the information.
  3. If all of the data is correct, according with the parameters, then the new actor is successfully registered and goes to the initial page of the website (already logged).
- **Description of possible deviations:**
  1. If there is already a username and/or an email equal to the one filled in.
     - System will return a warning saying that the user needs to change that data.
  2. System warns if the password is not in accordance with the security conditions.
     - The user needs to change is password for one that is in conformity with the specifications.
  3. The email is not valid.
     - The website will return an warning message saying that the user has to change the email to one that is valid (with @).
  4. The system returns a message saying that the number is not valid.
     - The user has to change the number to one that is valid (only with numbers)

UC4 - Login

- **Main participant:** Registered Users
- **Goal:** Access more functionalities of the website
- **Participants:** Database
- **Prerequisites:** None
- **Description of the basic course:**
  1. The main actor enters their login credentials (username and password)
  2. The login credentials are validated in the database

3. It's granted access to more functionalities of the website (enroll in a course, create courses, etc...)

- **Description of possible deviations:**
  2.a The login credentials are wrong
    1. The main actor tries to login again with a different username/password
    2. The main actor resets the password
    3. The main actor cancels the login

UC5 - Edit user's profile

- **Main participant:** Registered user
- **Goal:** Edit profile information
- **Participants:** Database
- **Prerequisites:** User should be registered and logged in (UC3 and UC4)
- **Description of the basic course:**
  1. Main actor selects form to edit profile
  2. Profile information is listed
  3. Main actor selects which information to update
  4. User enters updated information
  5. System allows main actor to save his updated information
- **Description of possible deviations:**
  5.a Main actor tries to exit without saving
    1. System prompts user to allow saving of changes
    2. Main actor exits without saving

UC6 - Enroll course

- **Main participant:** Registered user
- **Goal:** User can enroll a course
- **Participants:** None
- **Prerequisites:** Have a private account
- **Description of the basic course:**
  1. The main actor enroll on one or multiple courses
  2. In case of paid course the payment information is validated by the database
  3. System grants the user access to all of that courses functionalities (material, live chat scheduling, etc.)
- **Description of possible deviation:**

1. The payment details are incorrect/ payment fails (in case of paid course)
2. Enrollment deadline already expired

UC7 - Rate/Comment on Course

- **Main participant:** Registered Users
- **Goal:** Comment on another user's course
- **Participants:** Database, Registered User
- **Prerequisites:** User must be logged in (UC4), have selected a course to rate/comment on and be enrolled in that same course
- **Description of the basic course:**
  1. The main actor selects the course he wants to rate/comment on
  2. The main actor writes the comment and/or rates the course(on a scale from 0 to 5)
- **Description of possible deviations:**

UC8 - Communicate through a live chat with creator

- **Main participant:** Registered Users
- **Goal:** An user communicate with the creator during a live chat
- **Participants:** Database, Registered User
- **Prerequisites:** User must be logged in (UC4 - Login) and enrolled in that course.
- **Description of the basic course:**
  1. Main actors enters the live chat at a given schedule
  2. System allows for exchanging of messages between the two participants
- **Description of possible deviations:**
  1. The creator does not turn on the live chat
     - The website returns a message saying that the chat will be turned on in a few moments.
     - If 10 minutes have passed the administrator will be contacted and system notifies the user.
  2. The user is not registered in that course.
     - The user is notified that it is not enrolled in the course.
     - The user can enroll in the course and wait to be accepted by its creator.

UC9 - Access course materials

- **Main participant:** Registered Users
- **Goal:** Access materials from a course
- **Participants:** Database
- **Prerequisites:** User must be logged in (UC4 - Login) and must be enrolled in the course (UC6 - Enroll course)
- **Description of the basic course:**
  1. The main actor enters the course page
  2. The main actor chooses the option to access the materials
  3. System lists all the available materials from that course
- **Description of possible deviations:**

  1 There are no materials available yet

  - 1. The main actor can go back to the course page

UC10 - Create course

- **Main participant:** Registered Users
- **Goal:** Create a new course in the website
- **Participants:** Database
- **Prerequisites:** User must be logged in (UC4 - Login) and must to be using his public profile
- **Description of the basic course:**
  1. The user selects the option to create a course
  2. Step by step the user has to input the course's details
     - Course's name
     - Category
     - An estimate of how much time the average user needs to master all the materials
     - Price(optional)
     - A description about the content of the course
  3. Confirm the course creation
  4. System now lists the course as available
- **Description of possible deviations:**
  1. The course name already exists
  2. Going back to the main page before finishing the course creation

UC11 - Edit course

- **Main participant:** Course Owner

- **Goal:** Make changes to the details
- **Participants:** Database
- **Prerequisites:** Course owner must be logged in (UC4 - Login) with a public profile and the course should belong to him
- **Description of the basic course:**
  1. The Main Actor selects his course
  2. The Main Actor selects the option to edit his course
  3. Then the course owner chooses what he wants to change
     - Add materials(UC12)
     - Remove materials
     - Update existent materials
     - Enable/disable the option of online chat
- **Description of possible deviations:**

UC12 - Add Materials

- **Main participant:** Registered User
- **Goal:** Add materials on the page of the course.
- **Participants:** Database
- **Prerequisites:** Must be logged in (UC4- Login) and the creator of the course.
- **Description of the basic course:**
  1. Main Actor selects their course
  2. Main Actor selects the option to add materials
  3. The creator submits the materials and saves the changes
  4. The system returns a success message and all the materials are available to the users that are enrolled in that course
- **Description of possible deviations:**
  1. The attachment is not in an acceptable format
     - The website returns a warning saying that the file is not in a acceptable format. The main actor needs to submit the file in eligible format.
  2. Actor try to upload a file which size is higher than the imposed limit
     - The website returns a warning saying that the file is too large
  3. Creator attempts to leave without saving the changes.
     - The website returns a message asking to submit or discard.

UC13 - Remove Materials

- **Main participant:** Registered User
- **Goal:** Remove materials from the page of the course
- **Participants:** Database, User
- **Prerequisites:** Main actor must be logged in and be the creator of the course. The course needs to have materials already
- **Description of the basic course:**
  1. Main Actor clicks in their course
  2. Main Actor clicks in a button with the name ”Edit Materials”
  3. Main Actor selects which material(s) to remove from the platform
  4. Main Actor confirms the material’s deletion
- **Description of possible deviations:**
  1. Main actor decides not to delete the selected materials when presented with the confirmation prompt

UC14 - Set live chat properties

- **Main participant:** Registered Users
- **Goal:** Change the properties of a course live chat
- **Participants:** Database, Google Calendar API Provider
- **Prerequisites:** User must be logged in (UC4 - Login) and have created the course (UC10 - Create course)
- **Description of the basic course:**
  1. The main actor, creator of the course, enters the course page
  2. The main actor clicks the Definitions
  3. The main actor clicks the Live Chat
  4. All the properties of the course Live Chat are shown and editable (Schedule, Maximum number of users, etc...)
  5. The main actor can change any property
  6. To exit the live chat properties the main actor needs to save or discard the changes
  7. System updates all changes(if saved)
- **Description of possible deviations:**

  2.a The main actor discards all change

  2.b A property is set/changed and saved

  1. The course page is updated and saved

  2.c The schedule is set/changed

  1. The Google Calendar API Provider is updated

2. The users enrolled in the course receive a notification about the change

UC15 - Add User

- **Main participant:** System administrator
- **Goal:** Add new user
- **Participants:** Database
- **Prerequisites:** Must be administrator
- **Description of the basic course:**
  1. Administrators can create new user accounts
  2. System recognizes the creation of the new user
- **Description of possible deviation:**
  1. Administrator creates user with already existing credentials (username)

UC16 - Delete User

- **Main participant:** System administrator
- **Goal:** Delete user
- **Participants:** Database
- **Prerequisites:** Must be administrator
- **Description of the basic course:**
  1. In case of any violation the administrator can delete an user account
  2. User account is erased from the system
- **Description of possible deviation:**
  1. Administrator deletes wrong user account
  2. Administrator can't delete another administrator account

UC17 - Change Course Category

- **Main participant:** Registered User
- **Goal:** Change the category of the course
- **Participants:** Database
- **Prerequisites:** The user must be logged in (UC4 - Login) and the creator of the course (UC10 - Create Course)
- **Description of the basic course:**
  1. The main actor enters the course page

2. The main actor selects course properties
3. The creator selects Category
4. All categories present on the website are shown and the main actor can choose which category their course belongs to
5. Main Actor saves the changes
6. The system now lists the course under the new category

- **Description of possible deviations:**
  1. The creator attempts to leave without saving the changes
     - System allows user to save changes. If not saved they are discarded.

UC18 - Delete Course

- **Main participant:** Course Owner
- **Goal:** Remove a respective course
- **Participants:** Database
- **Prerequisites:** The owner must be logged in (UC4 - Login) whit the public profile and the course should belong to him
- **Description of the basic course:**
  1. Main Actor accesses his list of courses
  2. The Main Actor selects the course page that he wants to delete
  3. The Main Actor selects the option to delete his course
  4. The course in no longer existent in the system
- **Description of possible deviations:**

  1. The creator attempts to leave without saving the changes
     - System allows user to save changes. If not saved they are discarded.

Use case diagrams

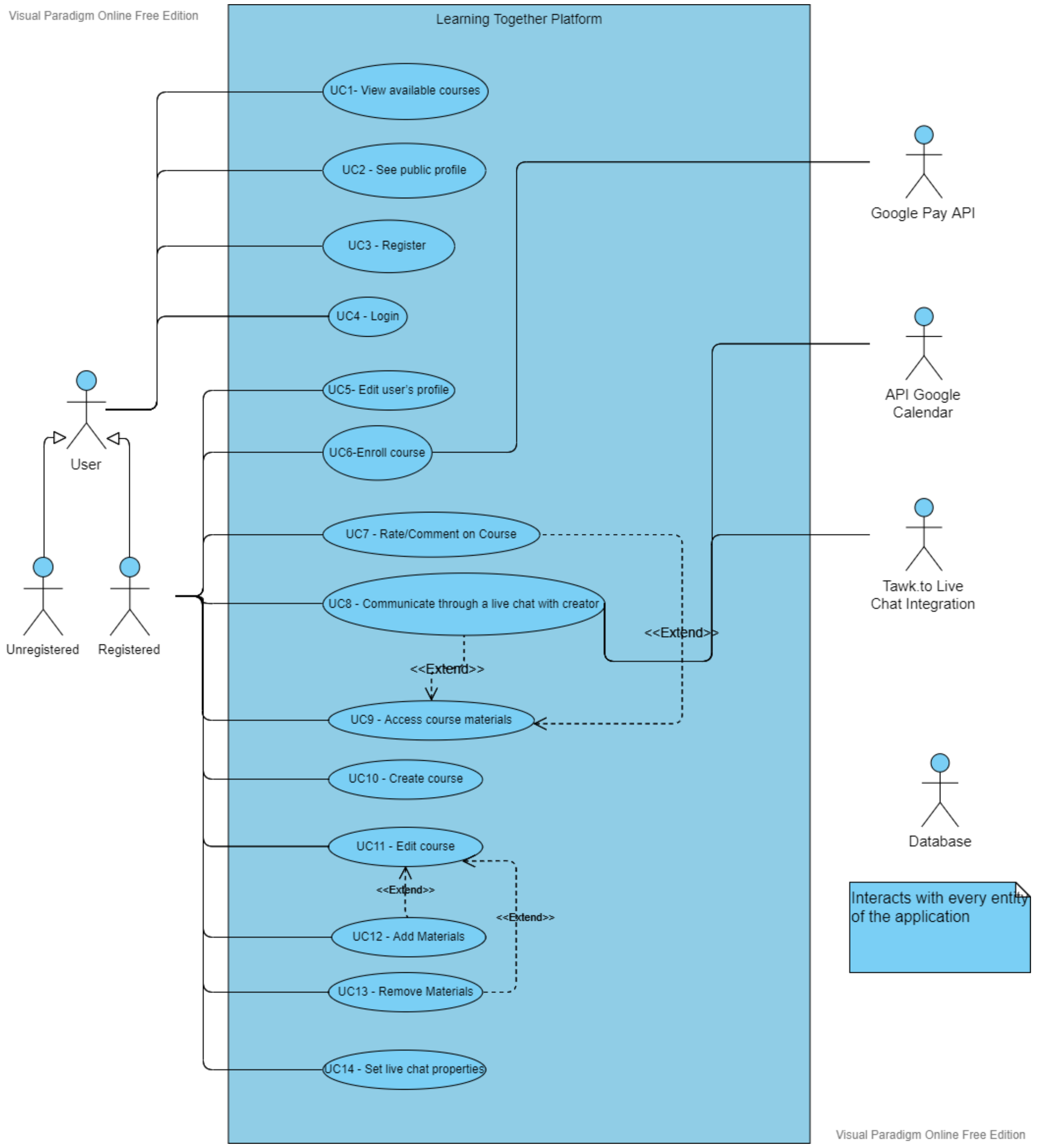
Visual Paradigm Online Free Edition
Learning Together Platform
UC1- View available courses
UC2 - See public profile
UC3 - Register
UC4 - Login
UC5- Edit user's profile
UC6-Enroll course
UC7 - Rate/Comment on Course
UC8 - Communicate through a live chat with creator
<<Extend>>
<<Extend>>
UC9 - Access course materials
UC10 - Create course
UC11 - Edit course
<<Extend>>
<<Extend>>
UC12 - Add Materials
UC13 - Remove Materials
UC14 - Set live chat properties
User
Unregistered
Registered
Google Pay API
API Google Calendar
Tawk.to Live Chat Integration
Database
Interacts with every entity of the application
Visual Paradigm Online Free Edition

Figure 3.1: Use case diagram no.1

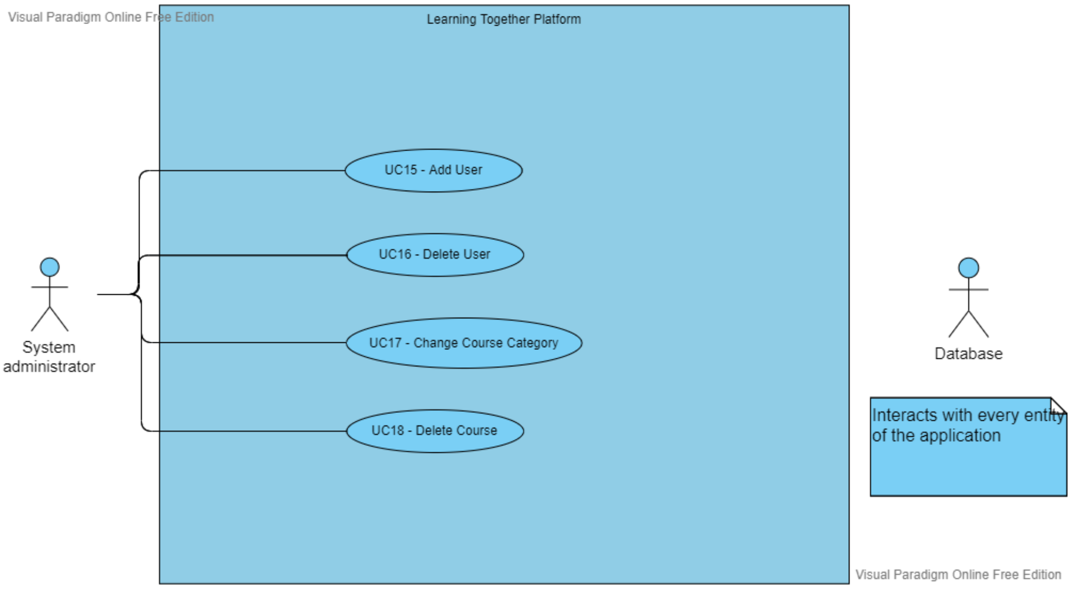
Visual Paradigm Online Free Edition
Learning Together Platform
UC15 - Add User
UC16 - Delete User
UC17 - Change Course Category
UC18 - Delete Course
System administrator
Database
Interacts with every entity of the application
Visual Paradigm Online Free Edition

Figure 3.2: Use case diagram no.2

3.1.2 Sequence diagrams

The figure 3.3 represents the sequence diagram for the action of registering an user. The user select Register and after this the web application gives them the register form, where they need to put their name, surname, email, phone number, username and password. The system verifies if the e-mail/username already exists in Database. In this case the web application returns a warning message. After the user puts a username/email valid then the system will verify if the email, phone number and password are in accordance with the conditions. If this data is not valid, then the website returns a warning message, otherwise the user is successfully registered.

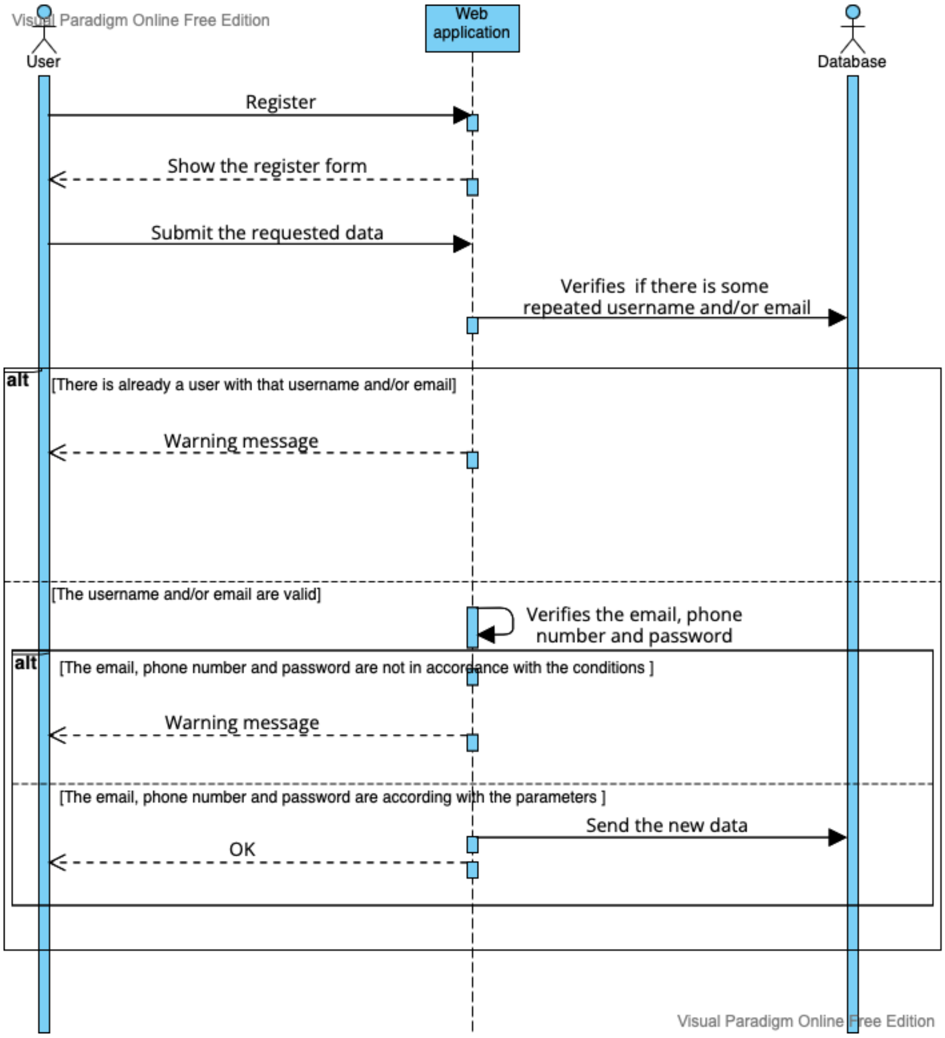
Visual Paradigm Online Free Edition
User
Web application
Database
Register
Show the register form
Submit the requested data
Verifies if there is some repeated username and/or email
alt
[There is already a user with that username and/or email]
Warning message
[The username and/or email are valid]
Verifies the email, phone number and password
alt
[The email, phone number and password are not in accordance with the conditions]
Warning message
[The email, phone number and password are according with the parameters]
Send the new data
OK
Visual Paradigm Online Free Edition

Figure 3.3: Register User Sequence Diagram

The diagram in figure 3.4 represents the enroll course action. First the system will verify if the user exists and retrieve their information from the data base, in case the search is successful. Then in case of the course being paid the system searches for the user's payment information, sending an error message in case of failure. In case of success, it informs the user that they are enrolled in the course. If the course doesn't require payment the system informs the user that they are enrolled in the course.

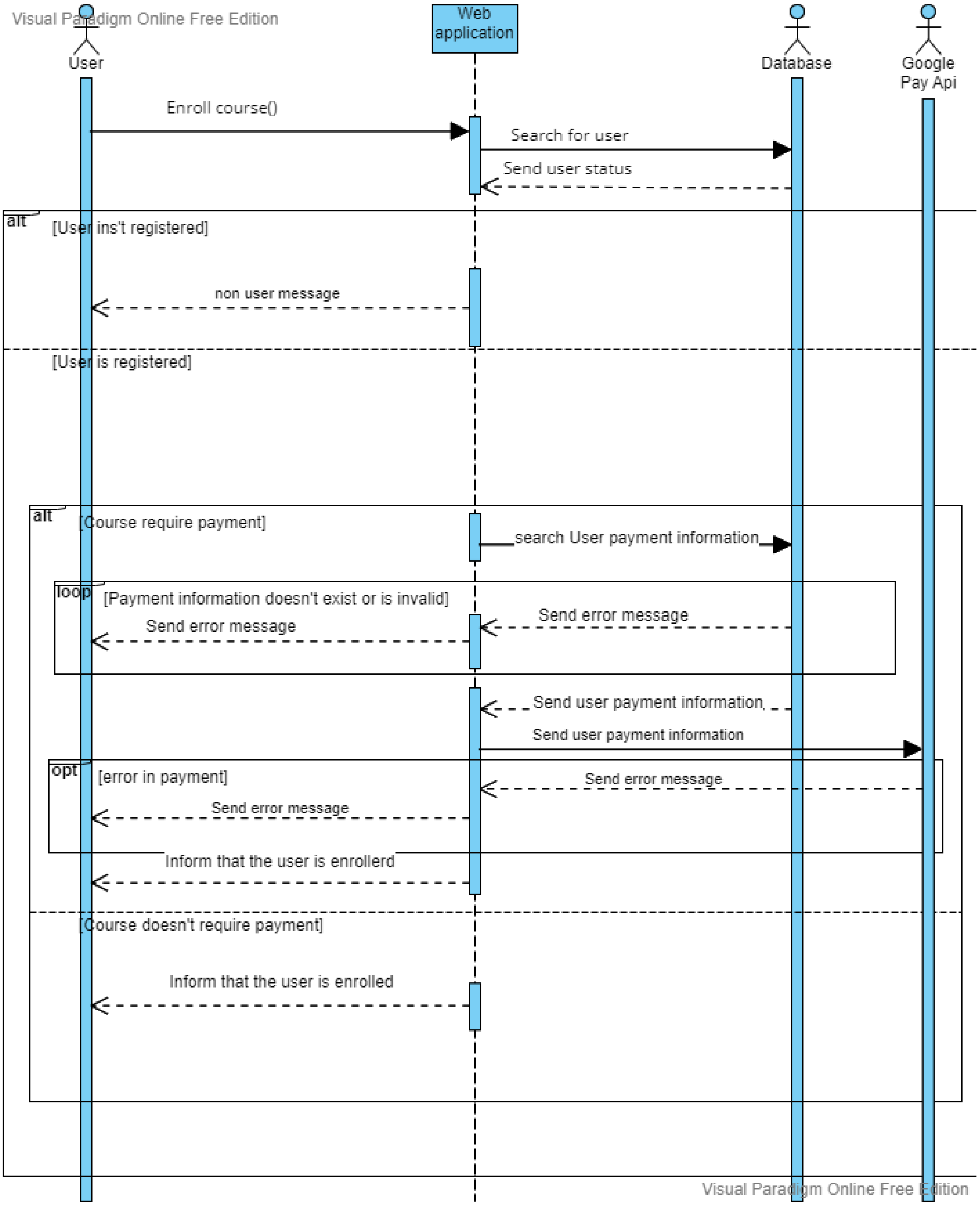
Visual Paradigm Online Free Edition
User
Web application
Database
Google Pay Api
Enroll course()
Search for user
Send user status
alt
[User ins't registered]
non user message
[User is registered]
alt
[Course require payment]
search User payment information
loop
[Payment information doesn't exist or is invalid]
Send error message
Send error message
Send user payment information
Send user payment information
opt
[error in payment]
Send error message
Send error message
Inform that the user is enrollerd
[Course doesn't require payment]
Inform that the user is enrolled
Visual Paradigm Online Free Edition

Figure 3.4: Enroll course Sequence Diagram

The figure 3.5 represents the sequence diagram for setting the Live Chat Properties. The user starts by selecting the course, followed by Definitions, and finally Live Chat. The user can then set and change all of the Live Chat properties which are saved temporarily in the web application. At the end the user can either save the changes made, updating the database with the new settings for the Live Chat or discard them, returning back to how they were at the start. One way or another it's shown to the user a descriptive message about their action and the page course is loaded on the screen.

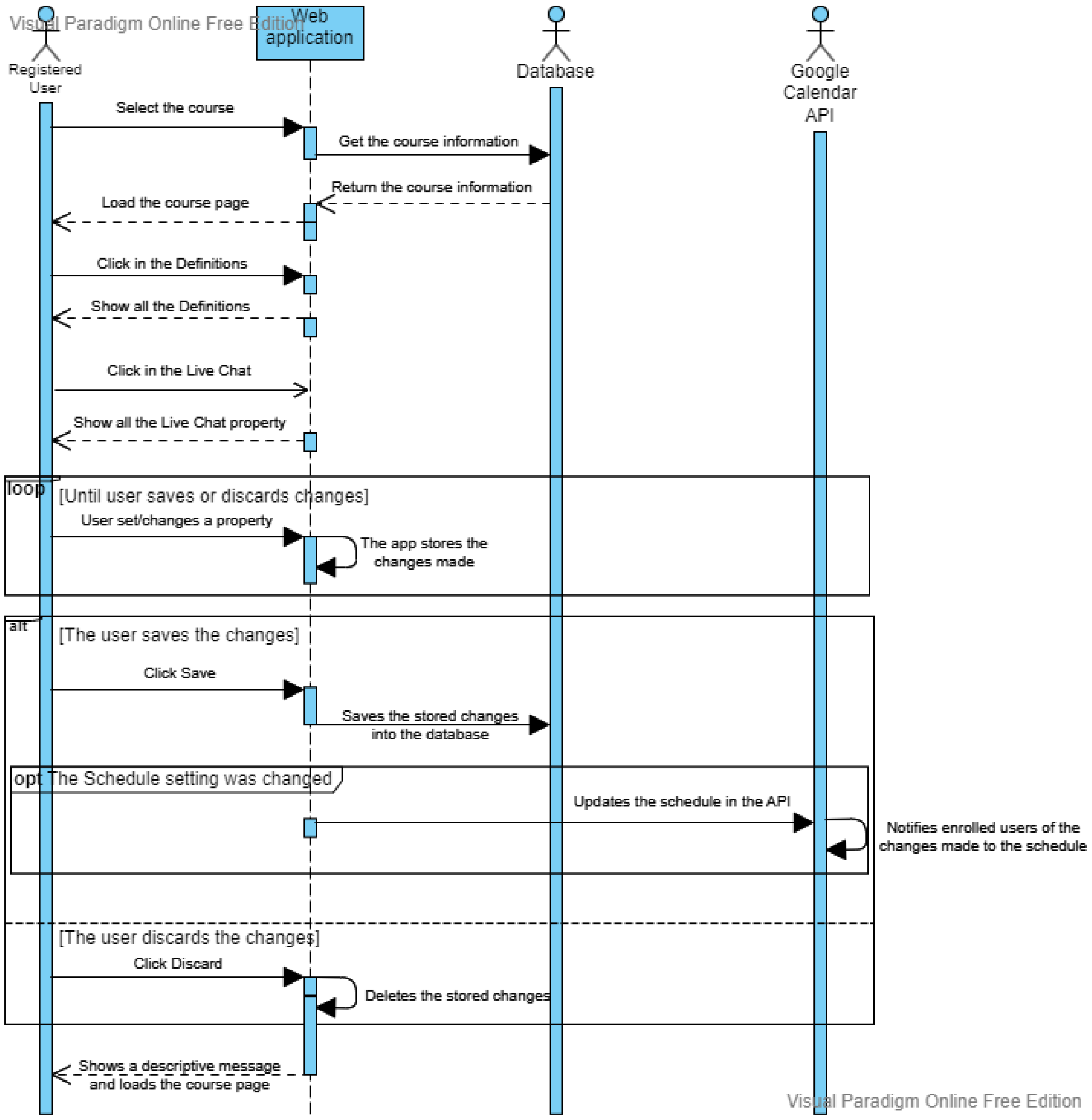
Registered User
Web application
Database
Google Calendar API
Select the course
Get the course information
Return the course information
Load the course page
Click in the Definitions
Show all the Definitions
Click in the Live Chat
Show all the Live Chat property
loop
[Until user saves or discards changes]
User set/changes a property
The app stores the changes made
alt
[The user saves the changes]
Click Save
Saves the stored changes into the database
opt The Schedule setting was changed
Updates the schedule in the API
Notifies enrolled users of the changes made to the schedule
[The user discards the changes]
Click Discard
Deletes the stored changes
Shows a descriptive message and loads the course page
Visual Paradigm Online Free Edition

Figure 3.5: Change live chat properties sequence diagram

The diagram in figure 3.6 represents the use case create course. If a registered user wants to create a course, first he choose the option to create a course and through a form the user fill it with the information about the course (name, category,etc). If the course's name chosen already exists, when he try to confirm the course creation the system will show the problem to him and he will have to fill the form again.

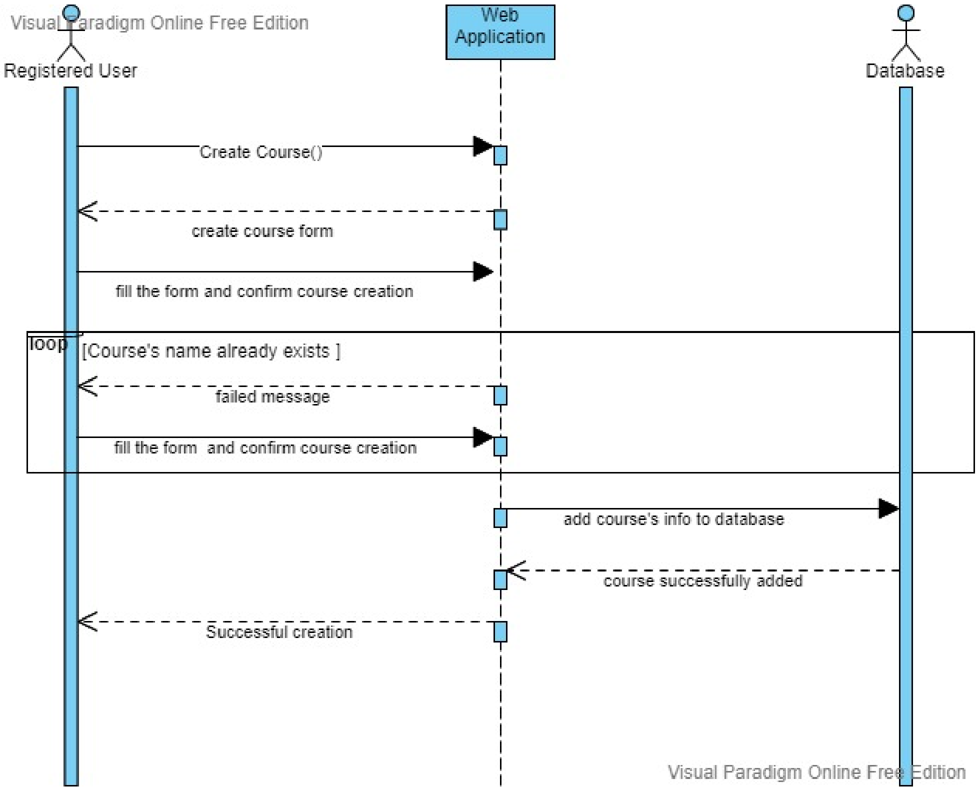
Visual Paradigm Online Free Edition
Registered User
Web Application
Database
Create Course()
create course form
fill the form and confirm course creation
loop
[Course's name already exists]
failed message
fill the form and confirm course creation
add course's info to database
course successfully added
Successful creation
Visual Paradigm Online Free Edition

Figure 3.6: Create Course Sequence Diagram

3.2 Other requirements

The website will be accessible from most browsers and their mobile variants. It will be responsive and comfortable to use on mobile devices in various sizes. Sensitive information such as passwords will be encrypted. There will be token-based authentication.

4. System architecture and design

Our web application has a separation between the implementation of frontend and backend Even though they are separated, they are both built around the Python Django Framework (as of the 1st revision). The backend uses Python while the frontend revolves around HTML, CSS and Javascript. They communicate and are connected via a REST API.

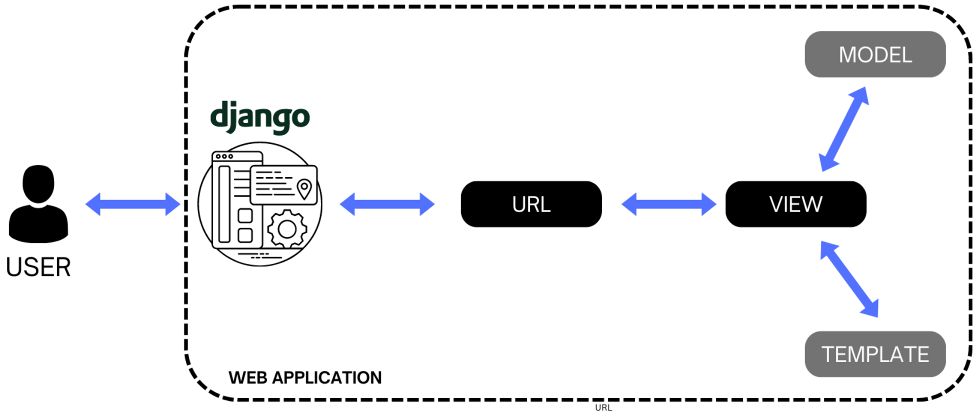

Figure 4.1: Django's Model View Template diagram

In our application, a Model-View-Template architecture is used, being generally represented on the diagram above. This architecture allows it to be separated in 3 parts, each with it's functionality. Model, which represents the logical data structure behind the whole application(as of 1st revision: a SQLite Database); View, being what the user sees when the browser renders a web page, in this case the HTML/CSS/Javascript files and Template, which consists of the static parts of HTML output and how it will be displayed. This architecture allows content to be separated according to it's use in the whole system, alllowing for a more efficient and dynamic development of code.

4.1 Database

At the time of the first revision, a SQLite database is being used for local development, with later plans of migrating to a server PostgreSQL database.

4.1.1 Table description

Tables in the database. First column is the name of the entry, followed by the type and a brief description. For every table primary key is highlighted green and foreign keys blue.

User		
id	INT	User's ID

Admin		
id	INT	Admin's ID
userId	INT	User's ID

Profile		
id	INT	Profile's ID
userId	INT	User's ID associated with the profile
username	VARCHAR	Profile's username
name	VARCHAR	Profile's name
surname	VARCHAR	Profile's surname
avatar	INT	Profile's avatar
createdDate	DATE	Profile's creation date

Private		
id	INT	Private profile's ID
profileId	INT	Profile's ID
email	VARCHAR	Private profile's email
password	VARCHAR	Private profile's password

PaymentDetails		
id	INT	Payment Details' ID
privateId	INT	Private profile's ID
cardNumber	VARCHAR	The card's number
expirationMonth	INT	The card's expiration month
expirationYear	INT	The card's expiration year
cvv	INT	The card's CVV

Category		
id	INT	Category's ID
category	VARCHAR	Category label

CategoriesLiked		
id	INT	Categories Liked's ID
privateId	INT	Private profile's ID
category	INT	Category's ID

CoursesMade		
id	INT	Courses Made's ID

Continued on next page

Continued from previous page

CoursesMade		
profileId	INT	Profile's ID
courseId	INT	Course's ID

Course		
id	INT	Course's ID
liveChatId	INT	Live Chat's ID
categoryId	INT	Category's ID
name	VARCHAR	Course's name
averageMasterTime	INT	Course's average time to master in minutes
price	DECIMAL	Course's price

Rating		
id	INT	Rating's ID
userId	INT	User's ID
courseId	INT	Course's ID
comment	VARCHAR	Rating's comment
rating	INT	Rating's rating (0-5)

TeachingUnit		
id	INT	Teaching Unit's ID
courseId	INT	Course's ID
description	VARCHAR	Teaching Unit's description

Material		
id	INT	Material's ID
unitId	INT	Teaching Unit's ID

Written		
id	INT	Written's ID
materialId	INT	Material's ID
content	VARCHAR	Written's content
length	INT	Written's number of characters

Photo		
id	INT	Photo's ID
materialId	INT	Material's ID
width	INT	Photo's width
height	INT	Photo's height
label	VARCHAR	Photo's label

Audio		
id	INT	Audio's ID
materialId	INT	Material's ID
time	INT	Audio's time in seconds

Video		
id	INT	Video's ID
materialId	INT	Material's ID

Continued on next page

Continued from previous page

Video		
resolution	VARCHAR	Video's resolution
time	INT	Video's time in seconds

LiveChat		
id	INT	Live Chat's ID
courseId	INT	Course's ID
maxParticipants	INT	Live Chat's max number of participants

Schedule		
id	INT	Schedule's ID
liveChatId	INT	Live Chat's ID
date	DATE	Schedule's date
time	TIME	Schedule's time

Participants		
id	INT	Participants' ID
userId	INT	User's ID
liveChatId	INT	Live Chat's ID

4.1.2 Database diagram

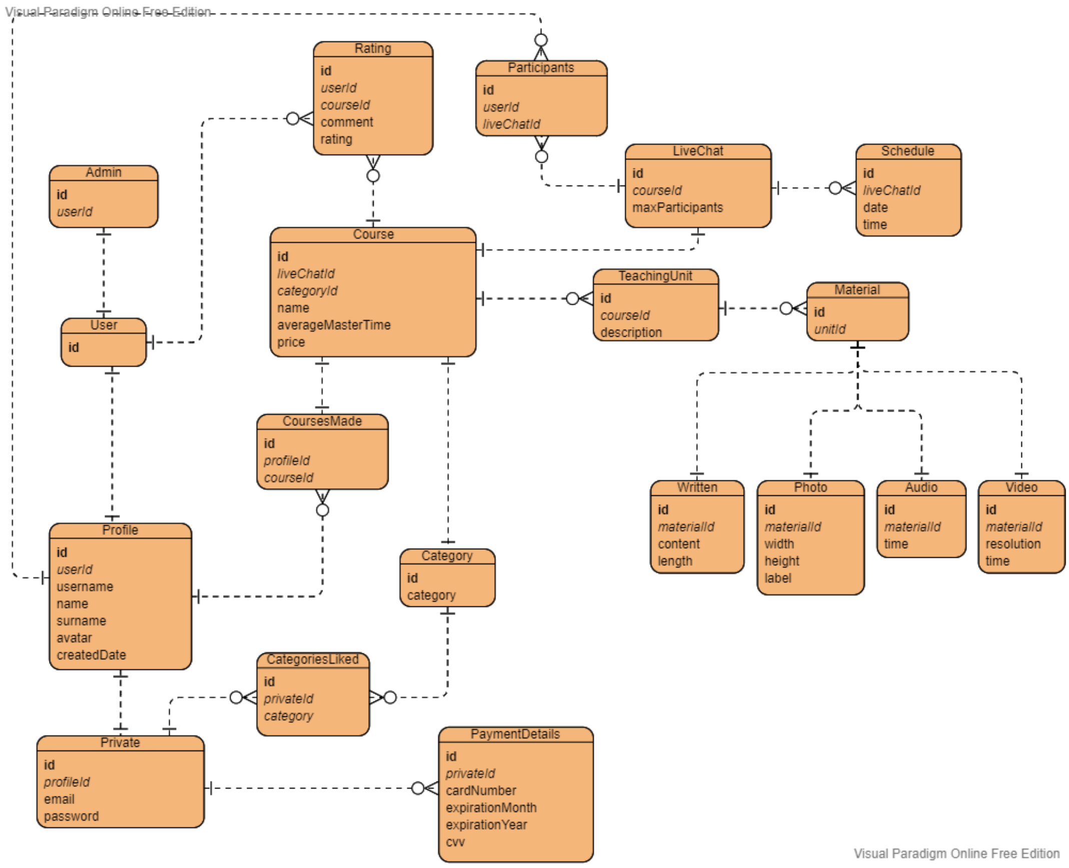

Figure 4.2: ER Database diagram

4.2 Class diagram

The diagram below shows main relationships between objects in our system.

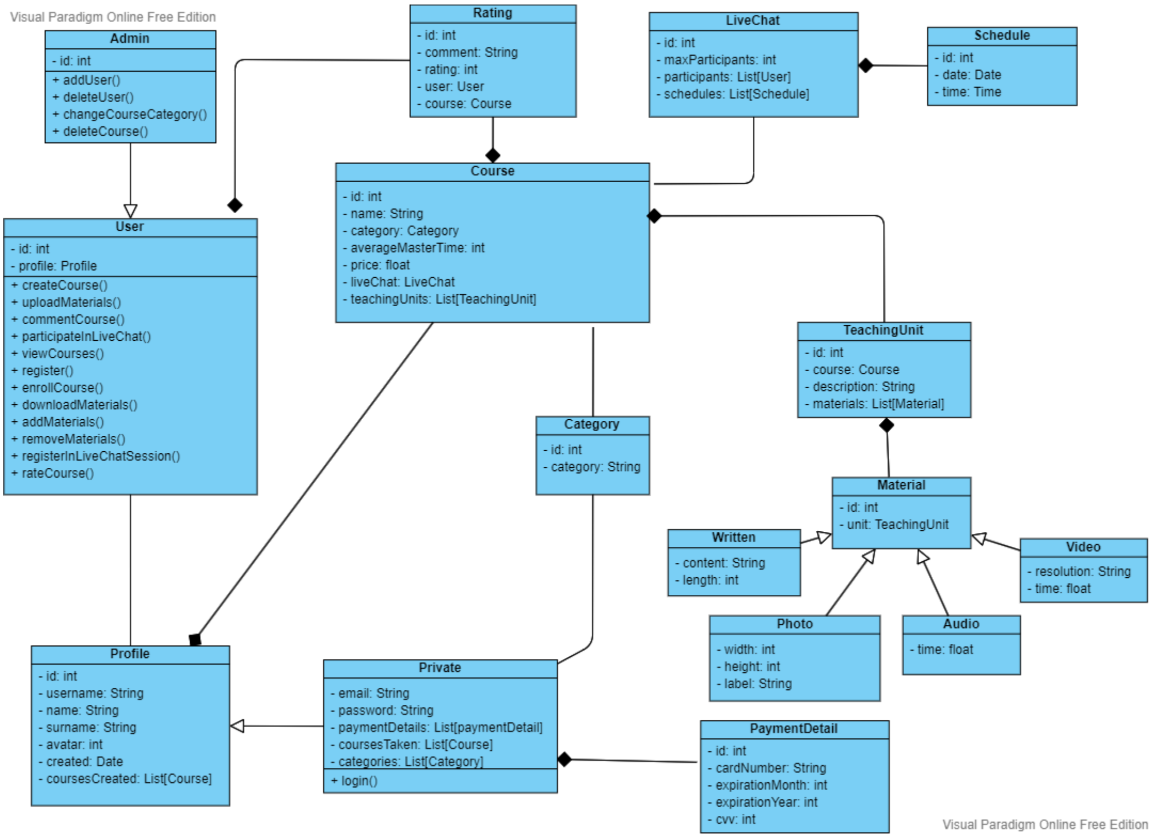

Figure 4.3: General functionalities class diagram (1st revision)

part of the 2nd revision

During the second submission of the project, the class diagram and descriptions must correspond to the actual state of implementation.

4.3 State machine diagram

part of the 2nd revision

It is necessary to attach a state diagram and describe it. One state diagram showing the ***significant part of the functionality*** *of the system is sufficient. For example, user interface states and the flow of use of some key functionality are a significant part of the system, and registration and login are not.*

4.4 Activity diagram

part of the 2nd revision

It is necessary to enclose a attach of activities with a corresponding description. The activity diagram should show a significant part of the system.

4.5 Component diagram

part of the 2nd revision

A component diagram with the accompanying description must be attached. The component diagram should show the structure of the whole application.

5. Implementation and user interface

5.1 Tools and technologies

part of the 2nd revision

List in detail all technologies and tools used in the development of documentation and applications. Briefly describe them, and state their meaning and place of application. For each of the listed tools and technology it is necessary to ***specify internet link*** *where you can download or learn more about them.*

5.2 Software testing

part of the 2nd revision

In this chapter it is necessary to describe the implementation of testing of implemented functionalities at the level of components and at the level of the whole system with the presentation of selected test cases. Students should examine core functionality and boundary conditions.

5.2.1 Component testing

It is necessary to conduct unit testing on classes that implement basic functionalities. Develop a ***minimum 6 test cases*** *in which regular cases, boundary conditions, and exception throwing will be examined. It is also desirable to create a test case that uses functionalities that are not implemented. It is necessary to enclose the source code of all exam cases and a presentation of the results of the exam in the development environment (passing / failing the exam).*

5.2.2 System testing

The system test should be performed and described using the Selenium footnote `https://www.seleniumhq.org/` *framework. Develop a* ***minimum 4 test cases*** *that will examine regular cases, boundary conditions, and call functionality that is not implemented / cause an error to see how the system responds when something is not fully realized. The test case should consist of an input (eg username and password), the expected output or result, the test step and the output or result obtained.*

The creation of test cases using the Selenium framework can be performed using one of the following two tools:

- *browser extension* ***Selenium IDE*** *- recording user actions for automatic exam repetition*
- ***Selenium WebDriver*** *- support for writing exams in Java, PHP languages using a special programming interface*

Details on the use of the Selenium tool will be presented in a special lecture during the semester.

5.3 Deployment diagram

part of the 2nd revision

You need to insert a ***specification*** *layout diagram and describe it. It is possible to insert an instance layout diagram instead of a specification layout diagram, provided that this diagram better describes some important part of the system.*

5.4 Deployment instructions

part of the 2nd revision

In this chapter it is necessary to give instructions for deployment of the realized application. For example, for web applications, describe the process by which the source code leads to a fully set up database and server that responds to user queries. For a mobile application, the process by which the application is built and placed on one of the stores. For a desktop application, the process by which an application is installed on a computer. If mobile and desktop applications communicate with the server and / or database, describe the procedure for setting them up. When creating instructions, it is recommended that ***highlight installation steps using hints*** *and use* ***screenshots*** *as much as possible to make the instructions clear and easy to follow.*

The completed application must be running on a publicly available server. Students are encouraged to use one of the following free services: Amazon AWS, Microsoft Azure or Heroku. Mobile apps should be released on the F-Droid, Google Play or Amazon App Store.

6. Conclusion and future work

part of the 2nd revision

In this chapter it is necessary to write a review of the time of project assignment, what technical challenges have been identified, whether they have been solved or how they could be solved, what knowledge was acquired during project development, what knowledge would be especially needed for faster and better project implementation. and what would be the prospects for continuing work in the project team.

It is necessary to accurately list the functionalities that are not implemented in the realized application.

References

Continuous updating

List all references and literature that helped in the realization of the project.

1. Oblikovanje programske potpore, FER ZEMRIS, `http://www.fer.hr/predmet/opp`

2. I. Sommerville, ”Software engineering”, 8th ed, Addison Wesley, 2007.

3. T.C.Lethbridge, R.Langaniere, ”Object-Oriented Software Engineering”, 2nd ed. McGraw-Hill, 2005.

4. I. Marsic, Software engineering book“, Department of Electrical and Computer Engineering, Rutgers University, `http://www.ece.rutgers.edu/~marsic/books/SE`

5. The Unified Modeling Language, `https://www.uml-diagrams.org/`

6. Astah Community, `http://astah.net/editions/uml-new`

Index of figures

Appendix: Group activity

Meeting log

Continuous updating

It's necessary to frequently update the meeting log according to the template.

1. Meeting
   - Date: October 23, 2022
   - Attendees: José Matos, Francisco Antunes, Lourenço Carvalho, Maria Inês Costa, Rui Guimarães
   - Meeting subjects:
     - General project discussion and organization
     - Functional and non-functional requirements
2. Meeting
   - Date: October 27, 2022
   - Attendees: José Matos, Francisco Antunes, Lourenço Carvalho, Maria Inês Costa, Rui Guimarães
   - Meeting subjects:
     - Technologies and architecture discussion
     - Planning of diagram development
3. Meeting
   - Date: November 17, 2022
   - Attendees: José Matos, Francisco Antunes, Lourenço Carvalho, Maria Inês Costa, Rui Guimarães
   - Meeting subjects:
     - Final diagram discussion
     - Final group revision before first version deadline

Activity table

Continuous updating

	José Matos	Francisco Antunes	Lourenço Carvalho	Maria Inês Costa	Rui Guimarães		
Project management	1.25	0	0.25	0	0		
Project task description	0.25	0.25	0.25	0.25	0.25		
Functional requirements	1	1	1	1	1		
Individual use case description	1.5	1	1	1	1		
Use case diagram	1	1	1	1	1		
Sequence diagram	0	0.5	0.5	0.5	1		
Other requirements description	0.25	0.25	0.25	0.25	0.25		
System architecture and design	0.75	0	0	0	0		
Database	0	0	1	0.25	0		
Class diagram	0	0	0.75	0	0		
State diagram							
Activity diagram							
Components diagram							
Used technologies and tools							
Solution testing							
Layout diagram							
Deployment instructions							
Meeting log	0.25	0	0	0	0		
Conclusion and future work							

Continued on next page

Continued from previous page

	José Matos	**Francisco Antunes**	**Lourenço Carvalho**	**Maria Inês Costa**	**Rui Guimarães**		
References							
Docker Integration	2	0	0	0	0		
Backend for Login and Registration	1.5	0	0	0	0		
Frontend for Login and Registration	2	0	0	0	0		
Database creation	0.25	0	0	0	0		
Connecting to the database	0.25	0	0	0	0		

Change log diagrams

part of the second revision

Import generated change log diagrams from GitLab to this chapter. Diagrams can be reached at GitLab at Repository/Contributors.