
OUR CONFERENCE MANAGEMENT SYSTEM

CATALIN-DACIAN PAP, NAGY ROLAND, PAPA CRISTIAN,
ELENA QUINTES, NOELIA IGLESIAS, CATALIN-COSTEL PAMFIL

TABLE OF CONTENTS

1.	Introduction to our Conference Management System	page 2
2.	Conference Management schedule:	
a.	Phase 1	page 3
b.	Phase 2	page 3, 4
c.	Phase 3	page 4
3.	Used technologies	
a.	Django	REST framework page 4
b.	Vue.js	page 4
c.	W3.CSS	page 4
d.	Custom	js Middleware page 5
4.	System models:	
a.	Use case diagram	page 5
b.	Scenarios:	
i.	Actions that the admin can do.....	page 6
ii.	Actions that the author can do.....	page 7
iii.	Actions that the chair can do.....	page 8
iv.	PC Member	page 9
v.	Participant paying the fee to the correspondent session.....	

	page 9
vi.	User registering to the application.....	
	page 10	
vii.	Actions that a reviewer can do.....	
	page 10	
viii.	User logging to the application.....	
	page 11	
c.	Class diagram	
	page
	12	

CMS is the acronym for Content Management System. As its name suggests, it is a system that allows us to manage content, allowing editors to change any text, image, video or similar, in a comfortable way and without having to deal with the code.

In general terms, a CMS would allow managing content in a digital medium and for the particular case at hand, a CMS would allow managing the content of a website. In other words, a CMS is a tool that allows an editor to create, classify and publish any type of information on a web page. Generally, CMS works against a database, so the editor simply updates a database, including new information or editing the existing one.

Currently most of the pages we visit are published under a CMS, since it allows us to maintain the content in an agile way and at the same time allows anyone to dedicate themselves to this task, regardless of their technical knowledge.

In our Conference Management System, the user first if doesn't have an account he can register and then start to manage himself. Firstly, he is introduced to the upcoming conferences that may be of his interest and so he can get a ticket for it. The conference is described and where it is going to take place and time, also for the fee the user will have to pay. Moreover, if he wants to propose a paper he can do it just by submitting a proposal with the correspondent information of it and uploading the document.

The consumer will find in the upper-left corner of the web the different parts of the application he can access. The "Home" one is the one described above where the user can see the close conferences.

When he enters the "Dashboard" screen, he will be able to see in the "My Conferences" tag the meetings he is going to assist either as a Listener or as a Steering. And in the "My Papers" tag he will find out all the papers he has uploaded that are proposed or that are in revision.

Under the papers section, the user can access the "Timeline" section in which he can see the schedule for the conferences.

The last item he can access is the "Add Conference" in which the user will be able to create and organize his own conferences, he can settle a description, location, time, fee and proposals for it.

When the users finish to organize his conferences and papers he can sign out and exit the app.

Conference Management schedule:

PHASE 1

Week 2 (3 – 7 March)

- Initial timeline draft

Week 3 (8 – 14 March)

- Decide what languages and frameworks we will be using
- Create github team and repository

Week 4 (15 – 21 March)

- Diagrams (Use case, Sequential, Class)
- Project codebase groundwork
- Test Django api which allows user registration
- Implement a login page from scratch, and one using a javascript frontend framework to decide which one will be used down the line.

PHASE 2

Week 5 (22 – 28 March)

- Implement basic entities expose CRUD functionality in the api
- Expand account creation and login functionality after the frontend method was decided

Week 6 (29 March – 4 April)

- Establish precise entity relationships and roles

Week 7 (5 – 11 April)

- Add ability to users with “steering committee” role to create conferences
- Show information about upcoming conferences on the homepage of users

Week 8 (12 – 18 April)

- Groundwork for paper submission and review phases

Week 9 (19 – 25 April)

- Paper submission phase: allow users to submit papers, and modify already submitted papers.

Week 10 (26 April – 2 May)

- Paper reviewing phase (first iteration): functionality for PC members to assign papers to reviewers.

Week 11 (10 – 16 May)

- Paper reviewing phase (second iteration): functionality for paper scoring and review submission
- Paper presentation phase: presentation sections, and participations

PHASE 3

Week 12 (17 – 23 May)

- Finishing touches and tweaks

Week 13 (24 – 30 May)

- Buffer zone

Week 14 (31 May – 2 June)

- Presentation day

Used technologies:

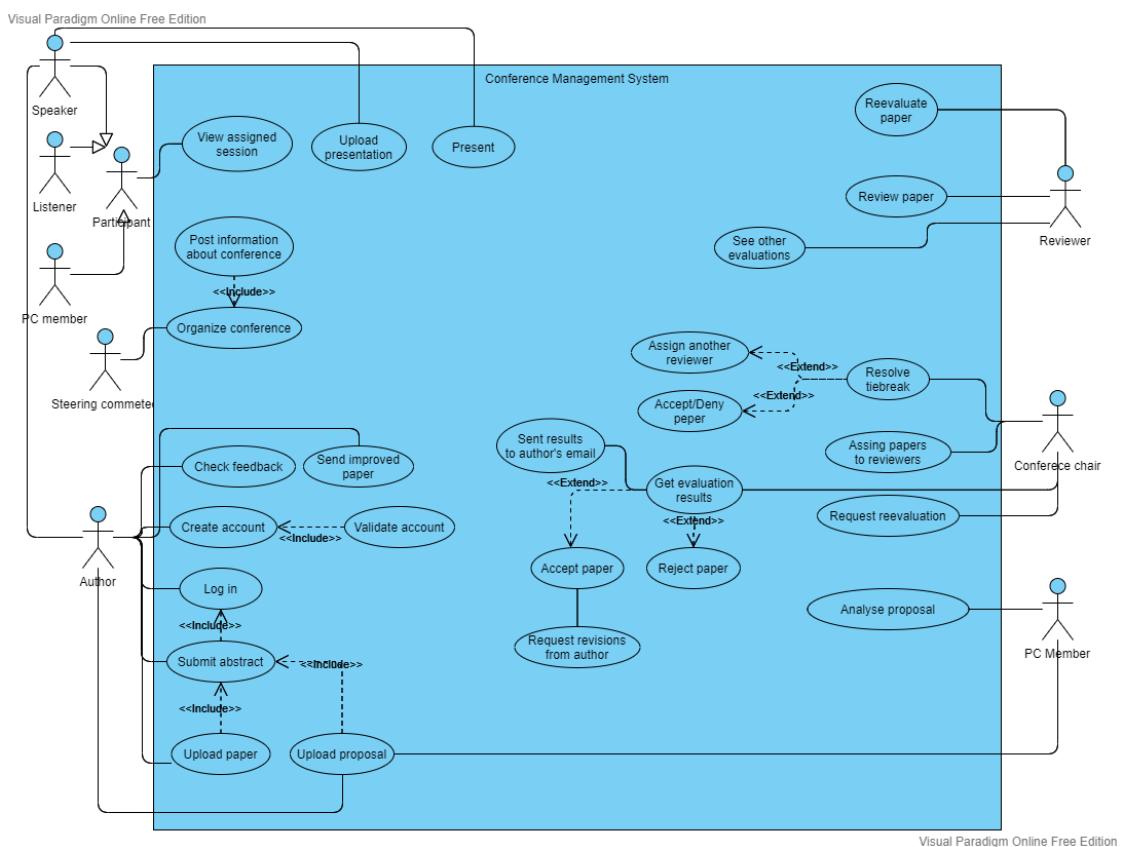
For building our Conference Management System application we have used the following technologies:

- Django REST framework, we use it for the back-end and it is a powerful and flexible toolkit for building Web APIs. Some of the advantages that DRF offers us when implementing APIs are:
 - The API is navigable from the browser which speeds up the work of developers.
 - It has integration with authentication based on OAuth1a or OAuth2.
 - It serializes data from ORM or other sources.
 - It has very good documentation and a wide community as it is open source.
 - Companies like Red Hat, Heroku or Mozilla use it.
- For the front-end we have used vuejs components, w3 CSS and custom js middleware.
 - Vue.js lets you extend HTML with HTML attributes called directives which offers functionality to HTML applications. Vue.js provides built-in directives and user defined directives.
 - W3.CSS is a modern, responsive, mobile-first CSS framework, it provides equality for all devices: desktop, laptop, tablet, mobile... W3.CSS is standard CSS only.

- Custom js middleware. Middleware is a function that executes the lifecycle method to an Express server, and utilizes the request and response cycles.

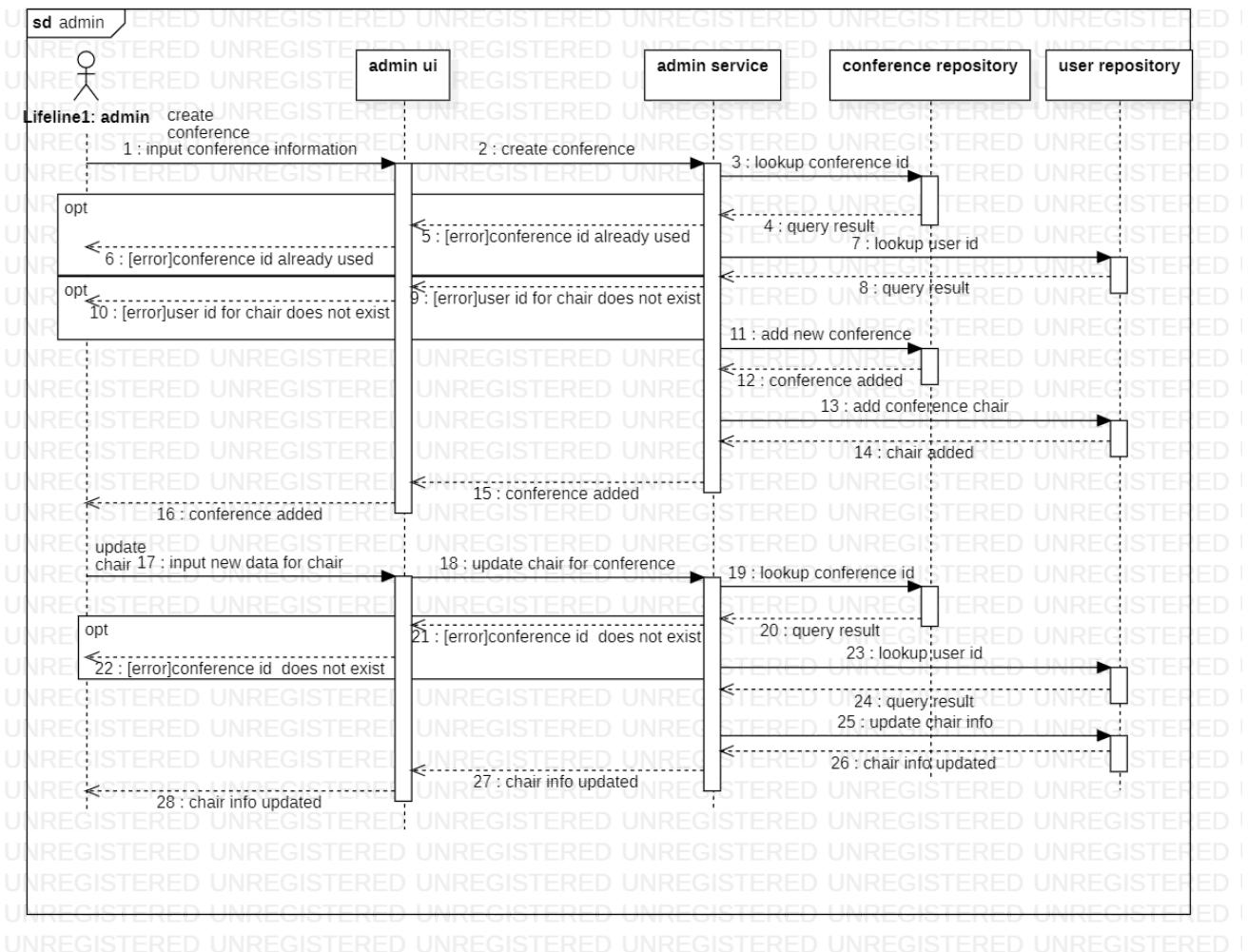
System models

1. Use case model:

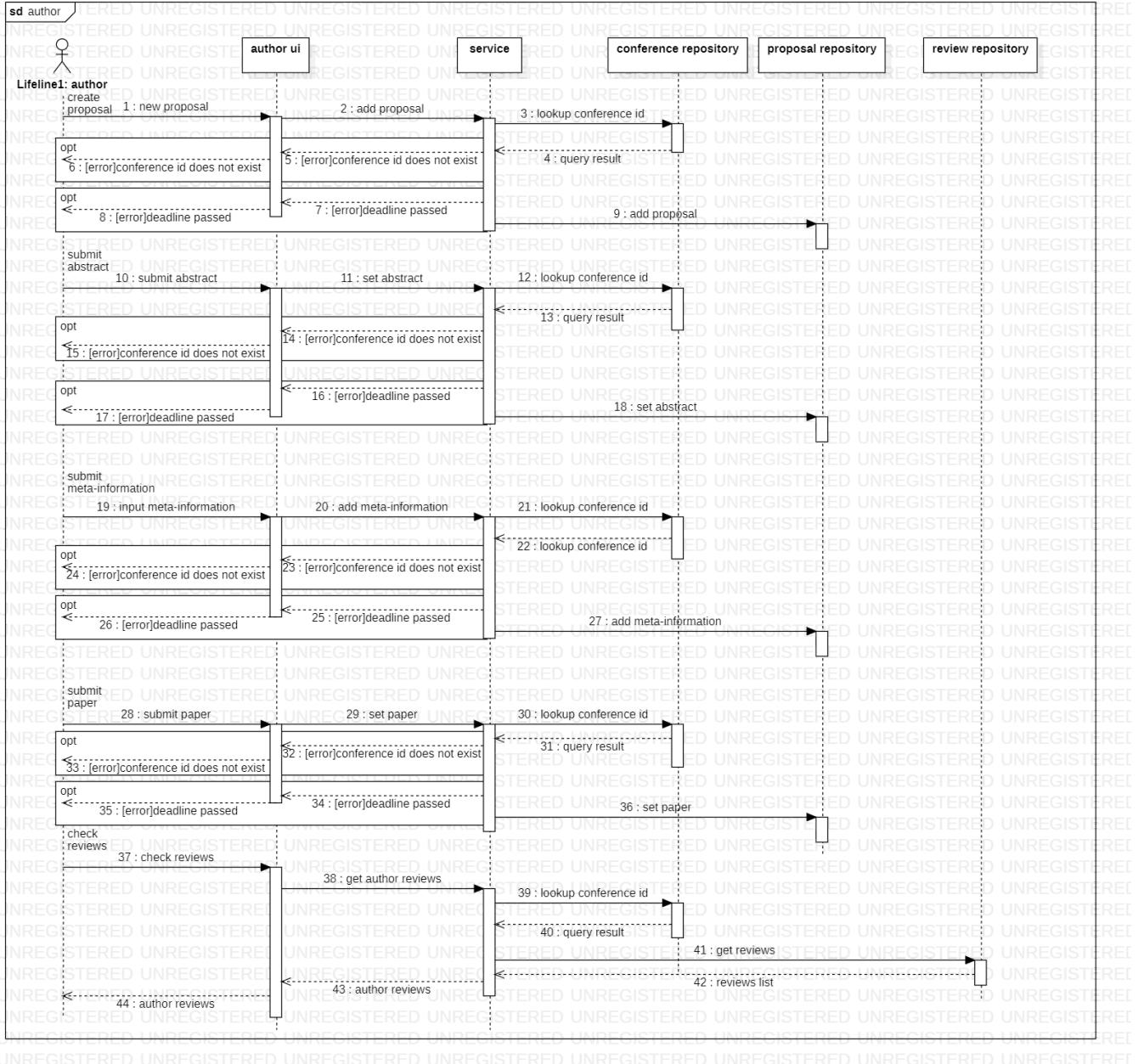


2. Scenarios

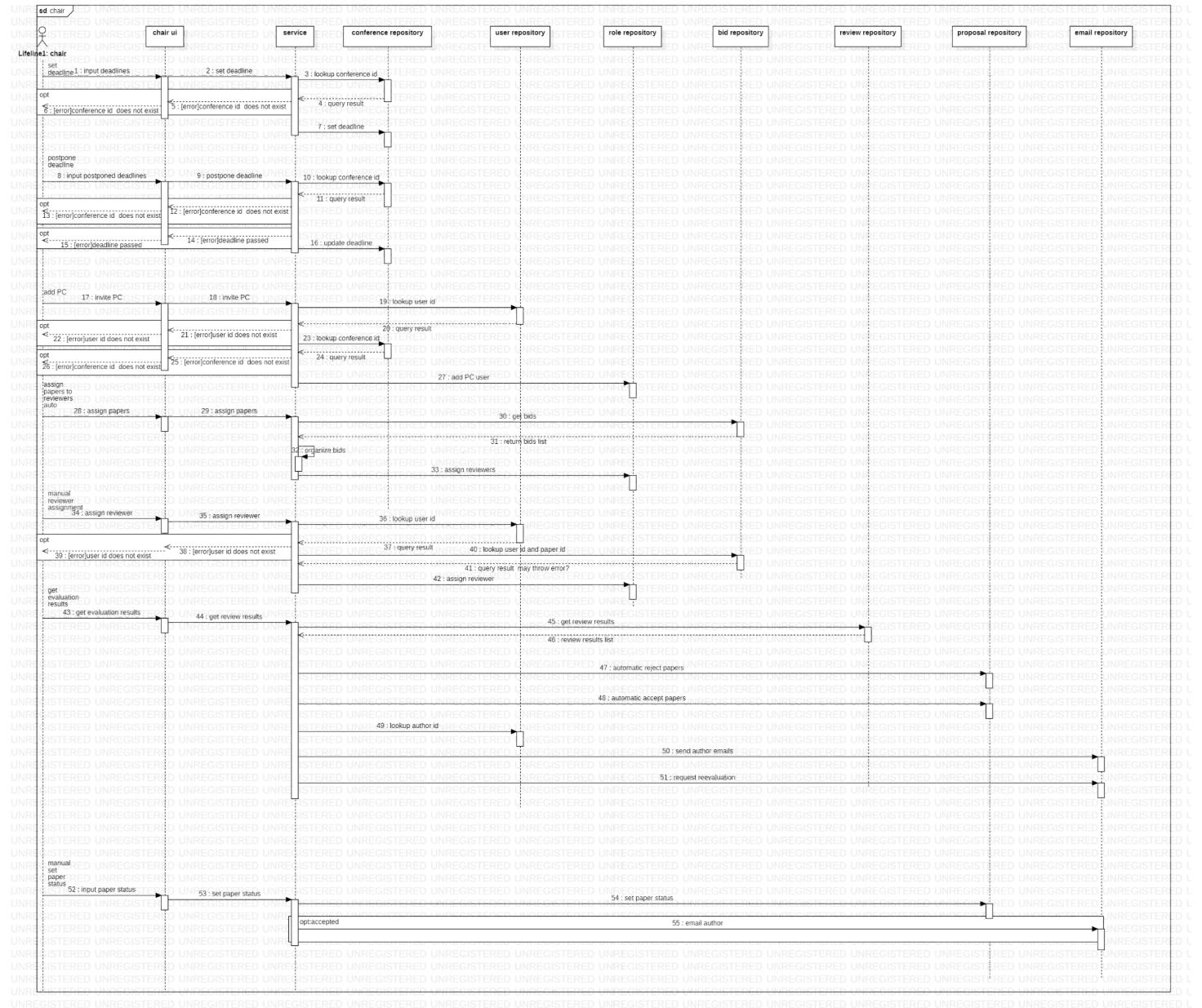
- Actions that the admin can do:



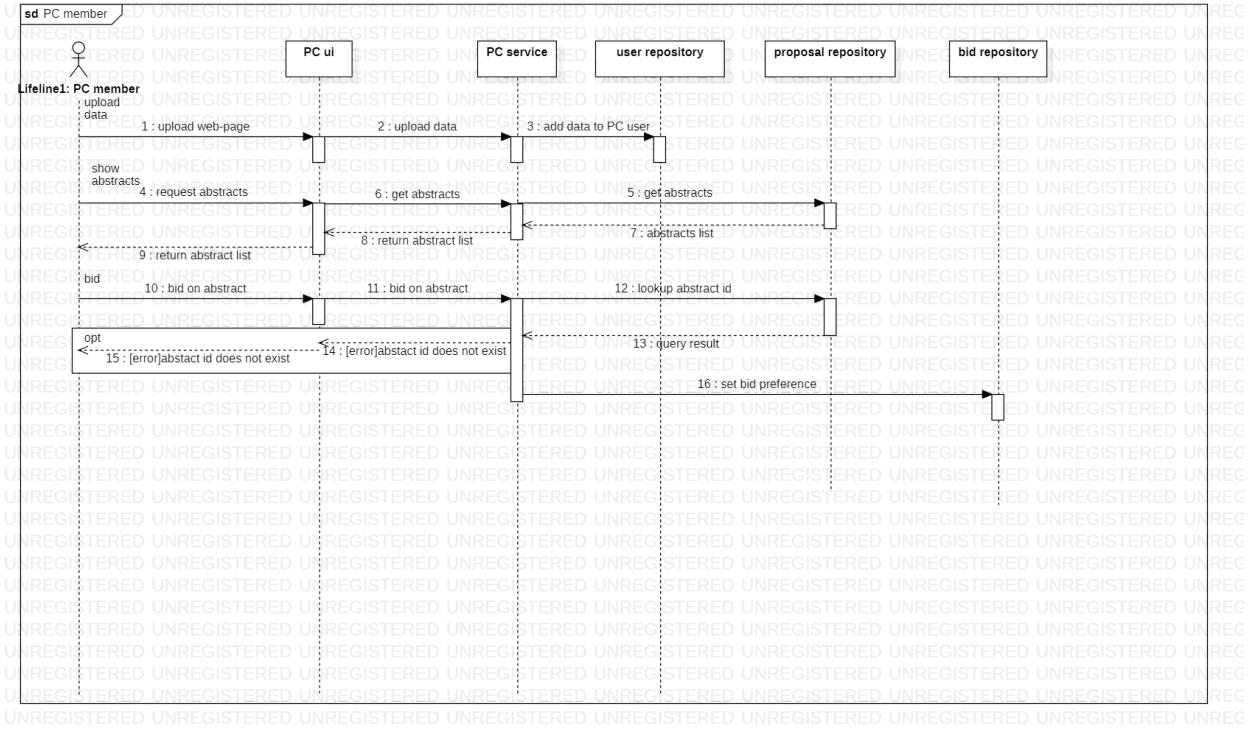
- Actions that the author can do:



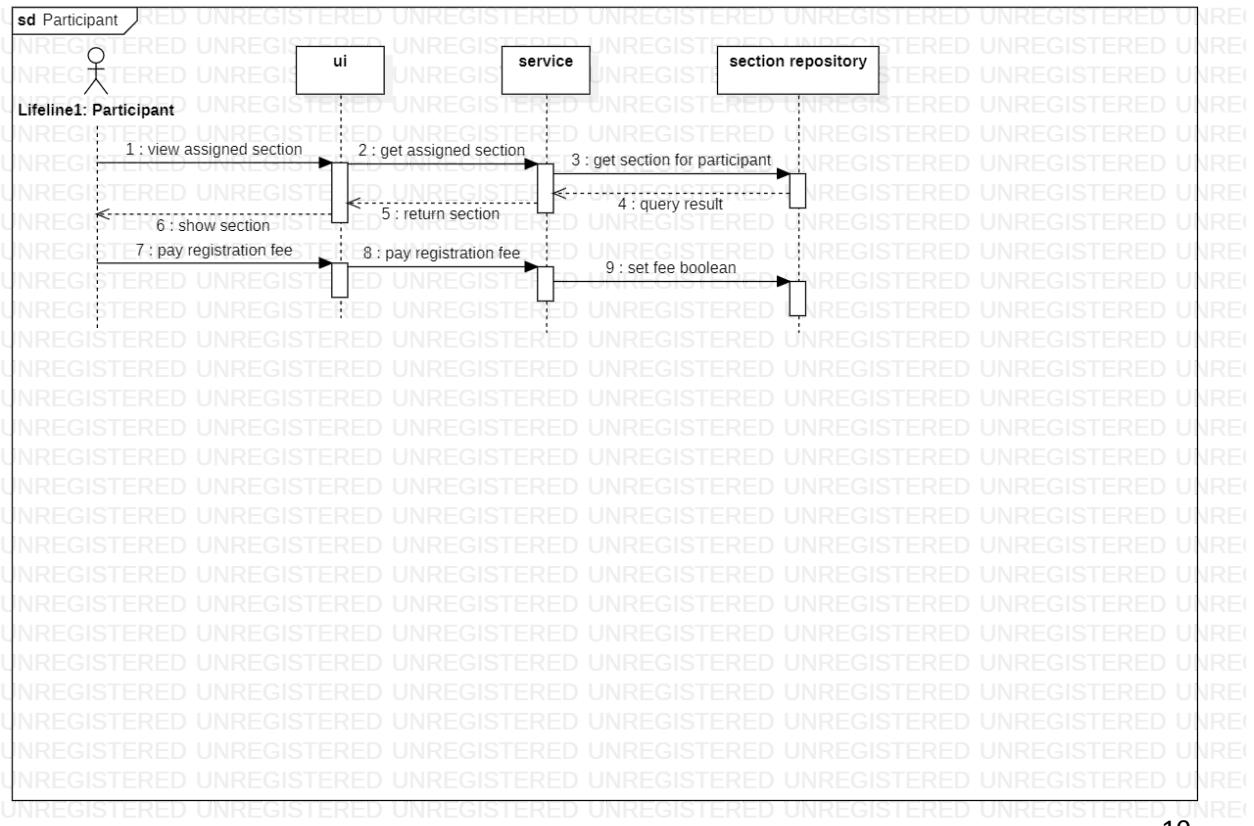
- Actions that the chair can do:



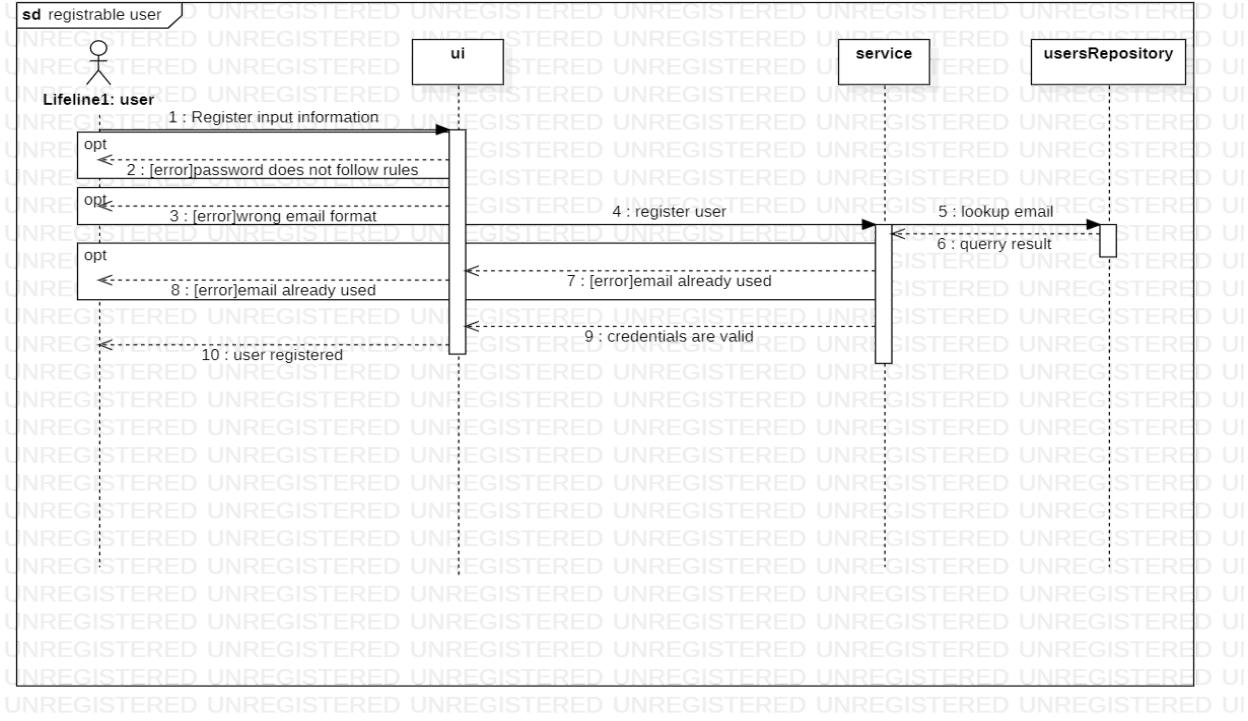
- PC Member submitting proposals:



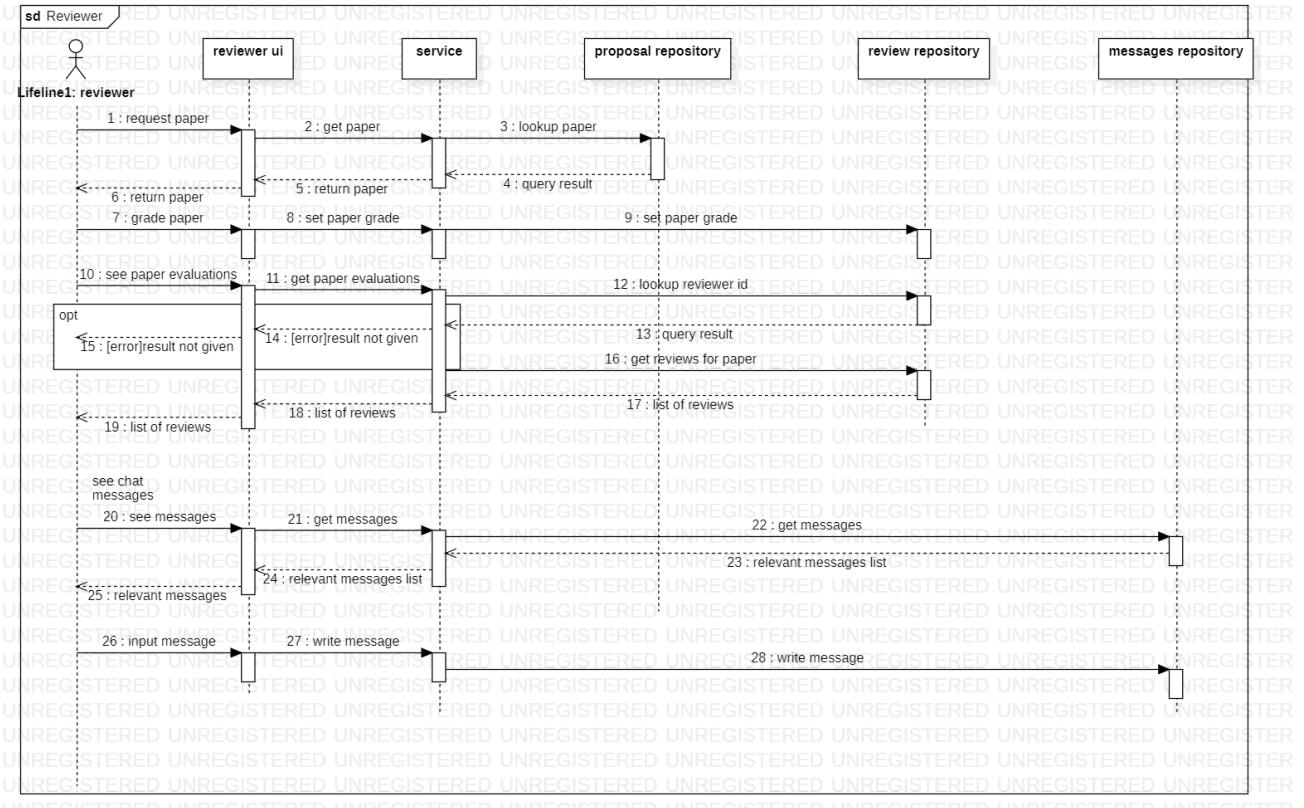
- Participant paying the fee to the correspondent session:



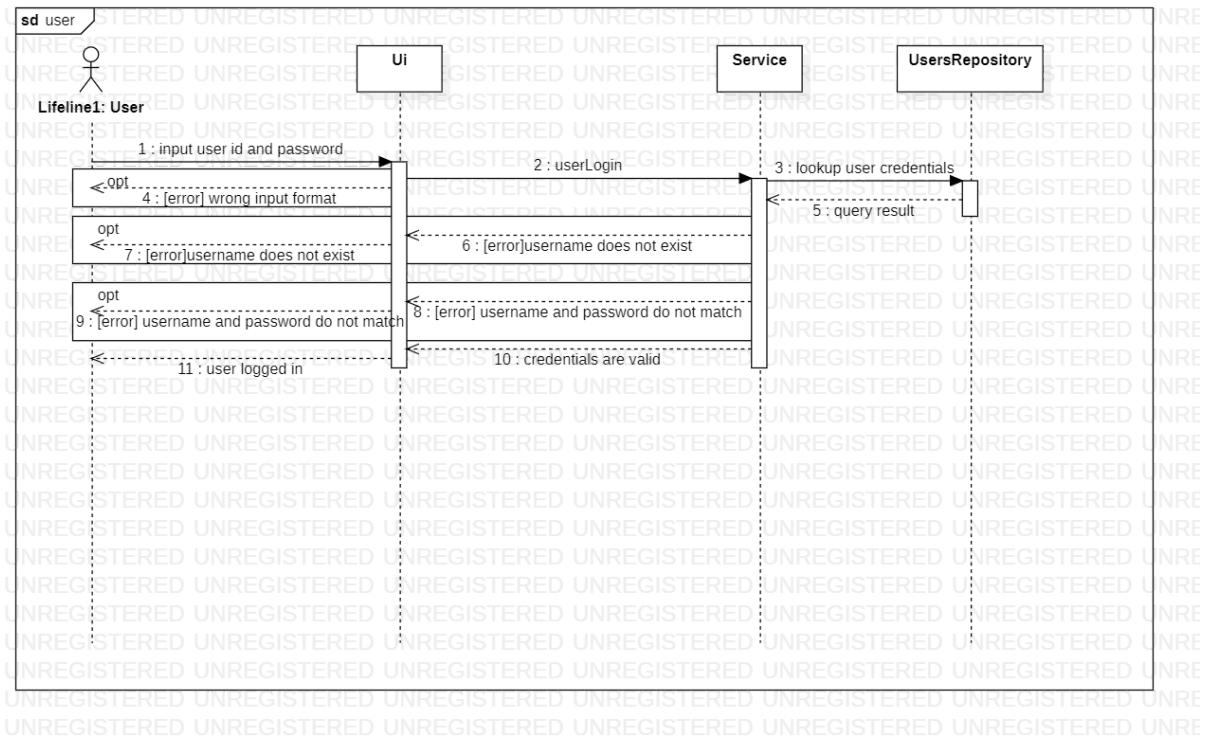
- User trying to register the application:



- Actions a reviewer can do:

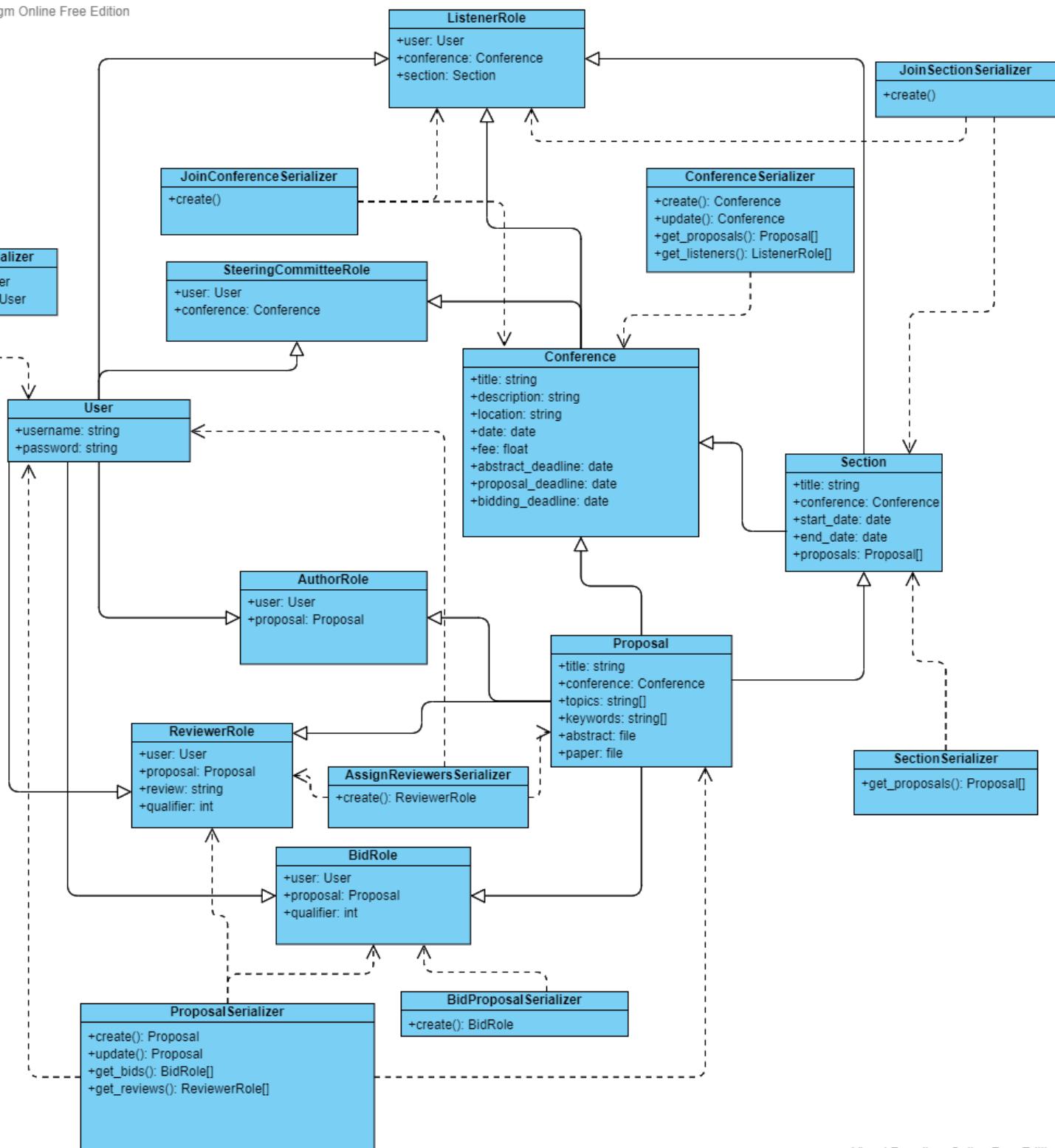


- User trying to log into the application:



3. Class diagram

Visual Paradigm Online Free Edition



Visual Paradigm Online Free Edition