**BSc (Hons) in Information Technology**

**Specializing in Software Engineering**

**Year 3 - 2021**

**SE3040 – Application Frameworks**

**Group Project**

**Draft Report**

Spark Coders

2021S1\_REG\_WD\_26

|  |  |
| --- | --- |
| Student ID | Name |
| IT19155944 | M.A.M Rishard |
| IT19170244 | E.M.P.U Ekanayake |
| IT19140544 | S.L Jayasekara |
| IT19168500 | K.M.J.S Wickramathilake |
|  |  |

1. **FUNCTIONS**
2. **Admin Management**

The Administrator monitors all the activities of the Conference Management System using a dashboard. The activities which are monitored include all the user registrations, research paper uploads, workshop proposal uploads, attendee registration payment income etc. Apart from monitoring activities, the other major responsibility of the Administrator is to review the conference details added or modified by the Editor. After the editor has made an addition or a modification, the Administrator receives a notification. They can view the modifications and additions in a separate table in the Admin Panel. The administrator can either approve the changes or reject them. Only after the Administrator has approved a certain modification of the conference details, that modification takes effect.

1. **User Management**

There are 3 types of users in the system. They are Researchers, Workshop Presenters and Attendees. Attendees can register to conference by purchasing a ticket by providing necessary contact details and making a payment. Researchers can register to the system by providing their contact details and uploading their research paper. Similary, workshop presenters can upload a workshop proposal and register to the system. Once the Research paper or Workshop proposal is submitted it is submitted to review and once they are approved the relevant user will receive a notification. Research paper presenters must pay if their papers got approved but Workshop presenters don’t have to pay. So, User management handle all the above mentioned functions in the system.

1. **Reviewer Management**

The functionality handles the reviewers in the system. Reviewers are able to register to the system and the reviewer can view all the reviews that the reviewer has reviewed. A reviewer also has the ability to view all the researchpaper uploads and workshop detail uploads. A reviewer can view the pending reviews and approve or decline the research papers and workshop proposals. Once a reviewer approves or declines a paper or a proposal a notification is sent by the system to the user.

1. **Editor Management**

Editors main task is to add and edit conference details. Editors need to login to the system before they can add, edit and delete conference details, such as time, date and venue. After adding the conference details, they are saved in the database with a pending status. After approval by the admin, the details are displayed on the website. The editor also has the ability to add and edit tickets and create a schedule.

# **REST API**

1. Admin Routes

**i. POST –** api/v1/admins

**ii. GET –** api/v1/admins

**iii. GET –** api/v1/admins/{id}

**iv. DELETE –** api/v1/admins/{id}

**v. PUT –** api/v1/admins/{id}

**vi. POST –** api/v1/auth

2. Editor Routes

**i. POST –** api/v1/editors

**ii. GET –** api/v1/editors

**iii. GET –** api/v1/editors/{id}

**iv. DELETE –** api/v1/editors/{id}

**v. PUT –** api/v1/editors/{id}

3. Reviewer Routes

**i. POST –** api/v1/reviewers

**ii. GET –** api/v1/reviewers

**iii. GET –** api/v1/reviewers/{id}

**iv. DELETE –** api/v1/reviewers/{id}

**v. PUT –** api/v1/reviewers/{id}

4. Researcher Routes

**i. POST –** api/v1/researchers

**ii. GET –** api/v1/researchers

**iii. GET –** api/v1/researchers/{id}

**iv. DELETE –** api/v1/researchers/{id}

**v. PUT –** api/v1/researchers/{id}

5. Workshop Routes

**i. POST –** api/v1/workshops

**ii. GET –** api/v1/workshops

**iii. GET –** api/v1/workshops/{id}

**iv. DELETE –** api/v1/workshops/{id}

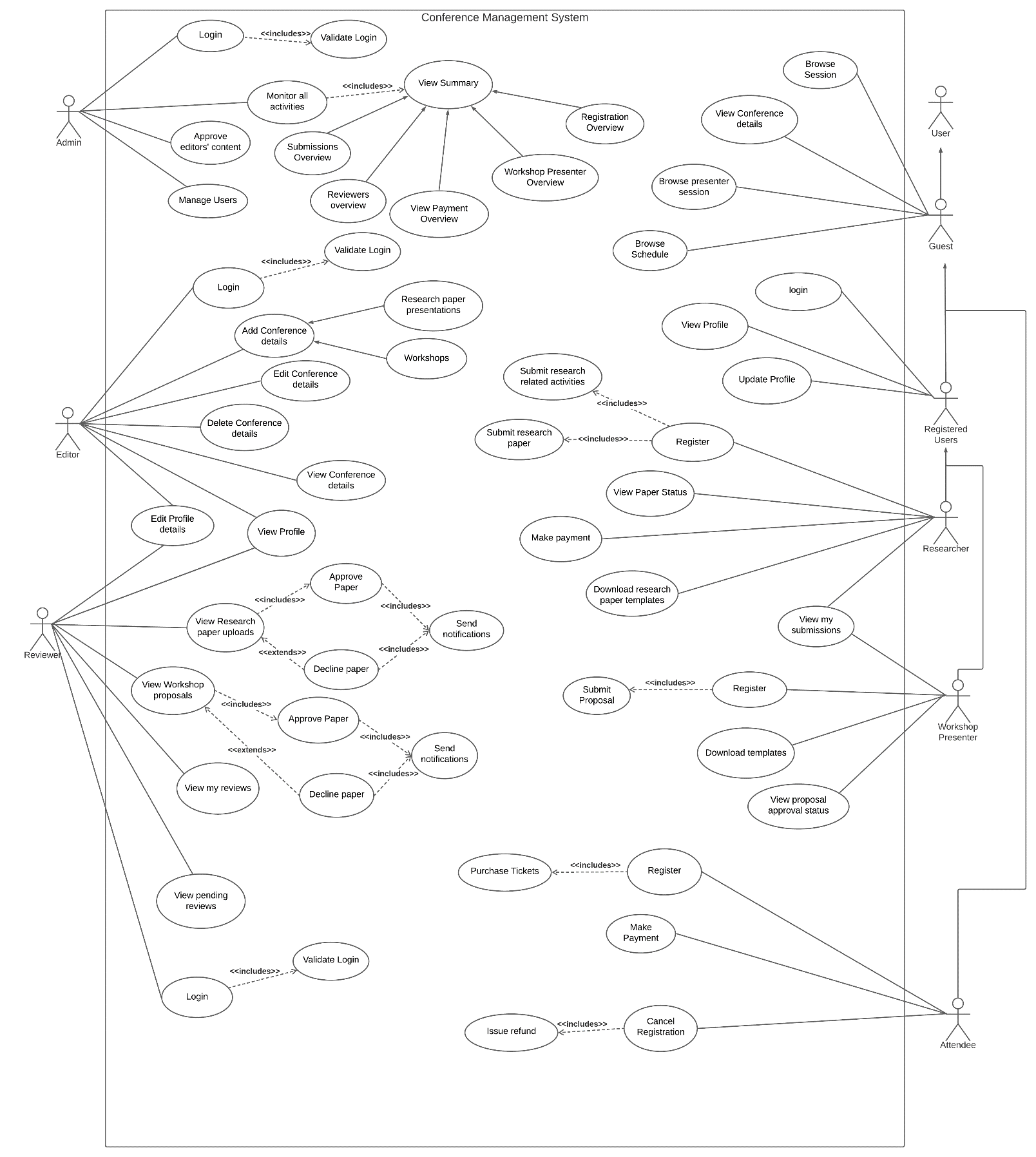
**v. PUT –** api/v1/workshops/{id}

6. Login Routes

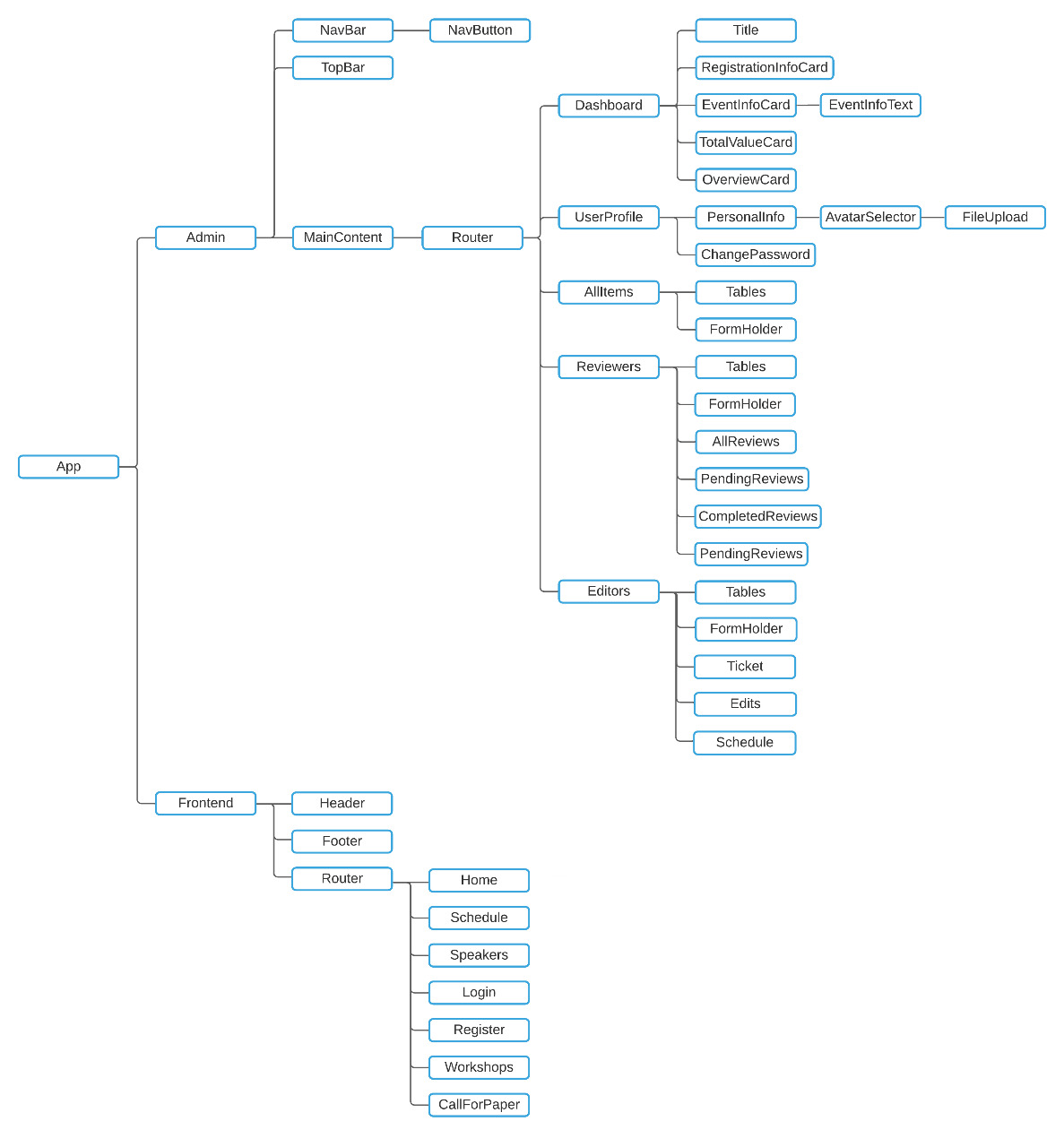
**i. POST –** api/v1/login

**ii. POST –** api/v1/login/admin

**Usecase Diagram**



**Component Tree**



**MongoDB Document Screenshots**

1. Admin

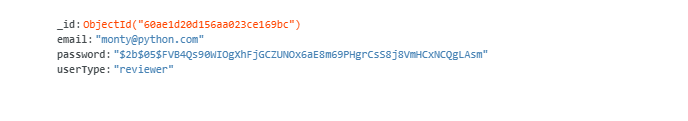


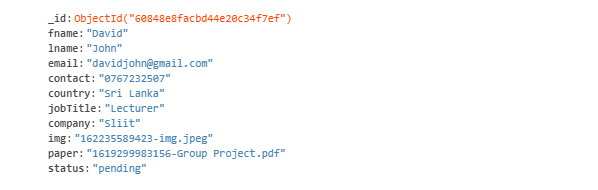
1. Reviewer



1. Editor



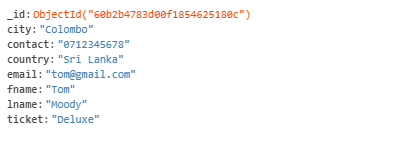
1. Login
2. Researcher



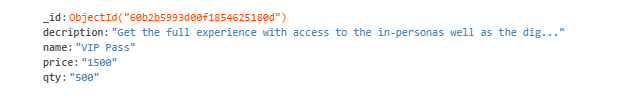
1. Workshop



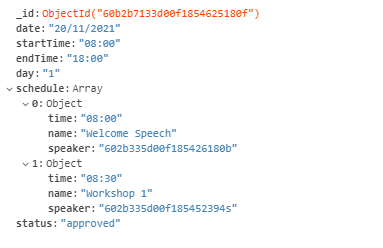
1. Attendee



1. Tickets



1. Schedule



1. General Info

