SLIIT UNI
THE KNOWLEDGE UNIVERSITY

BSc (Hons) in Information Technology

Specializing in Software Engineering

Year 3 - 2021

SE3040 – Application Frameworks

Group Project

<<Technical Report / User Guide>>

<<Group Name - Hype Codex>>

<<Group ID - 2021S1_REG_WE_24>>

<<Title - Conference Management Tool>>

Name	Registration Number	Functionality
YASAS R.M (Group leader)	**IT19133850**	Role Based Authentication and Innovation Management
Bandara M.H.M.N.D.T	**IT19152288**	Workshop management and Conference Management
Gunawardana K.H.R	**IT19215884**	Research paper management
Dhanasekara D.M.S.M	**IT19056258**	Reviewers management

Repository Link: https://github.com/manuka99/Conference-Management-Tool

---For Evaluators' use only---

Evaluator's Name:

Comments:

Group Details

Group ID	**2021S1_REG_WE_24**	
Name	**Registration Number**	**Functionality**
YASAS R.M (Group leader)	**IT19133850**	Role Based Authentication and Innovation Management
Bandara M.H.M.N.D.T	**IT19152288**	Workshop management and Conference Management
Gunawardana K.H.R	**IT19215884**	Research paper management
Dhanasekara D.M.S.M	**IT19056258**	Reviewers management

Table of Contents

1 Introduction

The functionalities of Conference management tool are developed as REST API by using Node JS and Express JS. Mongo DB is used for the database tier. React is used to developed the user interfaces of the application (MERN stack).

When a request is sent to the application, routes created using express JS will forward the request to the controller. Any operation related to database will be done through the models / schemas created using mongoose dependency. Once a request is processed a JSON response will be sent back. Environment constants will be saved as environment configuration file. Some external dependencies are used to support common tasks in the application. Below are a list of dependencies and its uses.

Passport JS – Used as an authentication middleware for Node Js.

Jsonwebtoken – To generate signed Json web tokens once the authentication credentials is matched in Node Js.

Bcrypt – To encrypt user sensitive data such as password in Node Js.

Nodemailer – To send email from Node JS

Donenv – To load environment variables in Node Js

Mongoose – Schema based solution to model application data using mongo DB in Node JS

CORS - For providing a Connect/Express middleware that can be used to enable **CORS** with various option in Node JS.

Device detector Js – To detect the user device details through useragent in request body in Node Js.

Twilio – To send SMS through twilio cloud messaging service in Node Js.

Material UI – To develop frontend user interfaces in React

Redux – To management state in React Js.

React Router V6 – For routing in React Js.

Axios – To make requests (XMLHttpRequests) from React to the Server.

2 Functionalities of the system

2.1 IT19133850 – Role Based Authentication and Innovation Management

2.1.1 Description

Conference management tool is a web application, which will be used by many users with different user roles. In order to secure data and information among user groups, a secure authentication mechanism must be used. The backend / server of the entire system will be developed using Node JS. Therefore, an authentication system based on JSON Web Token will be used. JSON Web Token are one of the secure practice or authentication method used to identify authorized users of the system. Guest users have to verify their credentials (Email and password) once and in return they will get a unique token which is allowed to access for 15 minutes. The token will have user's basic details and role details.

When registering a user and resetting a password the password entered, is encrypted (using bcrypt) and stored in the database. Even when querying we can get only the highest version of the password unless specially quarried for the real password.

User can request to reset the password then a hashed reset token will be generated and an email will be sent to the user's email address along with the reset URL. Once user visit the reset URL, server checks the validity of the reset token and allows changing the password.

There is no specific method to logout the user from application when using JWT. In order to perform logout functionality, a middleware will be created to verify authorization token of each request, if the request was verified (authorized) when the token will be saved or updated in the database. If the user logout of the application, then the current token which is saved in the database will be retrieved and it's property "isValid" is set to false and updated. To avoid user authenticating using the same token, a middleware is created to check the database if the property "isValid" of the token set to false or not. If it is set to false then the user will not be authenticated with the system. If the token is not present in database or if the property "isValid" is set to true then the jwt token will be verified. If the token was verified then the user object will be retrieved from the database and appended to the request (request.user).

Some functions must only be accessed by authorized users, therefore to verify the request is authenticated or not a middleware will be created to check if the request has a user object (request.user). In order to perform role-based authentication a similar middleware will be created in which it will accept an array of roles. In this middleware it will check if the user object in the request object include those roles, boolean response is returned.

Based on the requirements of the system, there are few user roles such as admin, editor, reviewer and user. Through the guest registration, a user will be given a role of user. Other user roles can be assigned to a user only by an admin.

Through the guest registration, user can register as a researcher, workshop presenter, innovator, attendee. After the completion of the registration, user will be given a role of user and a sub role of the type user selected in the form (researcher, workshop presenter or attendee). At the registration, users must include their basic details such as full name, email, contact number and birth of date.

Additionally, there are certain data that user must submit based on the sub role user selects. A researcher must include the research paper as pdf or word document. Workshop conductor must include a proposal containing all the necessary details about the workshop. An innovator must include a proposal containing all the necessary details about the innovator. Attendees are required to pay an amount of five hundred rupees to register for the conference. User can complete the payment through credit /debit cards, EZ cash or by Dialog Genie. Payhere payment gateway will be implemented to support the above requirement. All users except attendees, are required to be approved by the admin.

A user can register to the system as an innovator. In here user is required to submit detailed description about their findings and if the idea was approved by the admin then the user can present in the conference.

2.1.2 Rest API

Process of registering a new user

A POST request with the details (first name, last name, address, phone, date of birth, email, password, repeat password and user type) will be sent to the authentication service. The email should be unique, and the password and repeat password must match. Certain validations are checked based on the user type selected. A researcher must include the research paper as pdf or word document. Workshop conductor must include a proposal containing all the necessary details about the workshop. An innovator must include a proposal containing all the necessary details about the innovator. Attendees are required to pay an amount of five hundred rupees to register for the conference.

If the request is valid, a new user is created by adding new JSON document to the user collection in mongoDB. Simultaneously a bearer token for authentication also created (auth_token) . It will be used for the authentication of the user when login to the system . The password is converted to a hash code for better privacy using bcrypt and stored. When querying, password is not passing unless specially asked for it.

Process of login

A POST request with email and password will be sent to the authentication service. Once request is received the system check the availability of the user by searching for the email and match the passwords. If the passwords match, a sign in token will be generated by adding all details in user object except password and returned. If the user is not found, an error response with status code of 400 will be returned.

Process of password recovery

A POST request with the email will be sent to the authentication service. Verify the email provided exists in database. If the email exists a reset token is created which is expired in 10mins, and the hashed version of the token will be stored in the database. An email with the http message embedded with the reset url, email and the reset token will be sent to the provided email address. Nodemailer and sendGrid is used for this purpose. Once the email sending procedure is successfully done, success message will be generated. Otherwise, an error response with status code of 400 will be returned.

Process of resetting user password

A PUT request with a reset token, email, new password and repeat new password will be sent to the authentication service. User object will be fetched from the database from the email provided. Bycrypt is used to compare the token in the URL parameters with the hashed reset token in the user object retrieved. If the validation is success, new password will be matched with the repeat password. If passwords match, new password will be hashed and the user data will be updated. After a successful password reset, a bearer token for authentication is also generated and returned. If some error occurred during the process, an error response with status code of 400 will be returned.

Process of logging out the user

A POST request will be sent to the authentication service. Token object is retrieved from the database based on the token in the authorization header in the request. If the token exists its "isValid" property is set to false and updated else a new token object is created and saved setting it's "isValid" property to false.

Process of view all innovations

A GET request will be sent to the innovation service. An authenticated user with role admin and editor can view all innovations. If successfully validated, all the innovations submitted by the users in the DB will be fetched. In case if an error occurred, an error response with status code of 400 will be returned.

Process of view one innovation

A GET request with the innovation id will be sent to the innovation service. An authenticated user with role admin and editor can view innovations. If successfully validated, the innovation id will be validated and if there is no error, the innovation corresponding to the id will be fetched. In case if an error occurred, an error response with status code of 400 will be returned.

Process of updating an innovation

A Patch request with the innovation id and updated details will be sent to the innovation service. An authenticated user with role admin and editor can update innovations. If successfully validated, innovation details will be updated and saved in the DB. If there is error in the process, an error response with status code of 400 will be returned.

Process of deleting an innovation

A DELETE request with the innovation id will be sent to the innovation service. An authenticated user with role admin can update innovations. After the authentication, the request will be validated. Then the availability of the innovation object is checked and if available the innovation will be deleted. If there is error in the process error message will be returned.

2.1.3 Test cases

Test Case 01		
Author	User	
Summary	Register to the system as a researcher.	
Precondition	User must have a valid internet connection. User must have a valid email address.	
Step No.	**Step Action**	**Expected output**
1	User visit to the homepage.	Load the landing page.
2	User click on register button.	Navigate to user registration form.
3	User enter first name.	Text field should accept input data.
4	User enter last name.	Text field should accept input data.
5	User enter email.	Text field should accept input data.
6	User enter phone number.	Text field should accept input data.
7	User enter date of birth	Text field should accept input data.
8	User enter address	Text field should accept input data.
9	User enter password.	Text field should accept input data.
10	User enter same password in repeat password text field.	Text field should accept input data.
11	User select user type as researcher.	User type researcher is selected. Research paper upload field is visible.
12	User select research file,	File field should accept selected file.
13	User clicks on “Register” button.	Application alerts, “Your registration details were successfully saved” and navigates to the home page.

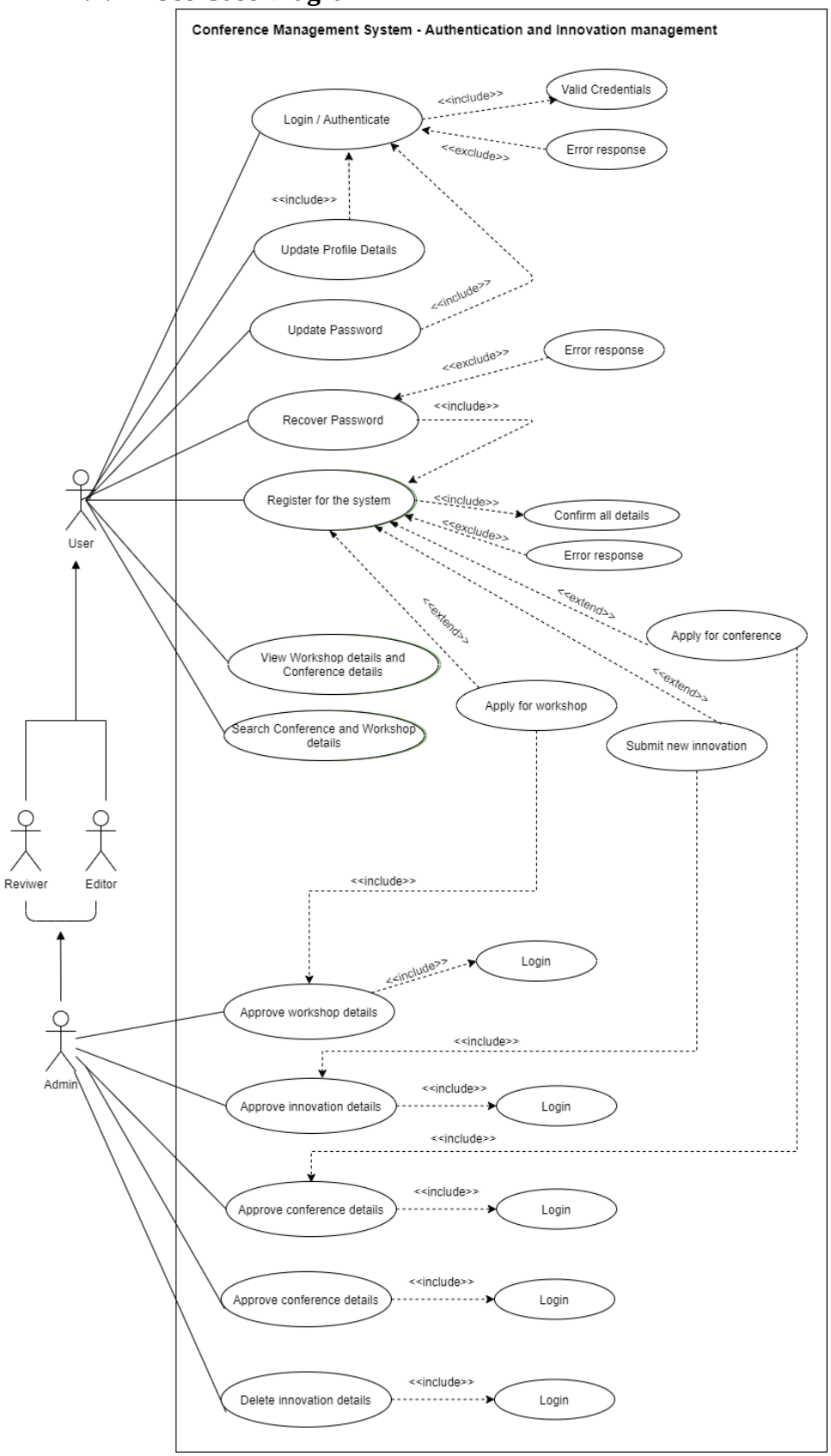
Test Case 02		
Author	User	
Summary	Register to the system as an attendee.	
Precondition	User must have a valid internet connection. User must have a valid email address. User must have a valid credit card with a balance of at least Rs. 500.00.	
Step No.	**Step Action**	**Expected output**
1	User visit to the homepage.	Load the landing page.
2	User click on register button.	Navigate to user registration form.
3	User enter first name.	Text field should accept input data.
4	User enter last name.	Text field should accept input data.
5	User enter email.	Text field should accept input data.
6	User enter phone number.	Text field should accept input data.
7	User enter date of birth	Text field should accept input data.
8	User enter address	Text field should accept input data.
9	User enter password.	Text field should accept input data.
10	User enter same password in repeat password text field.	Text field should accept input data.
11	User select user type as attendee.	Payment form is displayed.
12	User enter credit card number.	Text field should accept input data.
13	User enter CVV number.	Text field should accept input data.
14	User enter expiration date.	Text field should accept input data.
15	User clicks on “Register” button.	Application alerts, “Your registration details were successfully saved” and navigates to the home page.

Test Case 03		
Author	User	
Summary	Login to the system.	
Precondition	User must have a valid internet connection. User must have a valid registered account.	
Step No.	**Step Action**	**Expected output**
1	User visit to the homepage.	Load the landing page.
2	User click on “login” button.	Navigate the user login page.
3	User enter username.	Text field should accept input data.
4	User enter password.	Text field should accept input data.
5	User clicks on “Login” button.	Application alerts, “You are successfully login” and navigates to the home page.

Test Case 04		
Author	User	
Summary	Update user’s contact details.	
Precondition	User must have valid internet connection. User must be an authenticated user.	
Step No.	**Step Action**	**Expected output**
1	Visit home page.	Load the landing page.
2	User click on “Manage profile” button.	Navigate to the update profile page.
3	User edit first name.	Text field should accept input data.
4	User edit last name.	Text field should accept input data.
5	User edit phone number.	Text field should accept input data.
6	User edit address.	Text field should accept input data.
7	User clicks “Update contact details” button.	Application alerts, “You update details were successfully saved”.

Test Case 05		
Author	User	
Summary	Recover Password	
Precondition	User must have valid internet connection. User must have a valid registered account.	
Step No.	**Step Action**	**Expected output**
1	Visit home page.	Load the landing page.
	User click on “Recover Password” button.	Navigate to the recover password page.
2	User enter email address.	Text field should accept input data.
3	User click on “Recover” button.	Application alerts, “Reset URL has been sent to your email”.

2.1.4 Use Case Diagram



2.1.5 React Component Diagram

1. APP
    1. Homepage
        1. Header
        2. Footer
        3. Navbar
        3. Body
            1. Landing Page
                1. Hero Component
                2. Workshops Component
                3. Conferences Component
            2. Login Page
            3. Register Page
                1. Contact Form component
                2. Workshop Form component
                3. Research Form component
                4. Innovation Form component
                5. Payment Component
            3. Recover Password Page
            4. Reset Password Page
                1. Reset Password Form
            4. Manage Profile Page
                1. Contact Details Form
                2. Change Password Form
                3. Change Email Form
            5. My Innovation Page
            6. My Workshop Page
            7. My Research Page
            7. My Attendance Page
    2. Admin Dashboard
        1. Admin Header
        2. Footer
        3. Navbar
        4. Body
            1. Admin Homepage
                1. New Users component
                2. New Workshops Component
                3. New Conferences Component
                4. New Innovation Component
                5. New Attendees Component
            2. Manage Innovation Page
                1. Innovation Form component
                2. View Innovation Component
                3. View All Innovations Component

2.1.6 Mongo database document screenshot


```
_id: ObjectId("60b0c4f00b52574480d43491")
role: "user"
fname: "Manuka"
lname: "Yasas"
date_Of_birth: 1999-07-24T18:30:00.000+00:00
address: "asasasq2222222"
phone: "072-1146-092"
email: "manukayasas99@gmail.com"
password: "$2b$12$T31yG09qd8gt3Wh7WMXOouxdggF5tDQORjXz1ZJlPA2vi8woLh.lK"
createdAt: 2021-05-28T10:24:48.590+00:00
updatedAt: 2021-05-28T20:18:20.412+00:00
__v: 0
```



```
_id: ObjectId("60b38290ecd3b561e884b26e")
user_id: ObjectId("60b0c4f00b52574480d43491")
description: "This is a test description of the innovation."
name: "Smart belt"
short_description: "This is a test short description of the innovation."
category: "hardware"
status: "failed"
summary: "This is innovation violates our terms and conditions. We only accept s..."
createdAt: "2021-05-28T22:17:52.702+00:00"
updatedAt: "2021-05-28T19:26:58.702+00:00"
__v: 0
```

2.2 IT19152288 - Workshop management and Conference Management

2.2.1 Description

Workshop/Conference Management is based on user levels.Users can view conference/workshop details without login to the system. But only registered user can request for conduct workshop or conference. o registered to the system. For registration, user can add their details to the given form and submit it. For registration for conduct workshop also user need to submit given form.

Admin can approve or reject the user request for the conduct workshop. Once admin log in to the account admin can view all workshops published, workshop request, and rejected workshop. The request status will be notifying to the user via email.

Editor can edit the publish details anytime and user is responsible for the add workshop details to the system.

2.2.2 Rest API

Get a Workshop / Conference

A GET request will be sent with a workshop id as a parameter toward the authenticated service. The conferencing details will be fetching from the workshop model using the parameter send in the request. If the workshop details are fetched return successful message. If there is error it will return error message.

Get all Workshop/ Conference

A GET request will be sent through the authenticated service. After authentication of user, all the workshop/conference details in the Database will be fetch. If an error occurred during fetching an error message will be return else return successful message.

Submit registration form

A POST request with the relevant details will be sent. User will be authenticated to check whether the user id is valid. If there is an error occurred during authentication process the error response will be sent back. If the authenticated process success the provided data in the request body will be validated. If there is no error occurred during validation, a new workshop will be created and saved. If error occurred during saving the error response is returned. Else successful message will be sent back.

Update conference details

A PATCH request with the workshop id and updated details will be sent toward the authentication service. Then the request body get validated. Then the availability of workshop details will be checked and if the object is available updated details will be saved to the database. If error occurred during saving the error response is returned. Else successful message will be sent back.

Delete conference details

A DELETE request with the workshop id will be send toward the authenticated service. Then request body will be validated. The workshop details will be deleted from the database. If product was deleted successfully successful message will be send back. Else error response will be sent back.

2.2.3 Test cases

Test Case 01		
Author	User	
Summary	User register for Conference / Workshop details	
Precondition	User need to be authenticated.	
Step No.	**Step Action**	**Expected output**
1	User visit to the homepage.	Navigate to the homepage.
2	User click on “Workshop” button.	Navigate to the workshop page.
3	Click on “Register for workshop” button.	Navigate to the registration form.
4	User fill email.	Text field should accept input data.
5	User fill Name.	Details were filled in the text field.
6	User fill Affiliation	Details were filled in the text field.
7	User fill mobile number.	Text field should accept input data.
8	User fill prefeed workshop.	User select the preferred workshop.
9	User fill statement of interest.	Details were filled in the text field.
10	User click on “Register” button.	Application alerts, “Your registration details were successfully saved” and navigates to the home page

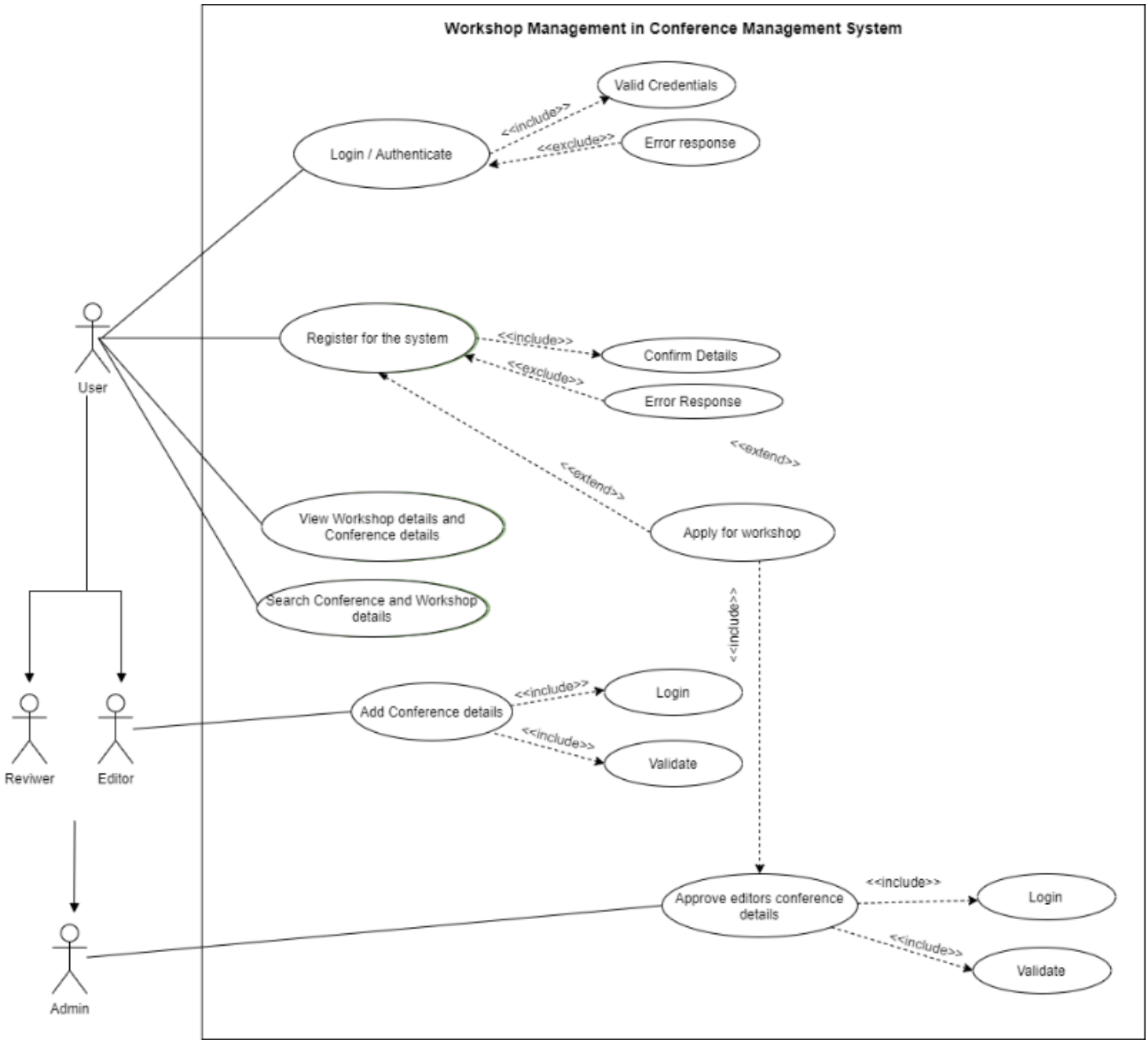
Test Case 02		
Author	User	
Summary	User register for conduct Conference / Workshop.	
Precondition	User need to be authenticated.	
Step No.	**Step Action**	**Expected output**
1	User visit to the homepage.	Navigate to the homepage.
2	User click on “Workshop” button.	Navigate to the workshop page.
3	Click on “Register for conduct workshop” button.	Navigate to the registration form.
4	User fill title of the workshop.	Details were filled in the text field.
5	User fill organizers name.	Details were filled in the text field.
6	User fill organizers Affiliation	Details were filled in the text field.
7	User fill organizers email.	Text field should accept input data.
8	User fill Scope and topic of workshop.	Details were filled in the text field.
9	User fill potential participant.	Details were filled in the text field.
10	User fill planned duration.	Details were filled in the text field.
11	User fill preferred day of workshop.	Details were filled in the text field.
12	User fill name of the referred papers.	Details were filled in the text field.
13	User click on “Register” button.	Application alerts, “Your registration details were successfully saved” and navigates to the home page

Test Case 03		
Author	Editor	
Summary	Add new workshop /conference to the system.	
Precondition	User need to be authenticated.	
Step No.	**Step Action**	**Expected output**
1	User visit to the homepage.	Navigate to the homepage.
2	Click on admin dashboard.	Navigate to the admin dashboard.
3	User click on “Workshop” button.	Navigate to the workshop page.
4	Click on “Add new workshop” button.	Navigate to the add workshop page.
5	User fill title of the workshop.	Details were filled in the text field.
6	User fill organizers name.	Details were filled in the text field.
7	User fill organizers Affiliation	Details were filled in the text field.
8	User fill Scope and topic of workshop.	Details were filled in the text field.
9	User fill potential participant.	Details were filled in the text field.
10	User fill planned duration.	Details were filled in the text field.
11	User fill preferred day of workshop.	Details were filled in the text field.
12	User click on “Add” button.	Application alerts, “Your details were added successfully”.

Test Case 04		
Author	Editor	
Summary	Edit existing workshop /conference in the system.	
Precondition	User need to be authenticated.	
Step No.	**Step Action**	**Expected output**
1	User visit to the homepage.	Navigate to the homepage.
2	Click on admin dashboard.	Navigate to the admin dashboard.
3	User click on “Workshop” button.	Navigate to the workshop page.
4	Click on “Edit workshop” button.	Navigate to the edit workshop page.
6	User edit organizers name.	Details were filled in the text field.
7	User edit organizers Affiliation	Details were filled in the text field.
9	User edit potential participant.	Details were filled in the text field.
10	User edit planned duration.	Details were filled in the text field.
11	User edit preferred day of workshop.	Details were filled in the text field.
12	User click on “Save” button.	Application alerts, “Your details were saved successfully”.

Test Case 05		
Author	Admin	
Summary	Accept the request for conducting workshop.	
Precondition	User need to be authenticated.	
Step No.	**Step Action**	**Expected output**
1	User visit to the homepage.	Navigate to the homepage.
2	Click on admin dashboard.	Navigate to the admin dashboard.
3	User click on “Workshop” button.	Navigate to the workshop page.
4	User Click on “Request for conduct workshop” button.	Navigate to the request page.
5	User click on “Accept” button.	Application alerts, “Your details were saved successfully”.

2.2.4 Use Case Diagram



2.2.5 React Component Diagram

1. App
    - My Workshop Page
        - Search workshop
        - Register for workshop form
        - Register for conduct workshop form
2. Admin Dashboard
    - Workshop component
        - Add new workshop
        - Edit workshop
        - Delete workshop
        - Accept request for conduct workshop

2.2.6 Mongo database document screenshot


```
_id: ObjectId("60b3ec2a2e6260b8562c7935")
Name: "Amali Perera"
Mobile_no: 716549889
Satement of intrest: "IT"
affiliation: "AB industries"
email: "amali1899gmail.com"
prefered_Workshop: "Modern treds in IT"
```

```
_id: ObjectId("60b3f0c22e6260b8562c7936")
Title of Workshop: "Modern trens in IT"
Name of organizer: "Nimal Fernando"
Affiliation of organiz… : "Lecture at SLIIT"
Email of organizer: "nimal123@gmail.com"
Scope of workshop: "Modern trednds in IT "
Potental participant: "Commitee memer at AB club"
Planned Duration: "3 hrs"
prepered Day: "2021.08.08"
Name of the refered pa… : "Software Engineering, Java official documentation"
```

2.3 IT19215884 - Research paper management

2.3.1 Description

Authors can submit, update, delete and see the details of a submitted papers under paper management. Not only that user can download templates from the site. Users can also create their own templates and upload them to the site. Furthermore, they can update and delete those templates.

When submitting a paper, the details of the paper such as title, subject of the paper, names of the authors, type of the paper, number of the pages in the paper and submitting date will be taking as the inputs. Date is not compulsory to fill, if that field is empty, the default value will be set to the current date and time. The content of the paper should be uploaded to the site in pdf format. The templates also can be uploaded to the site and if in need of uploading an updated version, user can delete the previous and upload the new one.

2.3.2 Rest API

Submit Paper

A POST request with the above-mentioned details will be sent. The user should be an authenticated user, so that user id will be gone through an authentication checkup. If there is an error, an error message will be returned. If the user is an authenticated user, the provided data in the request body will be validated. If there is no error occurred in validation, a new paper will be created and saved in the DB. If an error occurred, the user will be informed by an error message. After the new paper saved in the DB successfully, a success message will be returned. Otherwise, an error message will be returned.

Get all papers

A GET request will be sent to the paper management service. After authentication, the validation check will be taking placed and if it is successfully validated, all the papers submitted by the user in the DB will be fetched. In case if an error occurred, error message will be returned.

Fetch one paper

A GET request with the specified id will be sent to the paper management service. After authentication, the paper id will be validated and if there is no error, the paper corresponding to the id will be fetched. In case if an error occurred, error message will be returned.

Update paper

A PATCH request with the paper id and updated details will be sent to the paper management service. After the authentication, the request will be validated. Then the availability of the paper object is checked and if available, paper will be updated and saved in the DB. If there is error in the process error message will be returned.

Delete paper

A DELETE request with the paper id will be sent to the paper management service. After the authentication, the request will be validated. Then the availability of the paper object is checked and if available the paper will be deleted. If there is error in the process error message will be returned.

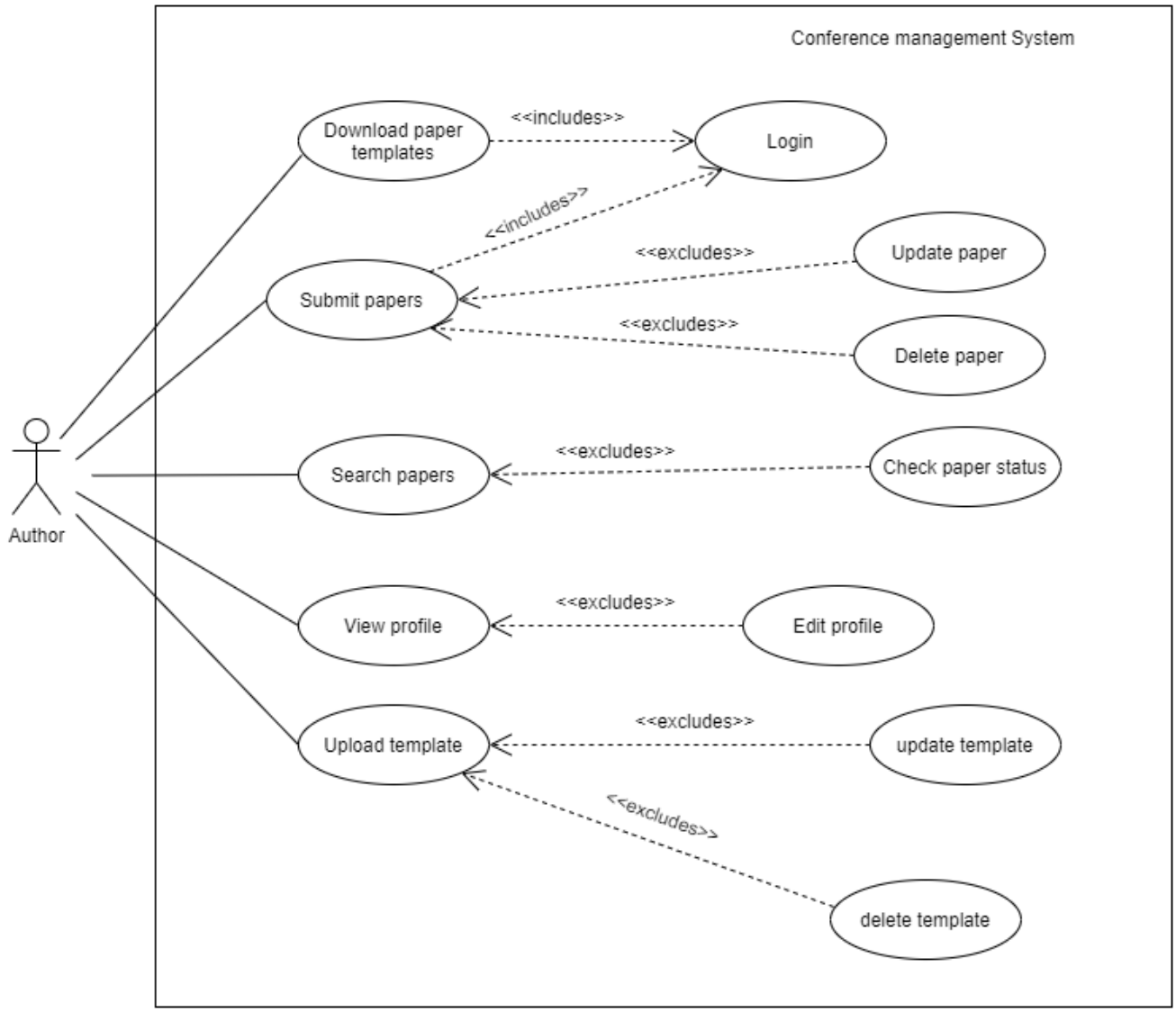
2.3.3 Test cases

1. Summary This test case checks whether the paper templates are downloaded in word format.		
Preconditions The user should be an authenticated user.		
Step number	**Step action**	**Expected result**
1	User clicks on "Explore templates"	User should successfully navigate to templates
2	User selects a template and clicks download button	The templated downloads in word format.

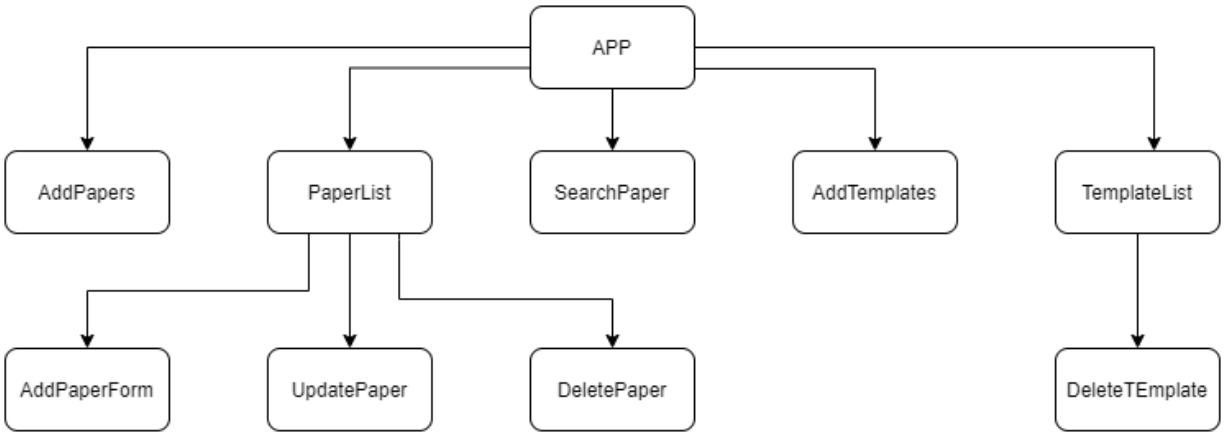
2. Summary This test case checks whether the user fills all the fields of the form when submitting a paper.		
Preconditions The user should be an authenticated user.		
Step number	**Step action**	**Expected result**
1	User clicks on "submit paper" button in the dashboard	User should successfully navigate to submit paper page
2	User enters title of the paper.	The text field should accept the entered data.
3	User enters subject of the paper.	The text field should accept the entered data.
4	User enters authors of the paper.	The text field should accept the entered data.
5	User selects the type of the paper.	The text field should accept the entered data.
6	User enters no: of pages of the paper.	The text field should accept the entered data.
7	User uploads the content of the paper in pdf format.	The text field should accept the entered data.
8	User clicks on "save" button	A paper is submitted successfully, and details saved in the DB.

3. Summary This test case checks whether the number of pages field does not accept letters		
Preconditions The user should be an authenticated user.		
Step number	**Step action**	**Expected result**
1	User clicks on "submit paper" button in the dashboard	User should successfully navigate to submit paper page
2	User enters characters to no: of pages field.	Error message should be prompt.

2.3.4 Use Case Diagram



2.3.5 React Component Diagram



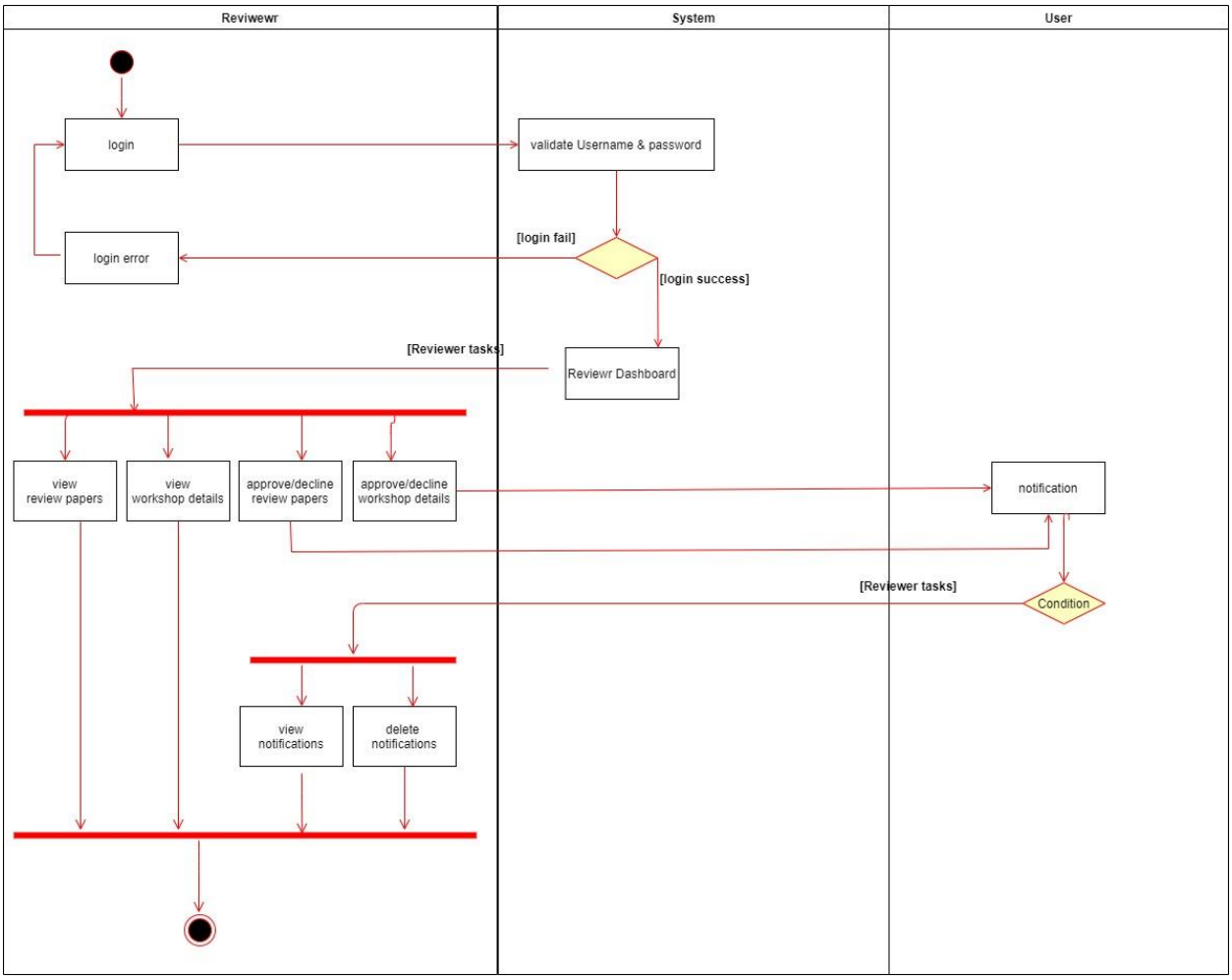
2.3.6 Mongo database document screenshot


```
_id: ObjectId("60b2654c2b13e62b60b60400")
title: "Corona Crisis"
subject: "Health"
author: "John Wells"
type: "Proposal"
pages: "10"
status: "Pending"
date: 2021-05-29T16:01:16.691+00:00
__v: 0
```

2.4 IT19056258 - Reviewers management

2.4.1 Description

Review can access the system by successfully login. After login reviewer able to view all workshop details and research paper details as list and again reviewer can see all the details and research papers in detail. Reviewers have authority to approve and decline the research papers and workshop details. After approving or decline reviewer will send the notification to user. Finally, reviewer can view list of notification as a list and can delete notifications from the database



Activity diagram for Reviewer function

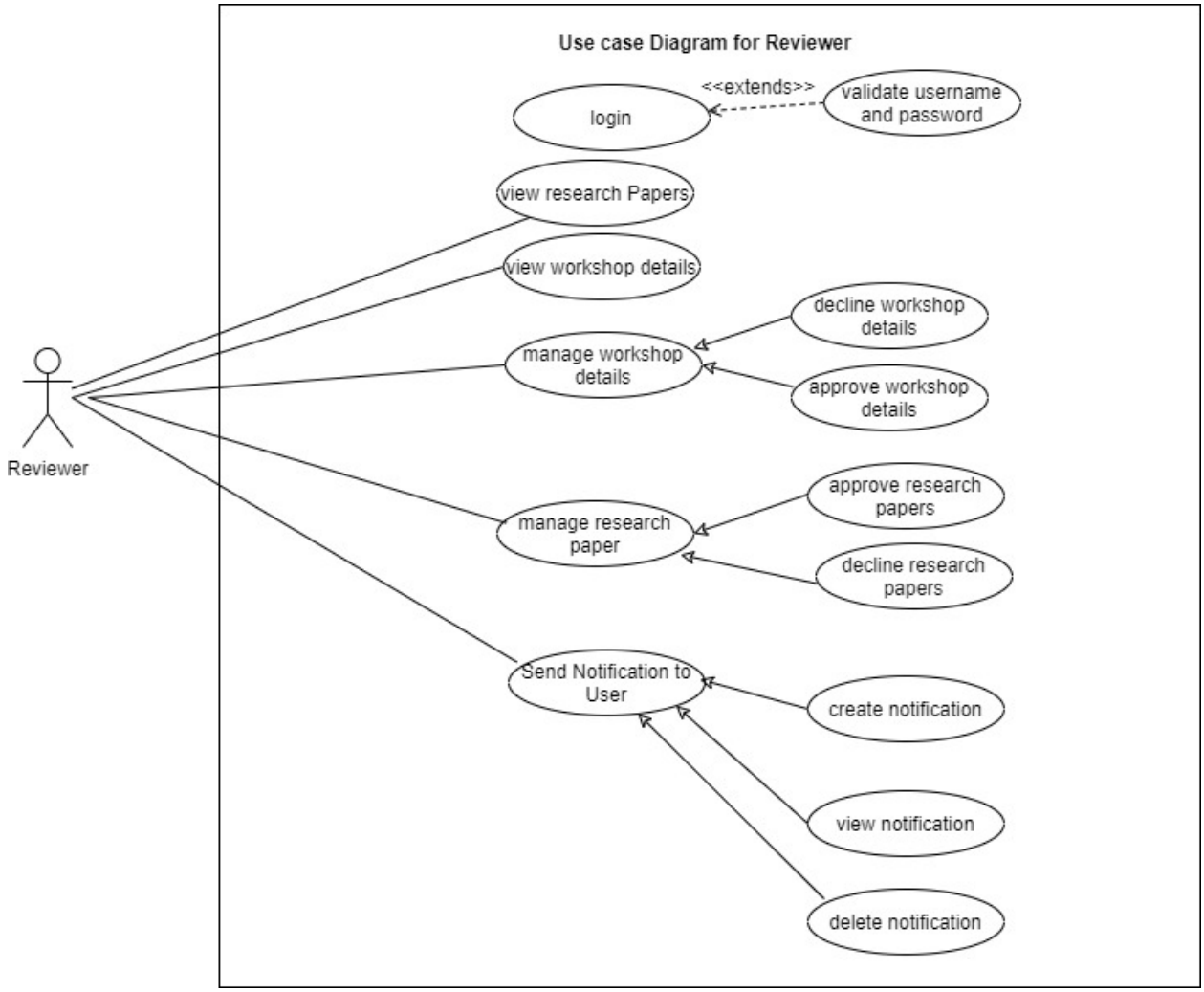
2.4.2 Rest API

Reviewer visit to web page then the component is mount to the DOM. The componentDidMount or useEffect method is called sending off the HTTP request . HTTP request can be as GET, POST, PUT or DELETE When reviewer click buttons it will send a request to router and from router it will send to controller. Then from controller it will call to paticular model and take the necessary data and will response to reviewer request by displaying data.

2.4.3 Test cases

Test case ID	Test Scenario	Test steps	Test Data	Expected Results	Actual Results	Pass/Fail
1	Check reviewer login with valid Data	1. Go to Site 2. Enter username 3. Enter password 4. Click submit	Userid= reviewer Password = reviewer123	reviewer should login to the dashboard	As expected	Pass
2	Verify field by entering valid email address of user to send notifications	1. Go to site 2. Move to send notification form 3. Type email address of user 4. Click send button	Email = nayomi@gmail.com	Notification should send successfully	As expected	Pass
3	Verify the field of status in approve or decline filled	1. Go to the site 2. Move to Approve or decline form 3. Select approve or decline to status field 4. Click submit	Status = Approved	Display Successful saved message	As Expected,	Pass

2.4.4 Use Case Diagram



2.4.5 React Component Diagram

reviewerModel
- viewResearchPaper form
- viewworkshopdetails form
- Approve/Decline form
- sendNotification form
  - addNotifications
  - view listOfNotifications
  - deleteNotifications

2.4.6 Mongo database document screenshot


```
_id: ObjectId("60b1d987e64bf5a3b521e159")
status: "approved"
user_email: "sureni@gmail.com"
message: "this is approved"
created_date: 2021-05-28T18:30:00.000+00:00
document_id: ObjectId("60b1d987e64bf5a3b521e159")
```