BSc (Hons) in Information Technology

Specializing in Software Engineering

Year 3 - 2021

SE3040 – Application Frameworks

Group Project

Technical Report

2021S1_JUNE_WE_07

International Conference of Application Frameworks Web Application

Student ID	Name
IT18233704	Yamasinghe N.R
IT18230048	Wijewardhana U.G.R.M
IT18209280	Dissanayake D.M.Y.S
IT18213072	Ranasinghe Y.S

Repository Link:

Evaluator’s Name:

Comments:

Table of Content

Table of Figures

1.Introduction

1.1 The System

This Web application for a Conference management tool. SLIIT is organizing an academic conference where researchers present results, workshops, and other activities. It is named “International Conference on Application Frameworks” – ICAF. Here, researchers will present their latest findings and implementations of different programming languages including Java, JavaScript, Python etc. Conference will be physically held at SLIIT. Only registered participants can attend the event.

System functionality will be as follows,

1. There are three roles as admin, editor, reviewer, and user.

2. Anyone can view the conference details as a guest

3. Only users who register have access can submit the research papers, conduct workshops, or present their research-related activities.

4. Editors can add the conference details and edit

5. Admin must approve the editor's content before it appears on the website.

6. The final product must contain the following:

a. A landing page that shows all the related information including venue and date.

b. Download page where the research paper templates, workshop PowerPoint templates, and other templates can be download.

c. Separate pages for each of the main events in the conference. (Research paper presentations, workshops, etc.)

d. A registration page where a user can be registered as a researcher, workshop presenter, or attendee.

e. When the researcher registered to the system, the research paper should be uploaded alongside the contact information.

f. When the workshop conductor is registered to the system, a proposal containing all the necessary details about the workshop should be uploaded alongside the contact information. g. When the user is registered to the system all the necessary contact information should be uploaded.

7. Reviewers should be able to see the research paper uploads and workshop detail uploads in separate pages. The uploads must be available to view.

8. Reviewers can approve or decline the research papers or workshop proposals. And notification should be sent to the relevant user.

9. Attendees must pay upfront to register for the conference (you can come up with the amount) 10. Research paper presenters must pay if their papers got approved to present them at the conference.

11. Workshop conductors do not have to pay.

12. Admin should be able to monitor all these activities in a dashboard (you can decide the format)

1.2 Technologies used in the System

The technologies used in the system as follows:

1. HTML/JavaScript frontend
2. ReactJS
3. NodeJS
4. Express Js
6. JSON based Web Services
7. NoSQL Database (MongoDB)
8. JEST

2. Functionalities Explained Using Diagrams

2.1 User Registration Sequence Diagram

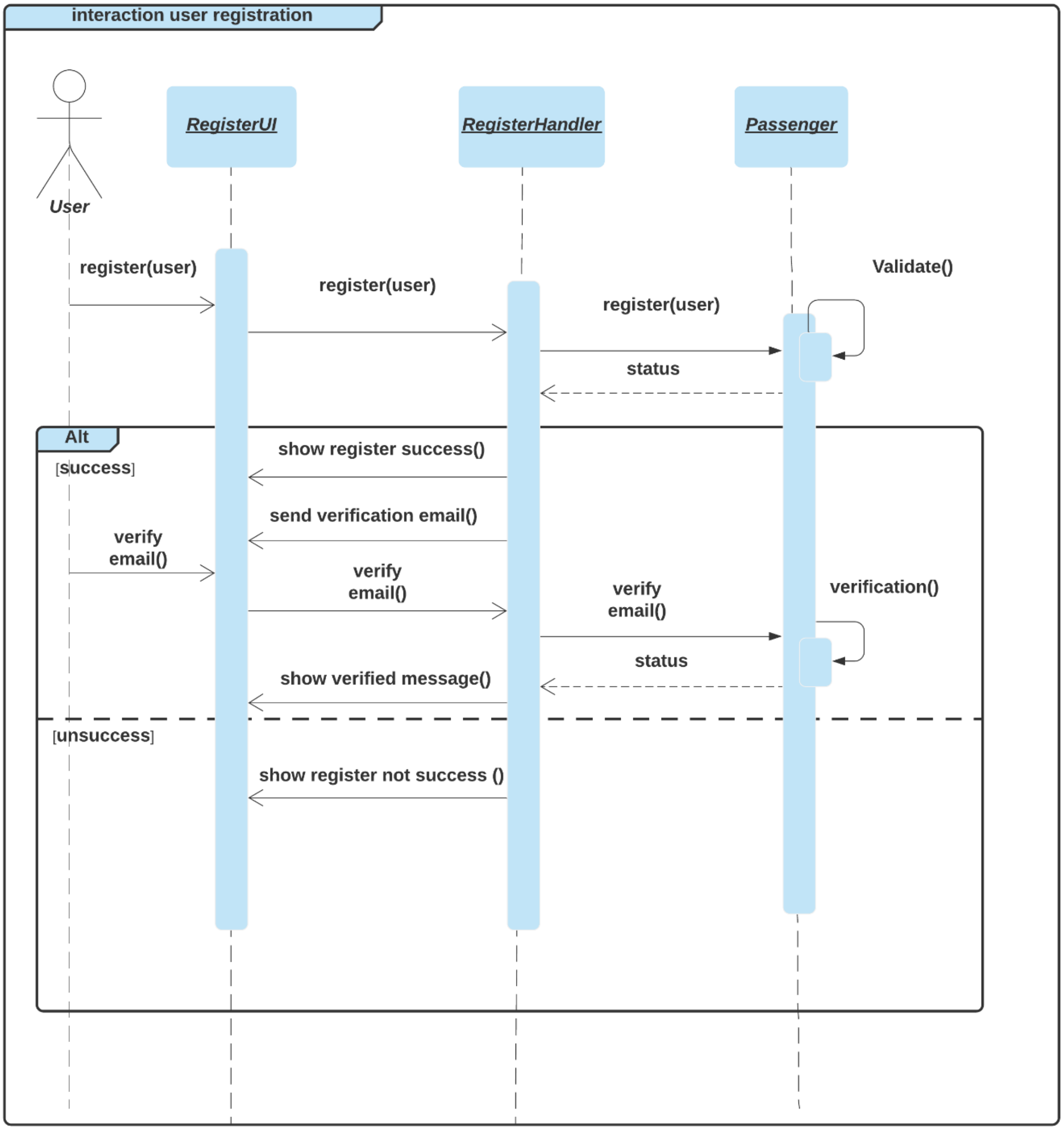


Figure 2.1: user registration sequence diagram

2.2 User Login Sequence Diagram

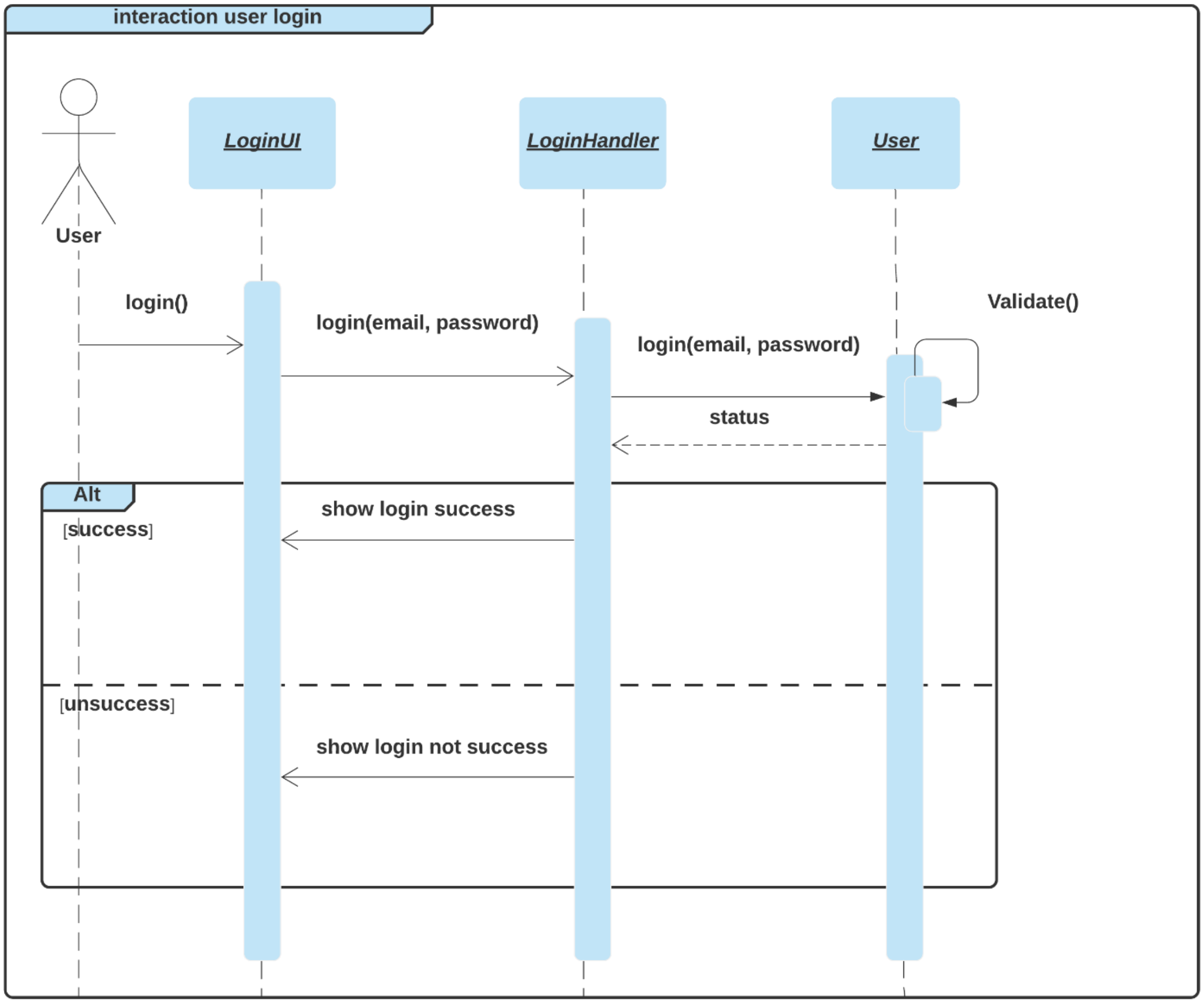


Figure 2.2: user login sequence diagram

2.3 Researcher (User) Activity Diagram

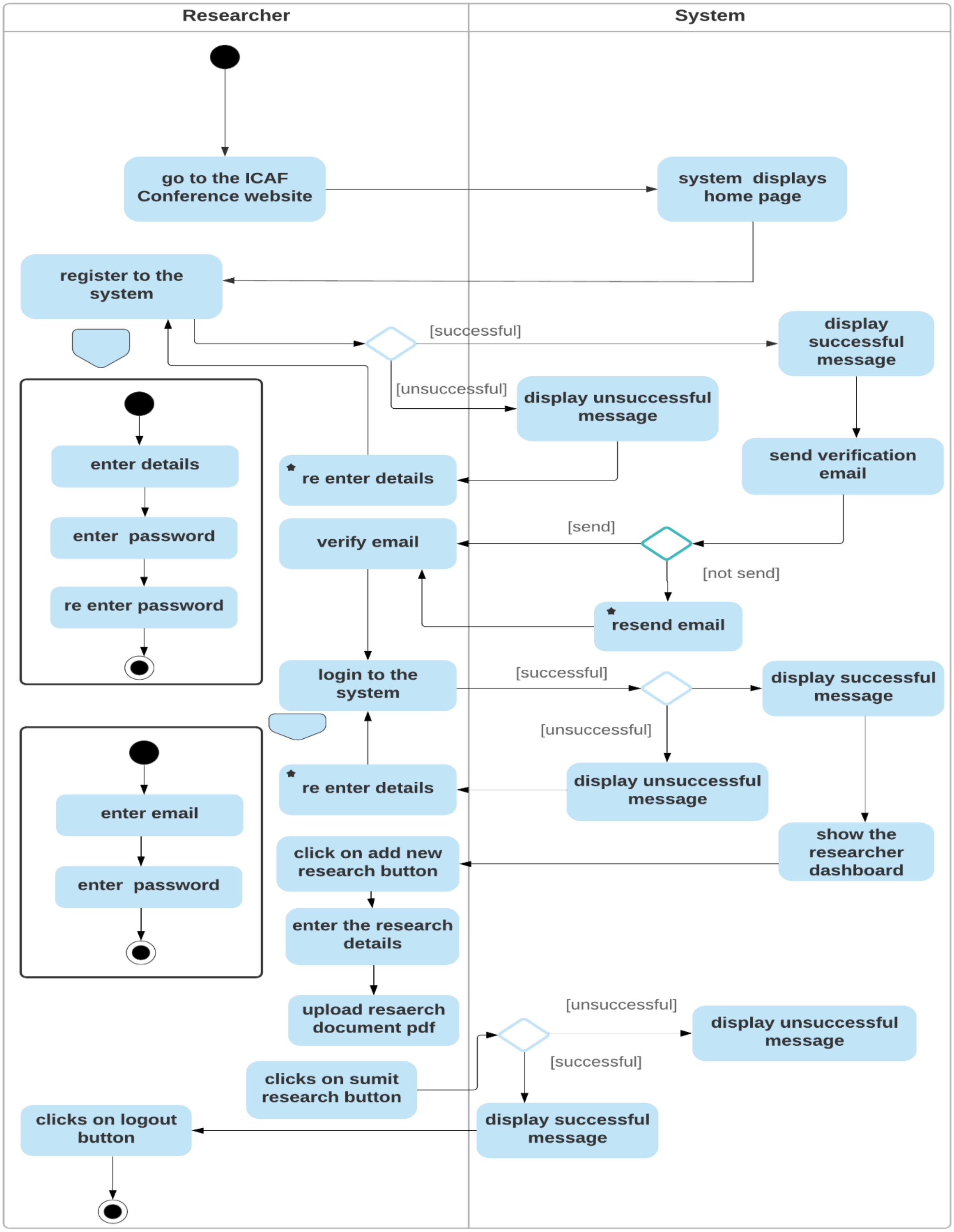

Researcher
System
go to the ICAF Conference website
system displays home page
register to the system
[successful]
display successful message
[unsuccessful]
display unsuccessful message
enter details
enter password
re enter password
re enter details
send verification email
[send]
verify email
[not send]
resend email
login to the system
[successful]
display successful message
[unsuccessful]
enter email
enter password
re enter details
display unsuccessful message
show the researcher dashboard
click on add new research button
enter the research details
upload resaerch document pdf
[unsuccessful]
display unsuccessful message
clicks on sumit research button
[successful]
clicks on logout button
display successful message

Figure 2.3: researcher activity diagram

2.4 Researcher Sequence Diagram

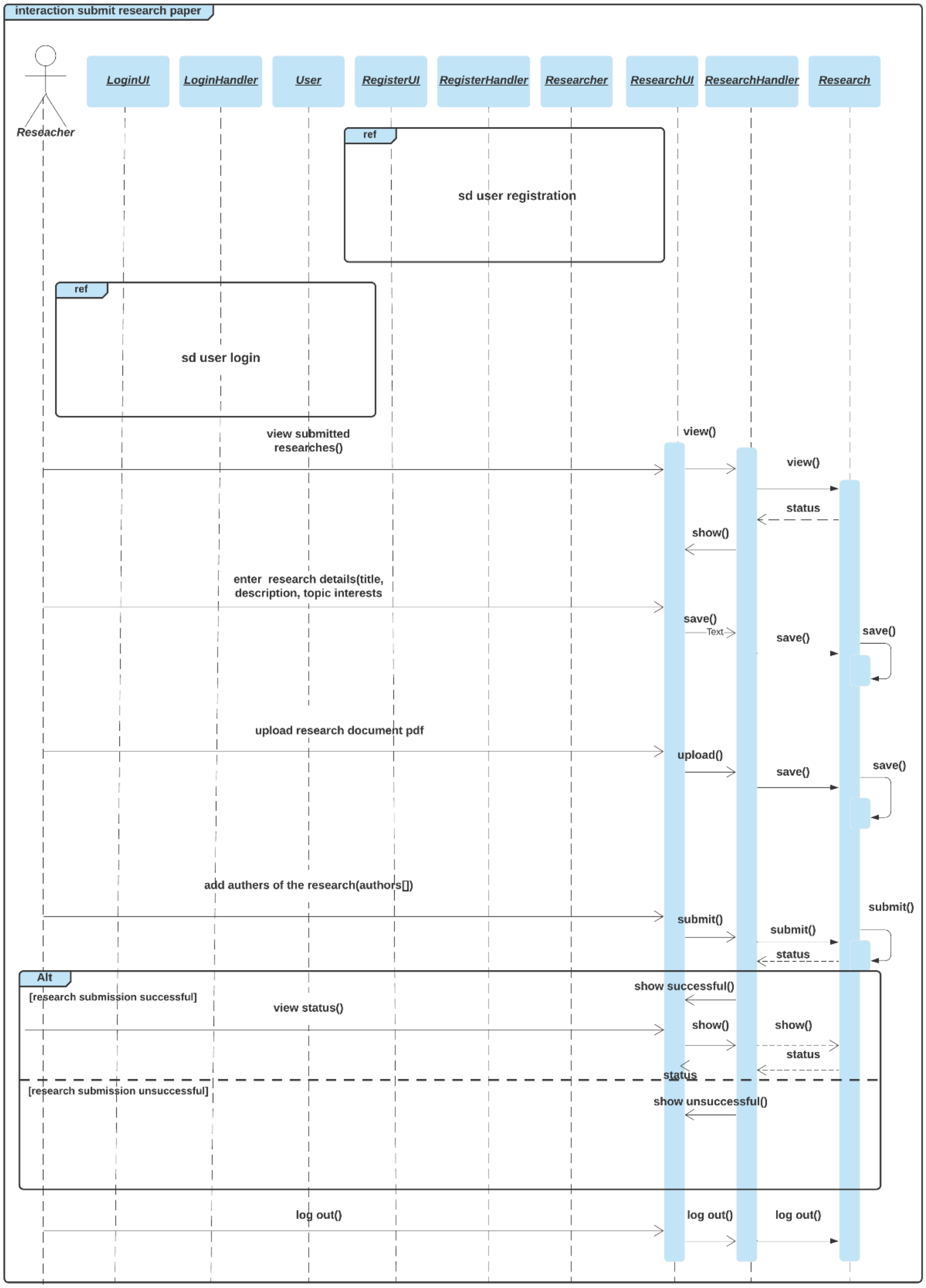


Figure 2.4: researcher sequence diagram

2.5 Reviewer Activity Diagram

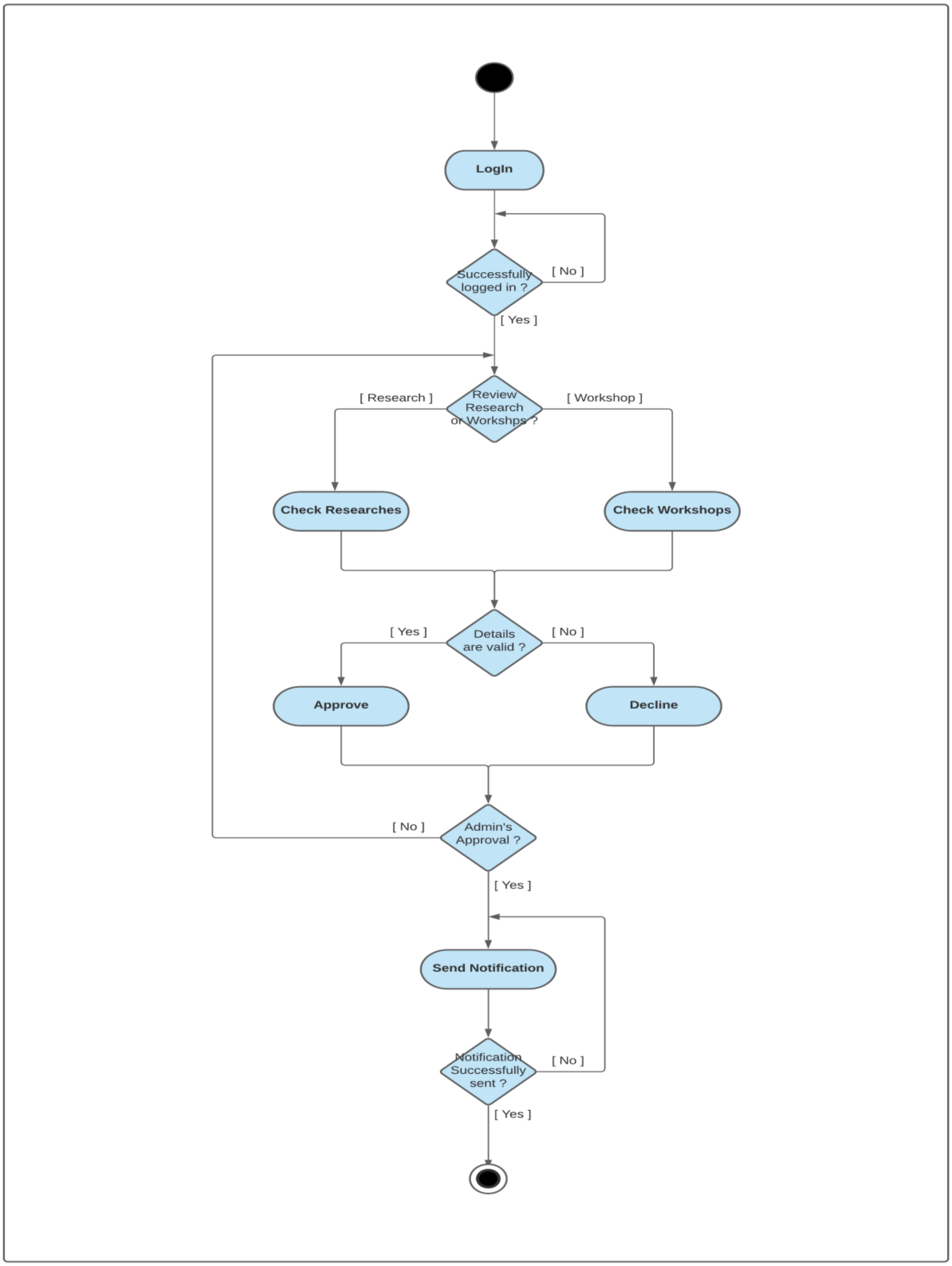


Figure 2.5: reviewer activity diagram

2.6 Reviewer Sequence Diagram

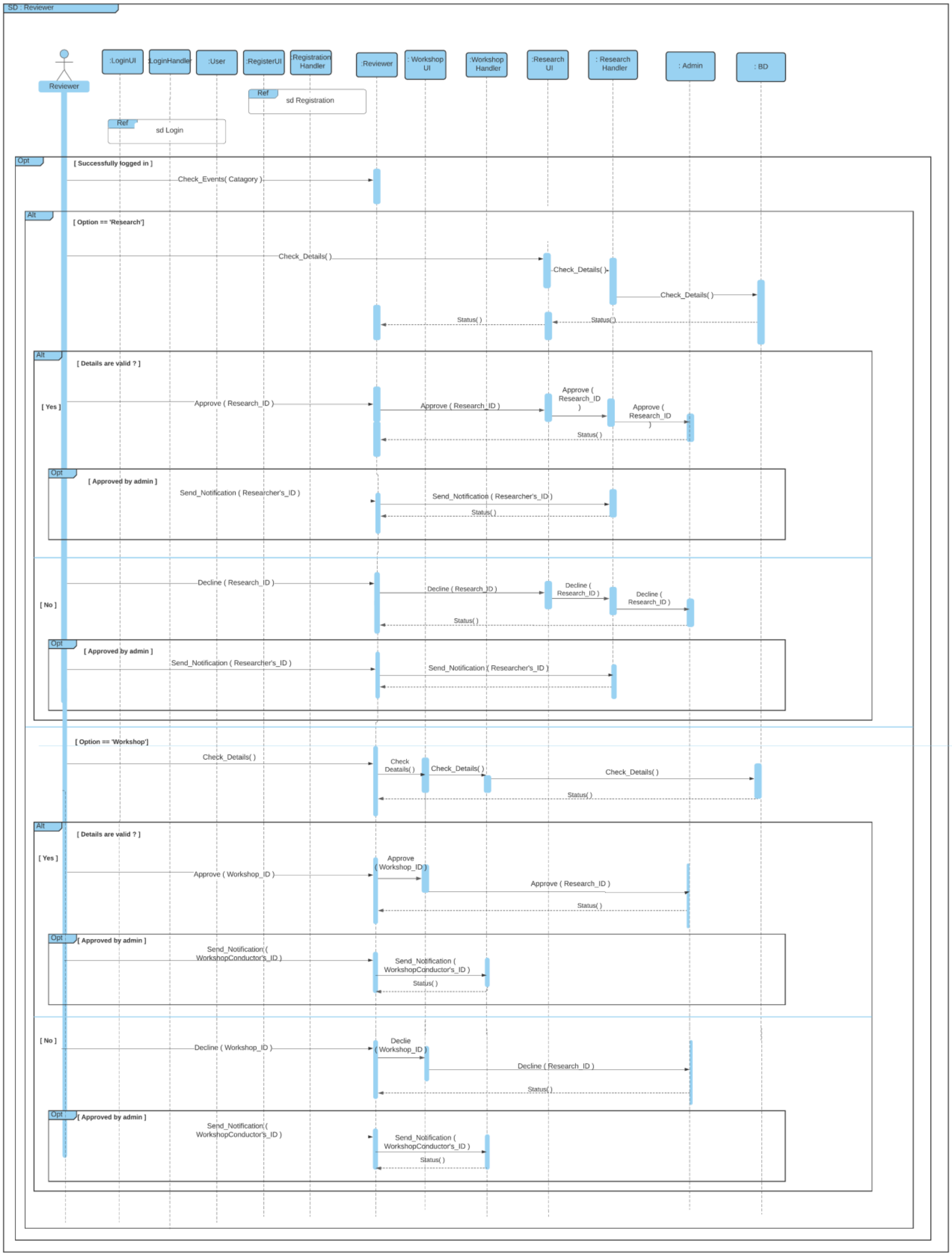


Figure 2.6: reviewer sequence diagram

2.7 Editor Activity Diagram

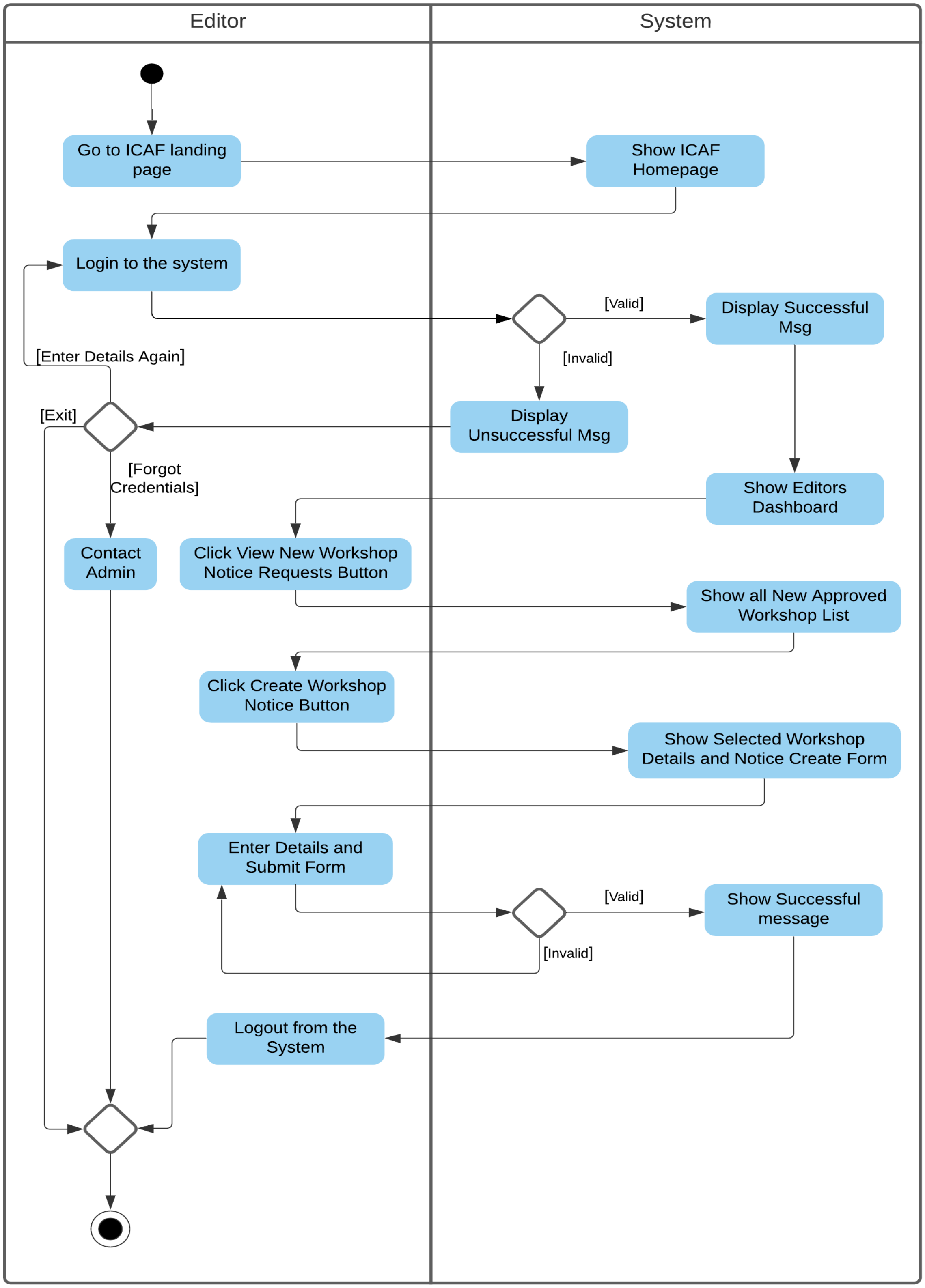


Figure 2.7: editor activity diagram

2.8 Editor Sequence Diagram

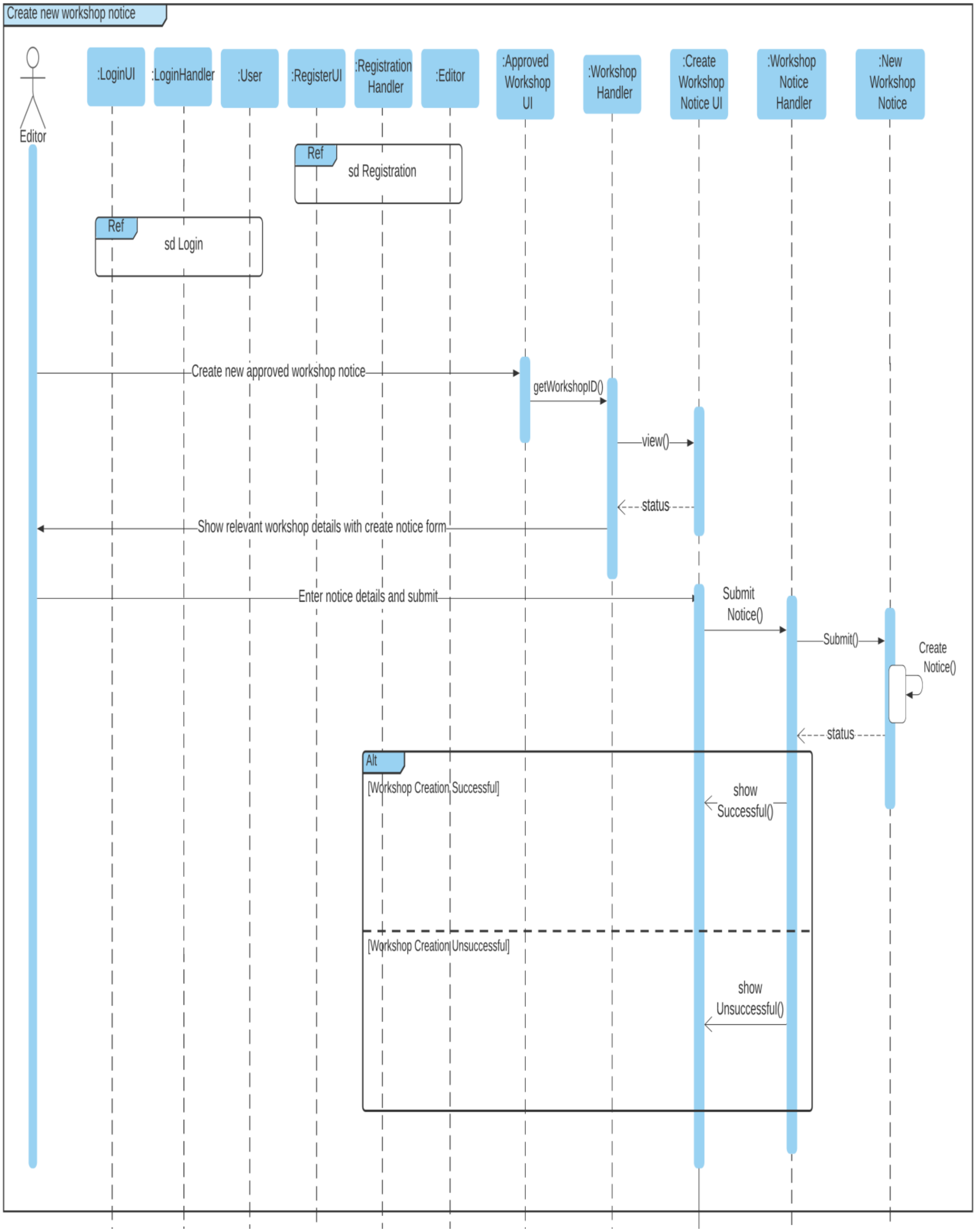


Figure 2.8: editor sequence diagram

2.9 Admin Activity Diagram

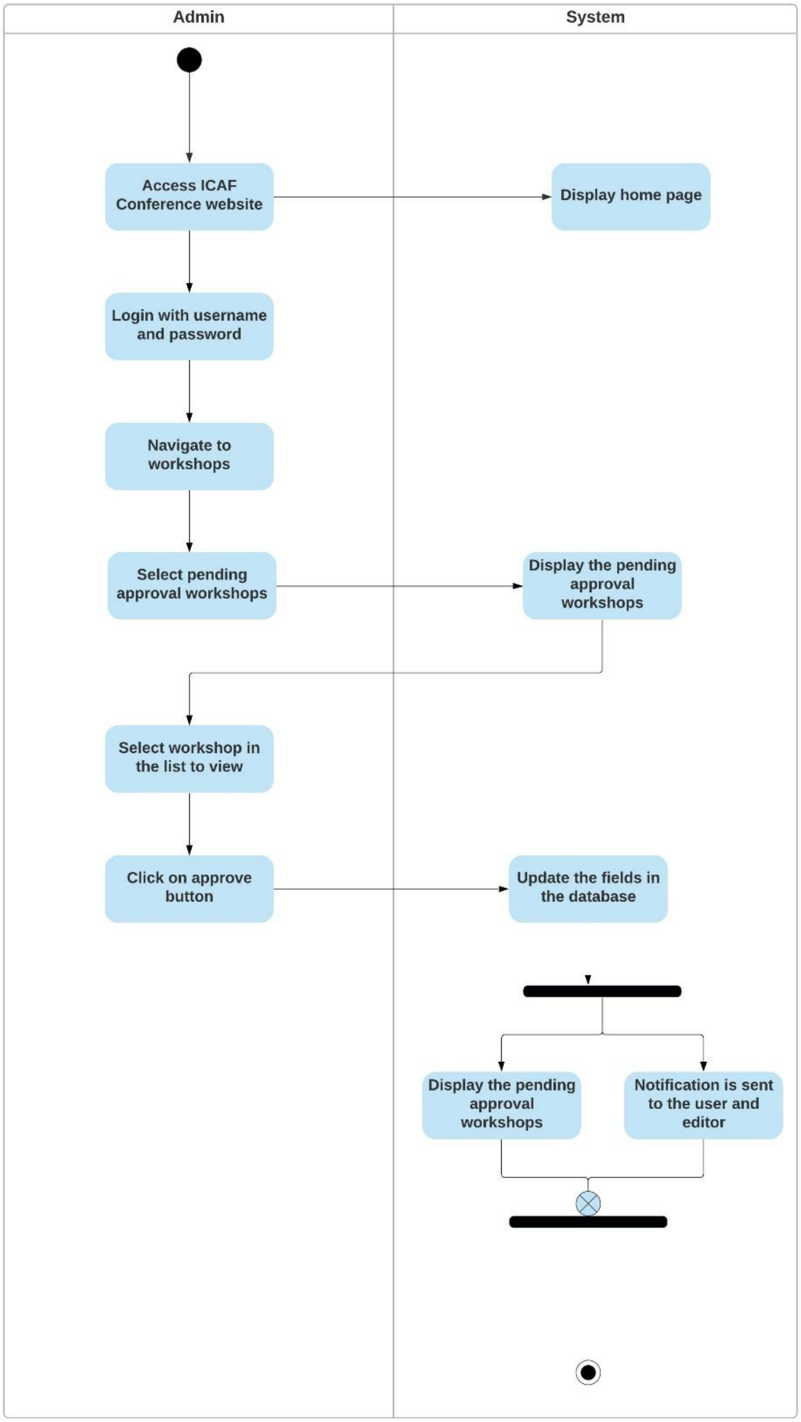

Admin
System
Access ICAF Conference website
Display home page
Login with username and password
Navigate to workshops
Select pending approval workshops
Display the pending approval workshops
Select workshop in the list to view
Click on approve button
Update the fields in the database
Display the pending approval workshops
Notification is sent to the user and editor

Figure 2.9: admin activity diagram

2.10 Admin Sequence Diagram

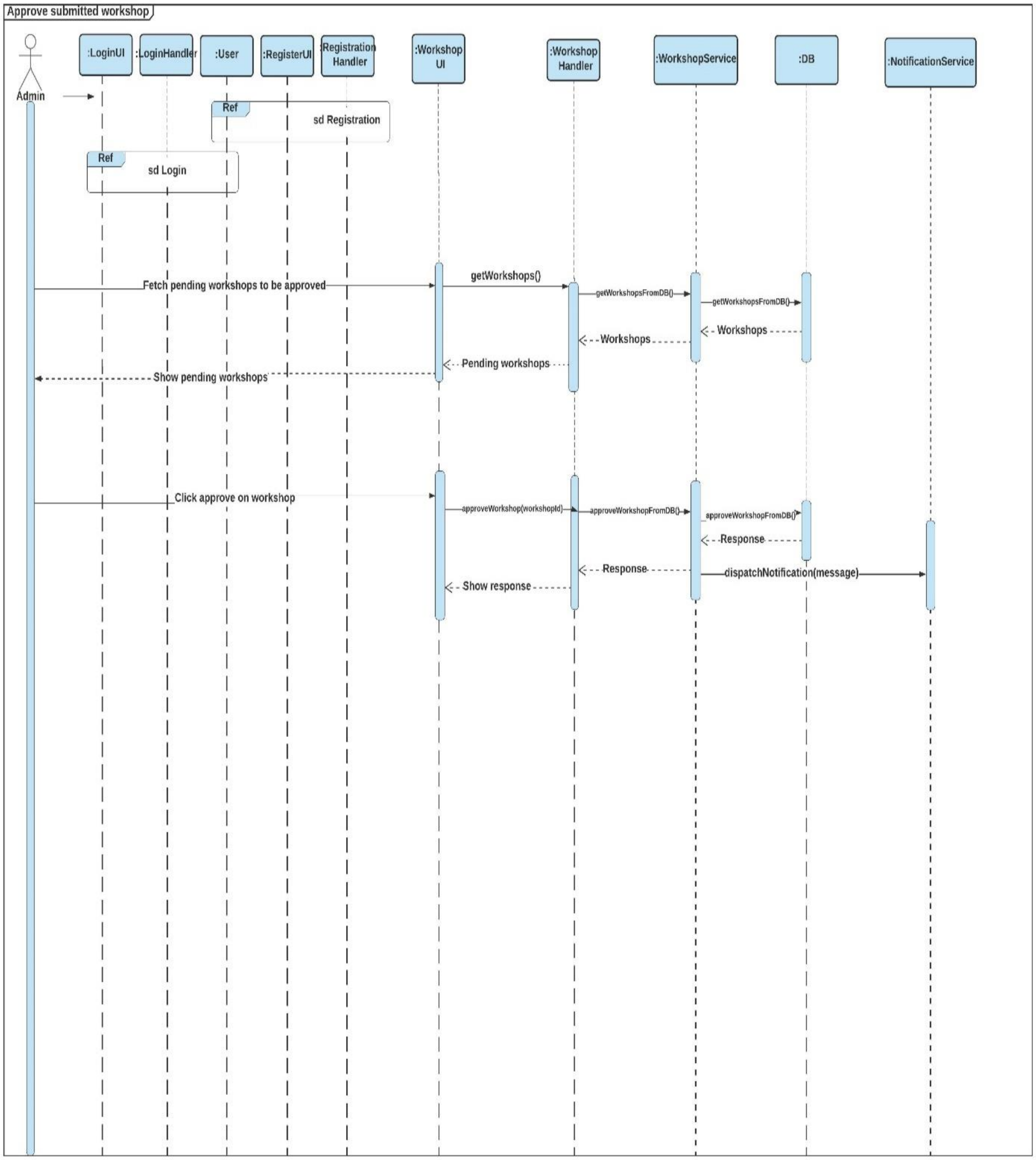


Figure 2.10: admin sequence diagram

3. System Diagrams

3.1 Class Diagram

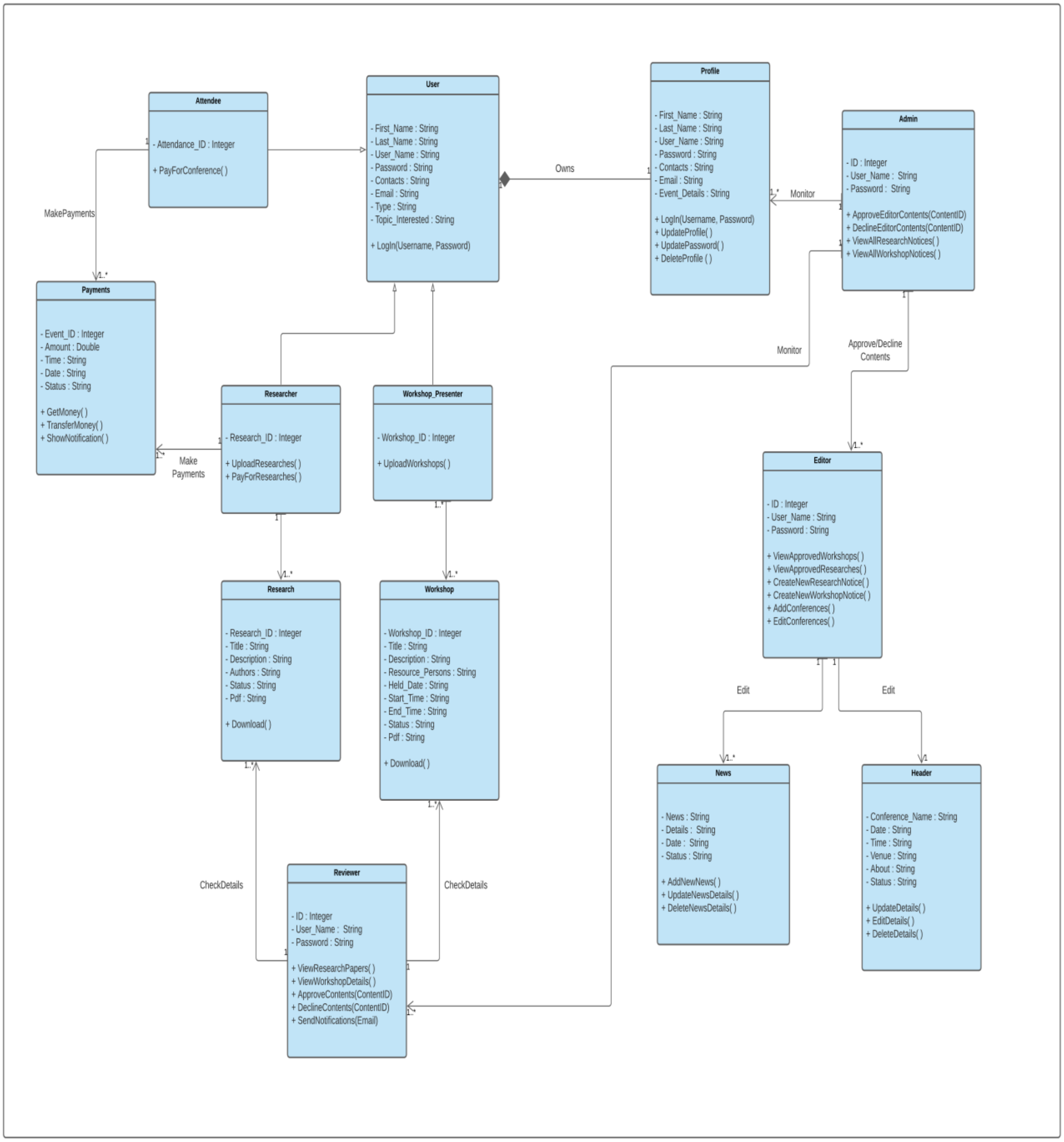


Figure 3.1: system class diagram

3.2 Use Case Diagram

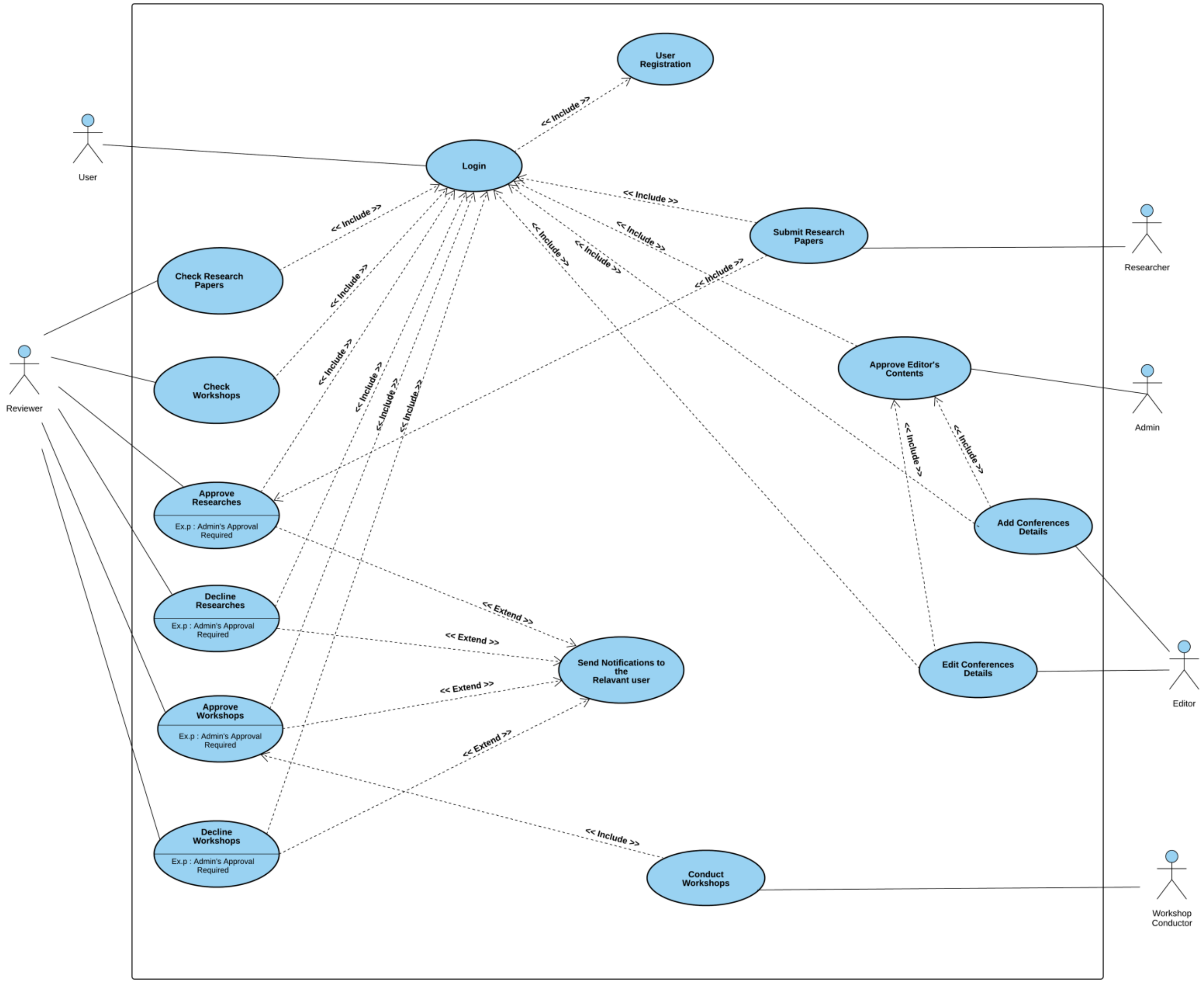


Figure 3.2: system use case diagram

4. React Component Trees

4.1 Home Page

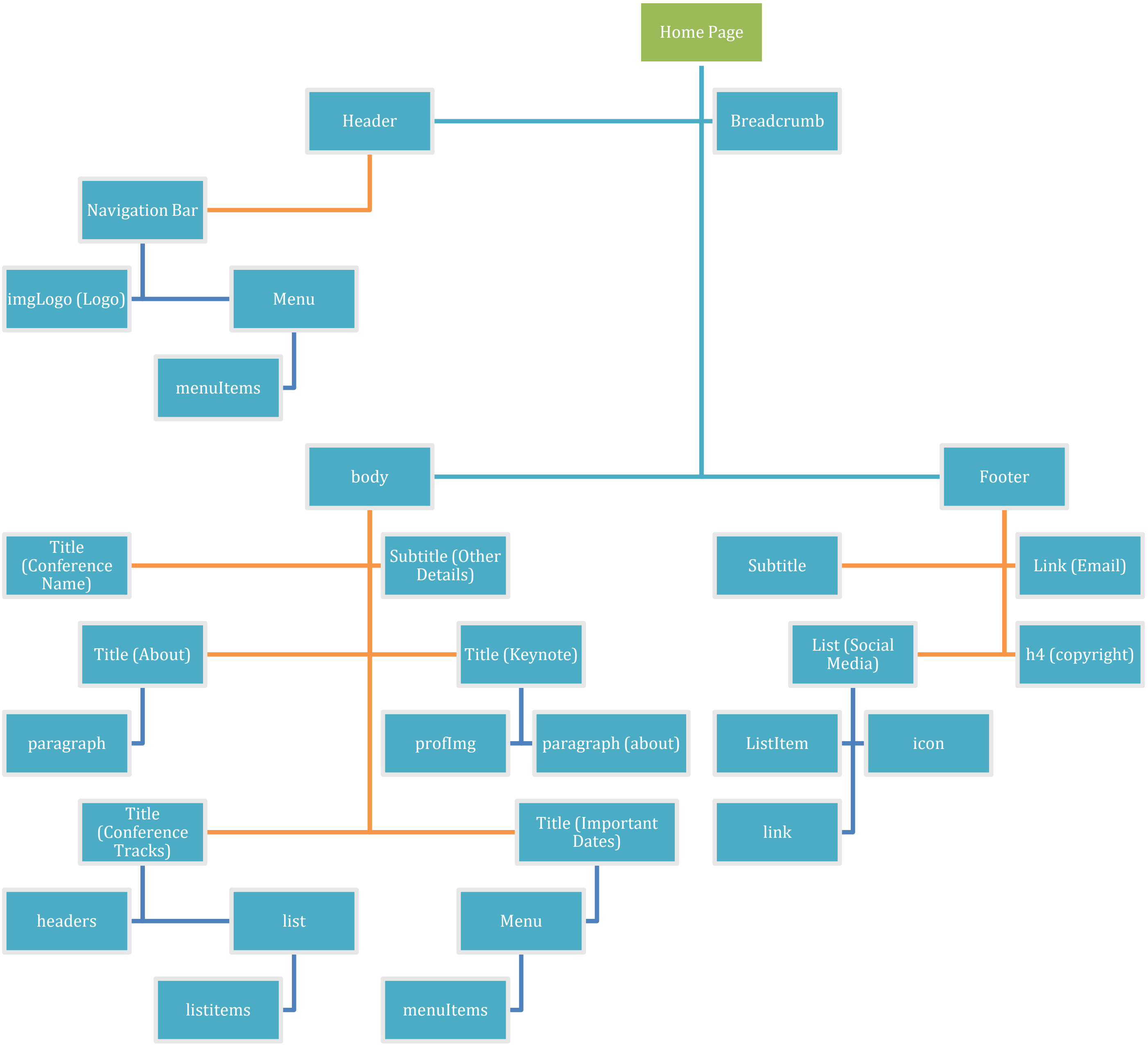


Figure 4.1: home page component tree

4.2 Workshops Page

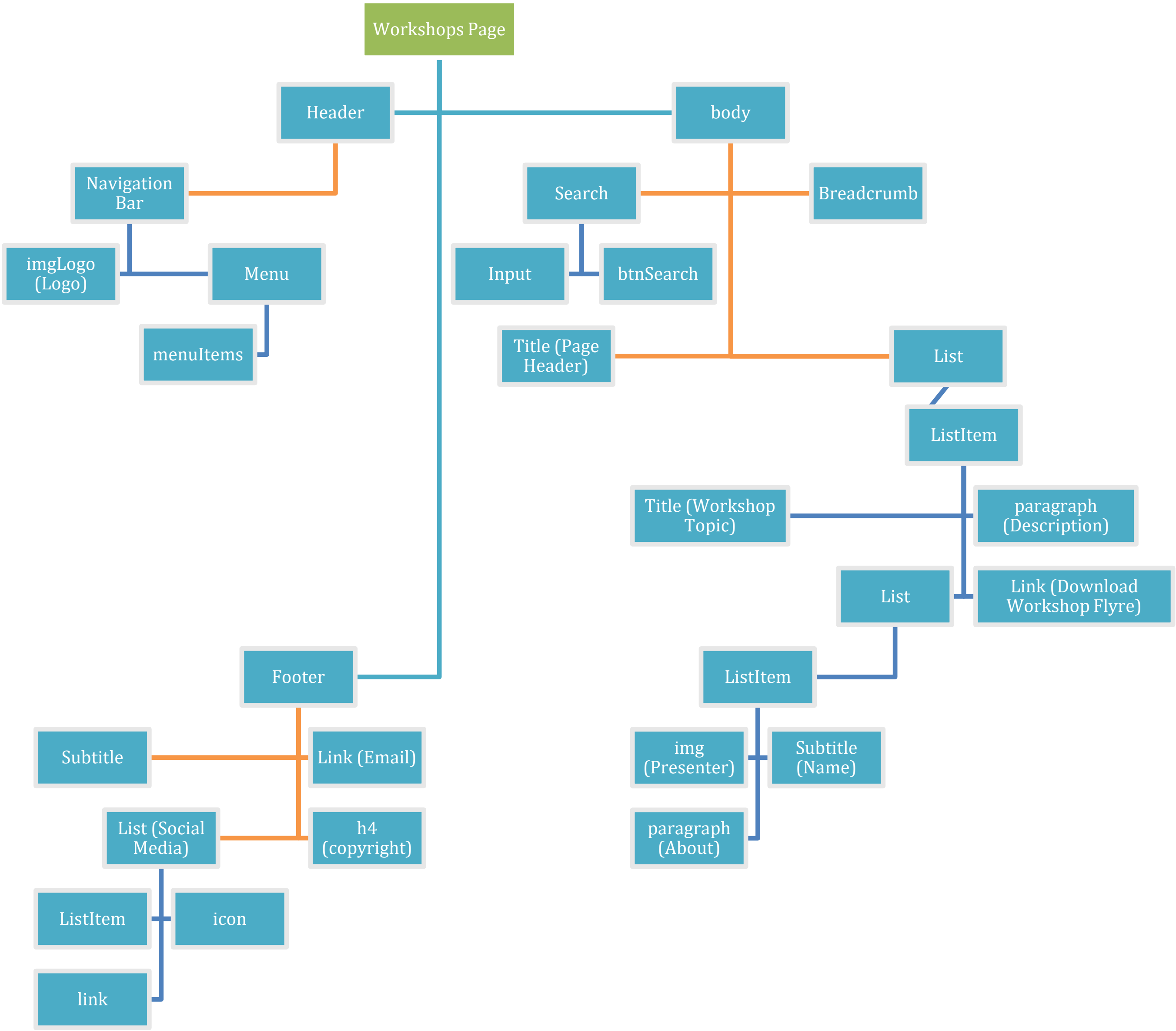


Figure 4.2: workshops page component tree

4.3 Conference Tracks Page

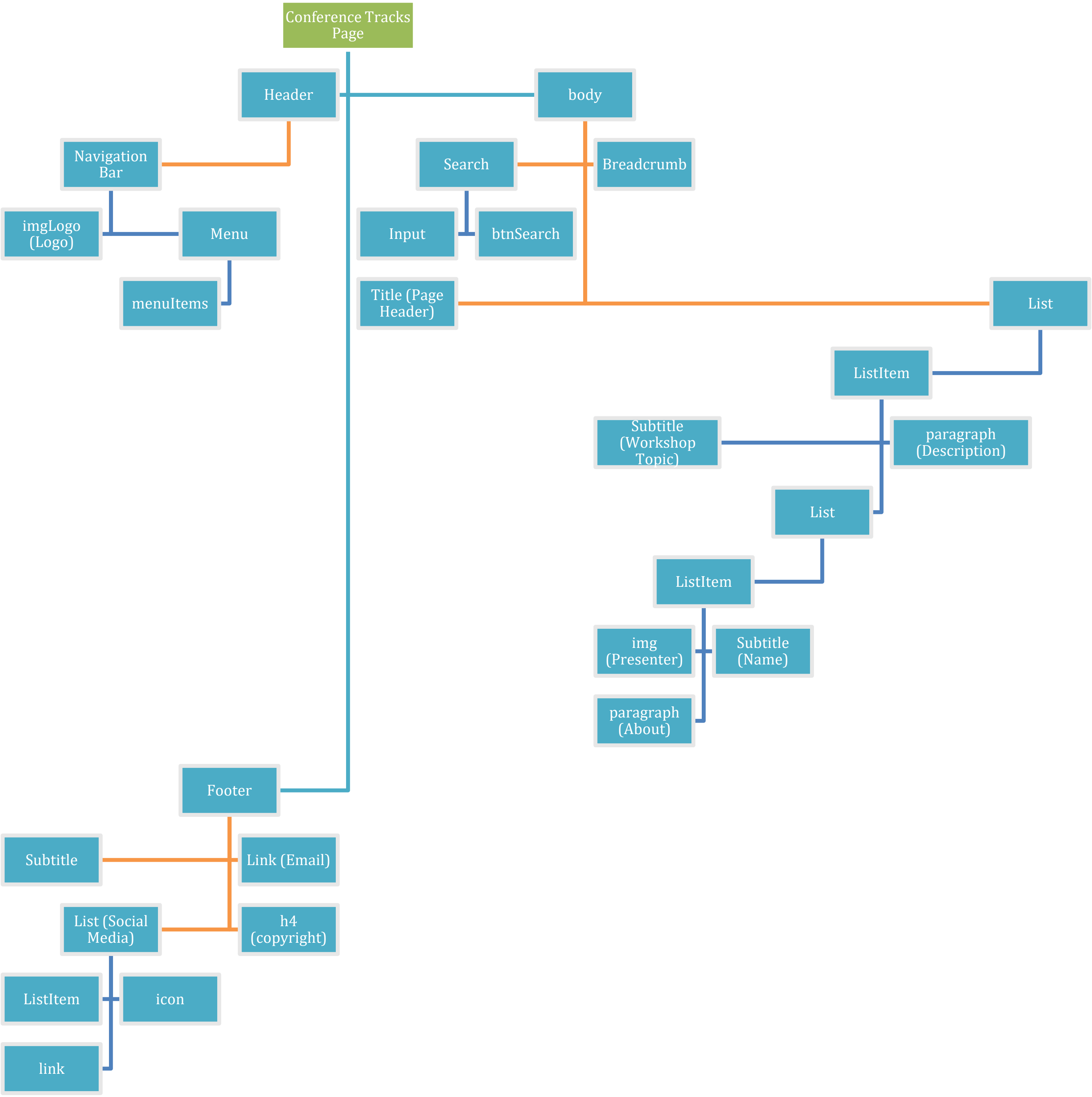


Figure 4.3: conference tracks page component tree

4.4 Attendee Registration Details Page

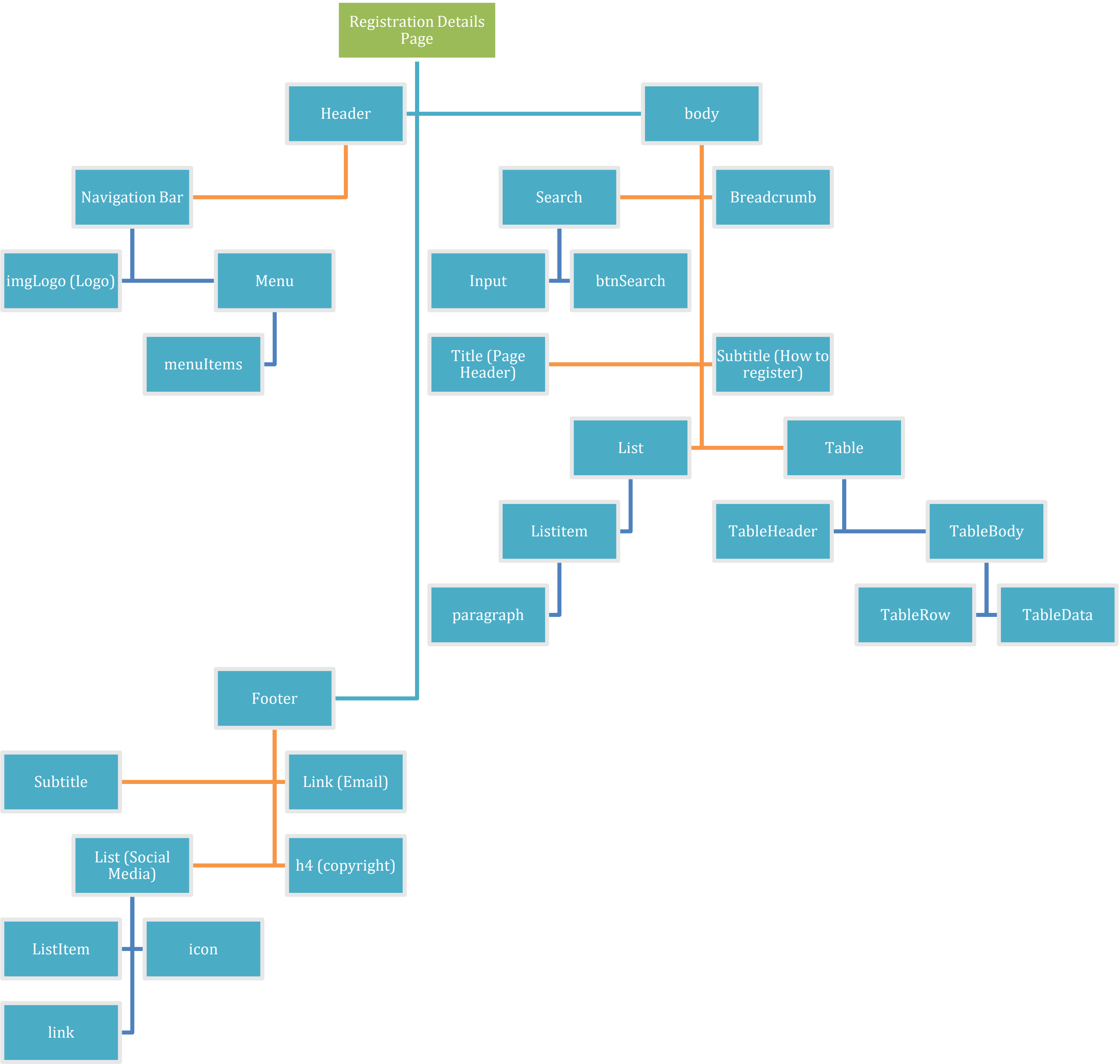


Figure 4.4: attendee registration page component tree

4.5 Login Page (All Users)

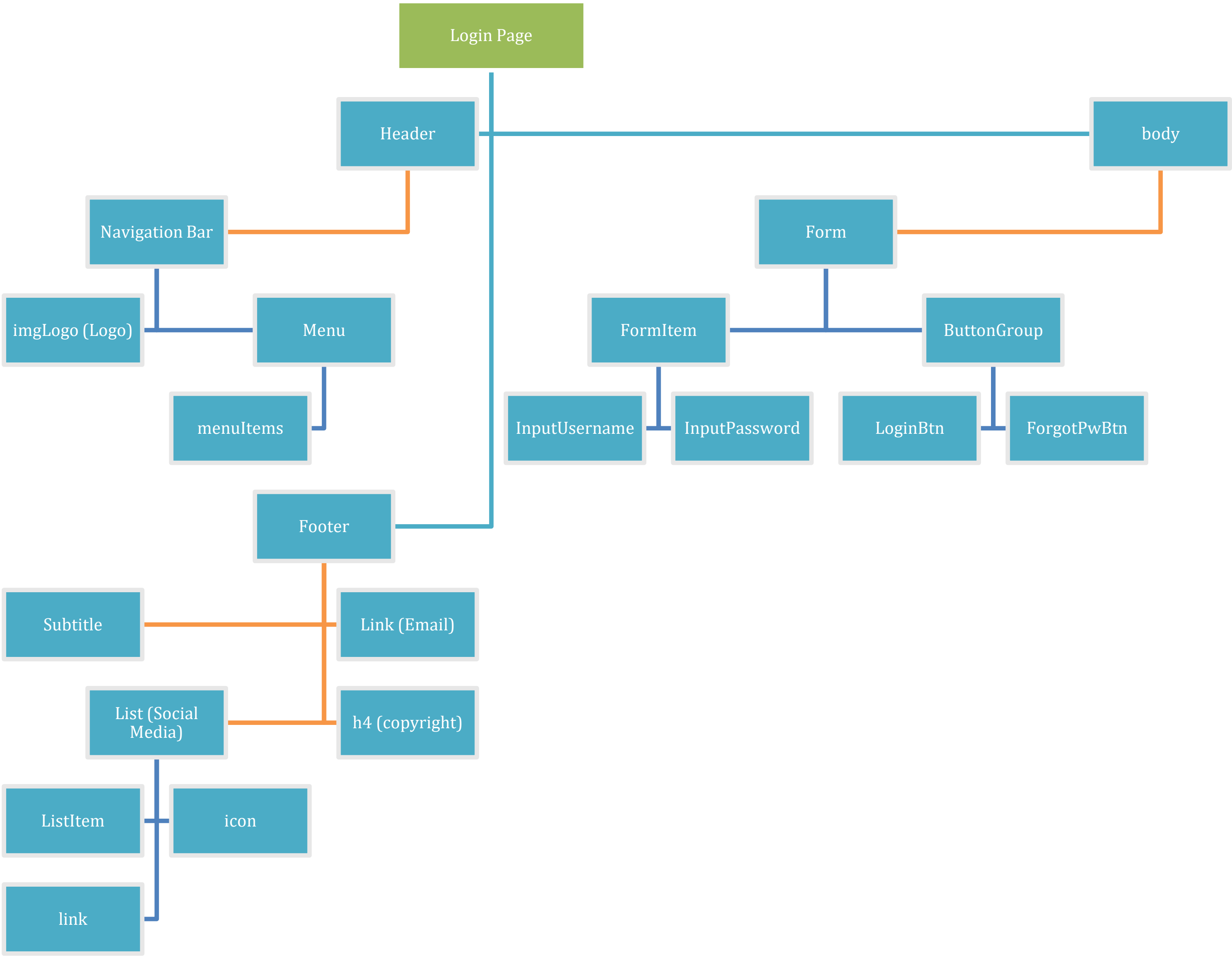


Figure 4.5: login page component tree

4.6 Admin Dashboard Page

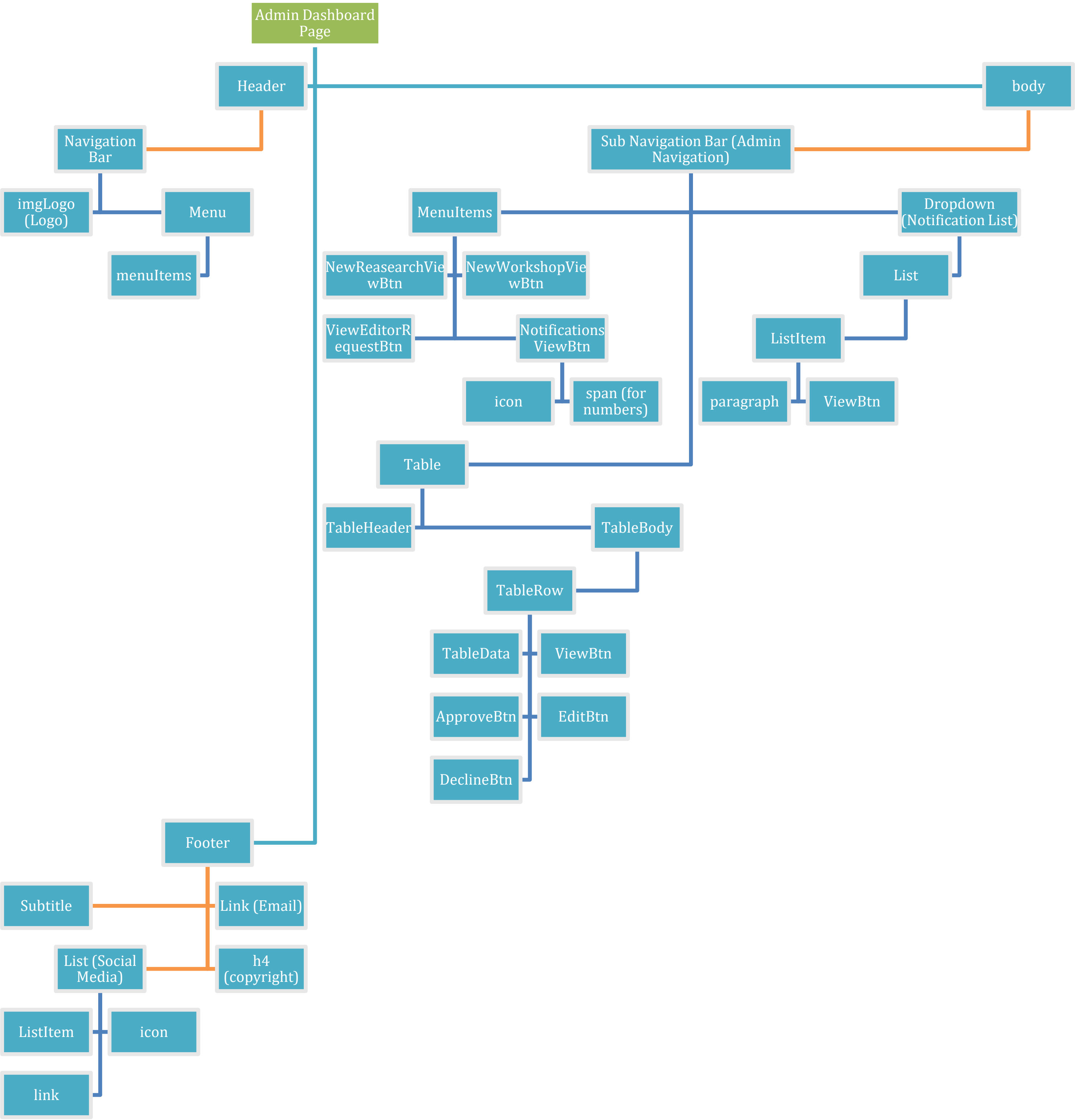
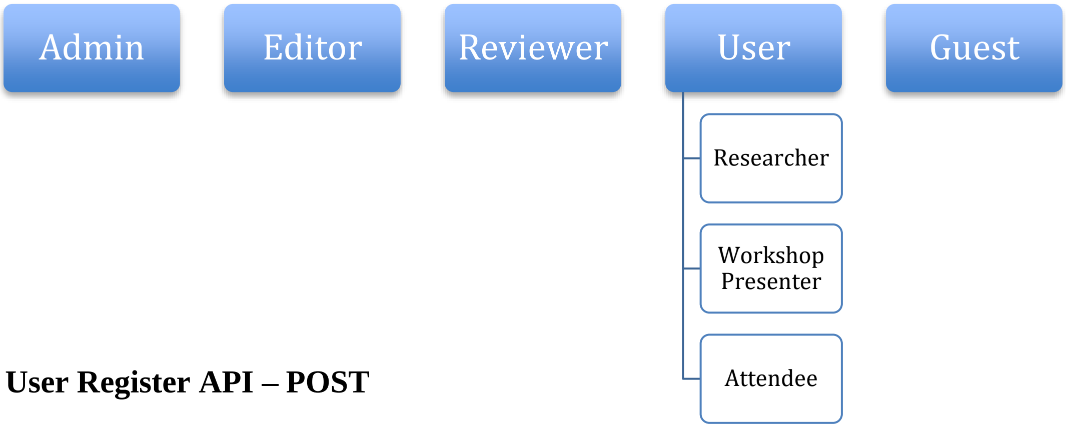


Figure 4.6: admin dashboard component tree

5. Usage of Rest APIs

Conference System APIs



5.1 User Register API – POST

```
{
        first_name: “”,
        last_name: “”,
        username: “”,
        contact_no: “”,
        email: “”,
        type: “”, -> [{id: “”, name: “Researcher” }]
        password:””
}
```

5.2 User Login API – GET (Web Token)

```
{
        username: “”,
        password:””,
}
```

5.2.1 Profile -> GET/POST/PATCH

5.3 Researcher (User)

5.3.1 Upload Research API – POST

```
{

        title: “”,
        description: “”,
        topic_interests:[“”, “”, “”], <- Custom Types also available and Need to show topics where
        uploaded by admin as a dropdown
        pdf: “”,
        authors: [“”,””,””],
        status: “”, <- (not reviewed yet, declined, pending payment, approved, complete)

}
```

5.3.2 View research of specific user (Researcher) API – GET (Send UserID in Header)

```
{

        title: “”,
        description: “”,
        topic_interests:[“”, “”, “”], <- Custome Types also available and Need to show topics where
        uploaded by admin as a dropdown
        pdf: “”,
        authors: “”,
        status: “approved”,

}
```

5.3.3 Pay for approved Research API – POST (Send UserID in Header)

```
{

        researchID: “”,
        paymentAmount:””,
        paymentStatus:””,
}
```

5.4 Workshop Presenter (User)

5.4.1 Upload New Workshop API – POST

```
{

        title: “”,
        description: “”,
        flyer: “”,
        resource_persons: “”,
        held_date: “”,
        start_time: “”,
        end_time:””,
        status: “”, <- this field not filled by user. This field update when reviewer review that research.

}
```

```
[
        {
                Date:”2020/01/01”,
                times: [{starttime: “”, endtime:””}, [{starttime: “”, endtime:””}]
        }, [], [], []
]
```

5.4.2 GET Workshop by specific user (Workshop Presenter) API – GET

```
{
        _id:””
        title: “”,
        description: “”,
        flyer: “”,
        resource_persons: “”,
        held_date: “”,
        status: “”,
}
```

5.5 Attendees (User)

5.5.1 Register API -> Payment Gateway -> Send Confirmation email/SMS with attendance id

5.6 Reviewer

5.6.1 View All Research docs uploaded by researchers - GET

```
{
        _id:””,
        title: “”,
        description: “”,
        topic_interests:[“”, “”, “”],
        pdf: “”,
        authors: “”,
        status: “”, (pending review, pending payment, approved, complete)
}
```

5.6.2 Accept or decline a research - PATCH

```
{
        research_id: “”
        status: “accept/decline/pay/complete”,
}
```

5.6.3 View All Workshops uploaded by workshop presenters - GET

```
{
        _id: “”,
        title: “”,
        description: “”,
        flyer: “”,
        resource_persons: “”,
        held_date: “”,
        status: “”,
}
```

5.6.4 Accept or decline a Workshops - PATCH

```
{
        Id: “”
        status: “accept/decline”,
}
```

5.7 Editor

5.7.1 View All Accepted Research complete (paid and accepted) by Reviewer - GET

```
{
        _id:”1234567”,
        title: “”,
        description: “”,
        topic_interests:[“”, “”, “”],
        pdf: “”,
        authors: “”,
        status: “”,

}
```

5.7.2 Create new Research Notice – POST/ PATCH

```
{
        Research_id: “1234567”,
        title: “”,  <- This title provide by editor and this shows in the web interface
        description: “”, <- Description provide by editor and this shows in the web interface
        topic_interests:[“”, “”, “”],
        pdf: “”,
        authors: “”,
        status: “pending”,  <- this status for admin. If he accepts, this status changed to accepted. then this
        notice shows in the website.
}
```

5.7.3 View All Accepted Workshops approved by Reviewer – GET

```
{
        _id: “”,
        title: “”,
        description: “”,
        flyer: “”,
        resource_persons: “”,
        held_date: “”,
        status: “”,

}
```

5.7.4 Create new Workshop Notice - POST

```
{
        workshop_id: “”,
        title: “”,  <- This title provide by editor and this shows in the web interface
        description: “”, <- Description provide by editor and this shows in the web interface
        flyer: “”,
        resource_persons: “”,
        held_date: “”,
        start_time: “”,
        end_time:””,
        status: “pending”,  <- this status for admin. If he accepts, this status changed to accepted. then this
        notice shows in the website.
}
```

5.7.5 Create Header – POST

```
    {
        conferenceName,
        date,
        time,
        venue,
        about,
        status:””
    }
```

5.7.6 Create News – POST

```
{
    date,
    details,
    news,
    status:””
}
```

5.7.7 Create Keynote Data – POST

```
{
    name,
    university,
    profile_img,
    about,
    status:””
}
```

5.7.8 Create Gallery Data – POST

```
{
    Image:””
}
```

5.7.9 Create Sponsor By Data – POST

```
{
    logo:””
}
```

5.8 Administrator

5.8.1 View All research notices publish by editor - GET

```
{
        _id:””
        Research_id: “1234567”,
        title: “”,  <- This title provide by editor and this shows in the web interface
        description: “”, <- Description provide by editor and this shows in the web interface
        topic_interests:[“”, “”, “”],
        pdf: “”,
        authors: “”,
        status: “pending”,  <- this status for admin. If he accepts, this status changed to accepted. then this notice shows in the website.
}
```

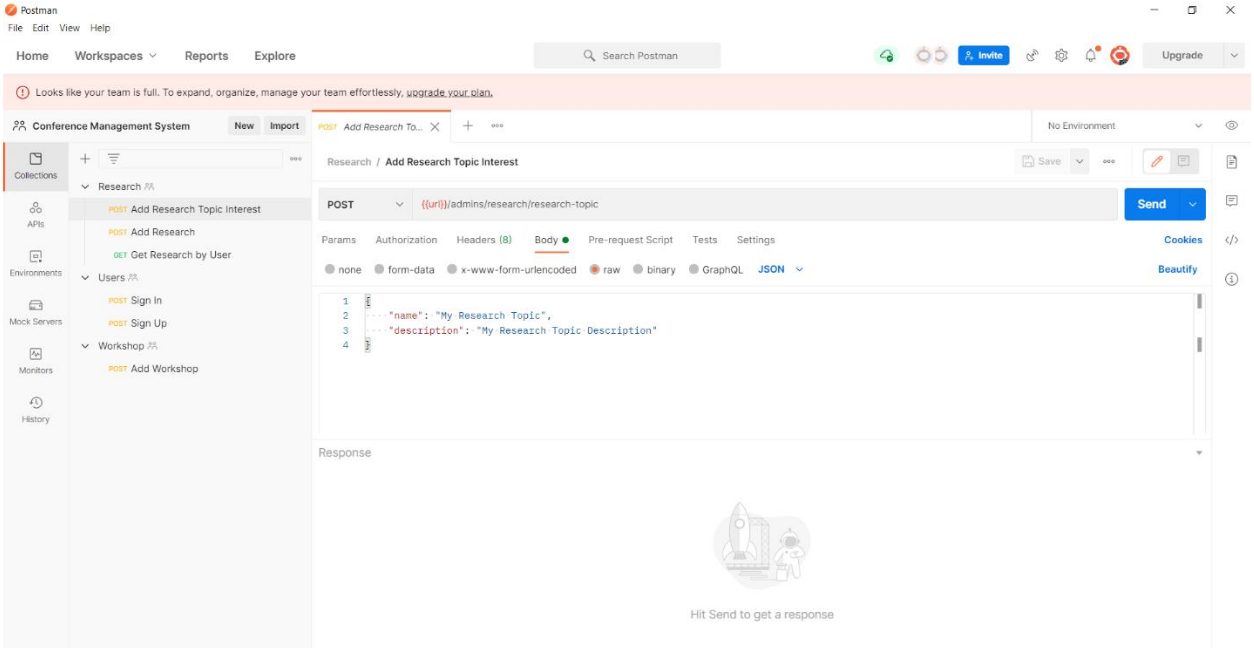
5.8.2 View All research notices publish by editor - PATCH

```
{
        _id:””
        Status:”” <-approve
}
```

5.8.3 View All workshop notices publish by editor - PATCH

```
{
        _id:””
        Status:”” <-approve
}
```

6. Rest API Post Man



Postman
File Edit View Help
Home Workspaces Reports Explore
Search Postman
Invite
Upgrade
Looks like your team is full. To expand, organize, manage your team effortlessly, upgrade your plan.
Conference Management System
New Import
Collections
APIs
Environments
Mock Servers
Monitors
History
Research
POST Add Research Topic Interest
POST Add Research
GET Get Research by User
Users
POST Sign In
POST Sign Up
Workshop
POST Add Workshop
POST Add Research To...
No Environment
Save
Send
Params Authorization Headers (8) Body Pre-request Script Tests Settings
Cookies
Beautify
Response
Hit Send to get a response

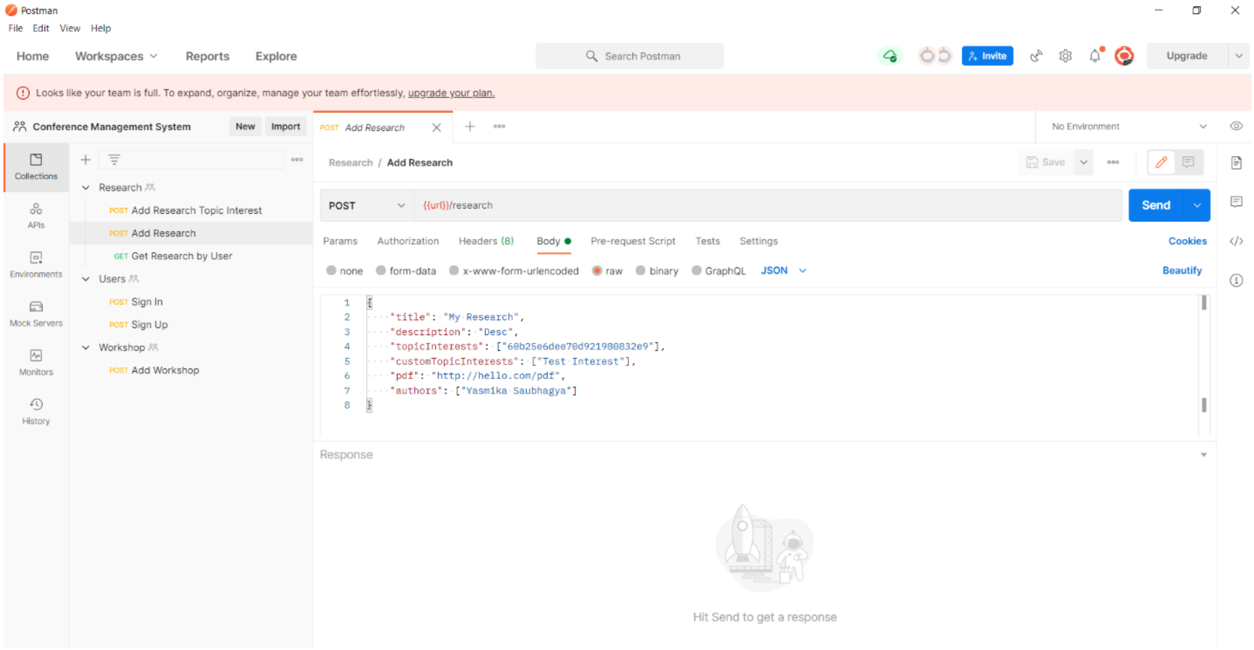
Research / Add Research Topic Interest

POST {{url}}/admins/research/research-topic

Body: ○ none ○ form-data ○ x-www-form-urlencoded ● raw ○ binary ○ GraphQL JSON

```
{
    "name": "My Research Topic",
    "description": "My Research Topic Description"
}
```

Figure 6.1: add research topic interests api end point



Postman
File Edit View Help
Home Workspaces Reports Explore
Search Postman
Invite
Upgrade
Looks like your team is full. To expand, organize, manage your team effortlessly, upgrade your plan.
Conference Management System
New Import
Collections
APIs
Environments
Mock Servers
Monitors
History
Research
POST Add Research Topic Interest
POST Add Research
GET Get Research by User
Users
POST Sign In
POST Sign Up
Workshop
POST Add Workshop
POST Add Research
No Environment
Save
Send
Params Authorization Headers (8) Body Pre-request Script Tests Settings
Cookies
Beautify
Response
Hit Send to get a response

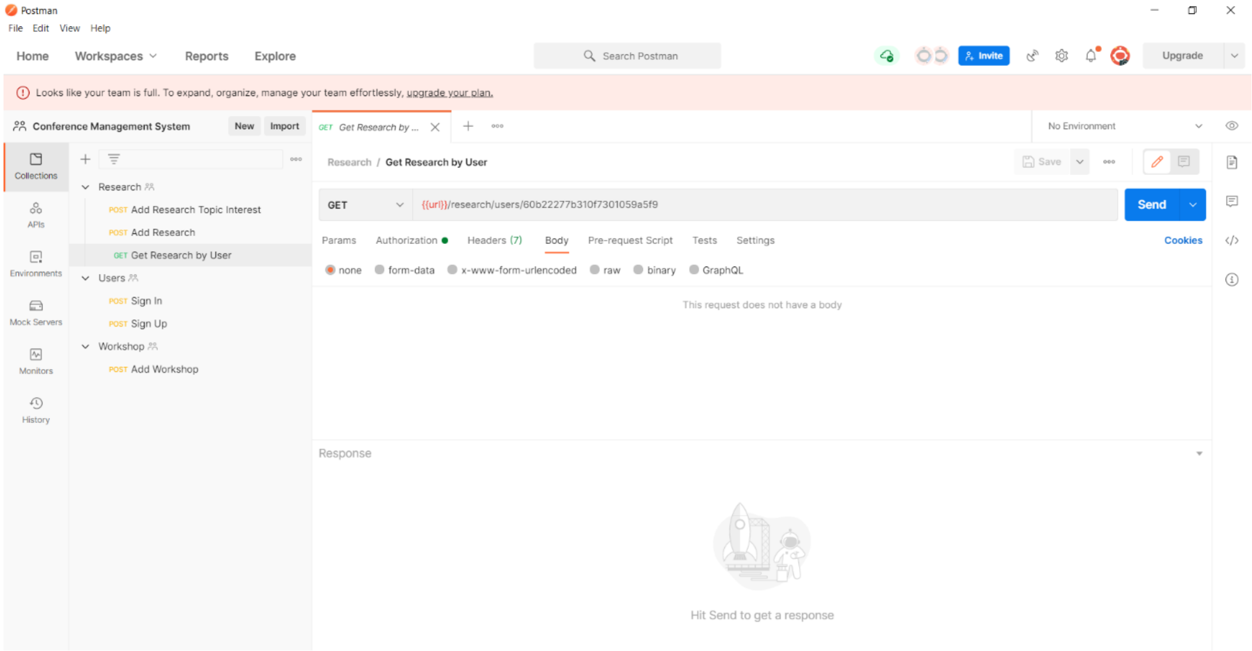
Research / Add Research

POST {{url}}/research

Body: ○ none ○ form-data ○ x-www-form-urlencoded ● raw ○ binary ○ GraphQL JSON

```
{
    "title": "My Research",
    "description": "Desc",
    "topicInterests": ["60b25e6dee70d921980832e9"],
    "customTopicInterests": ["Test Interest"],
    "pdf": "http://hello.com/pdf",
    "authors": ["Yasmika Saubhagya"]
}
```

Figure 6.2: add research api end point



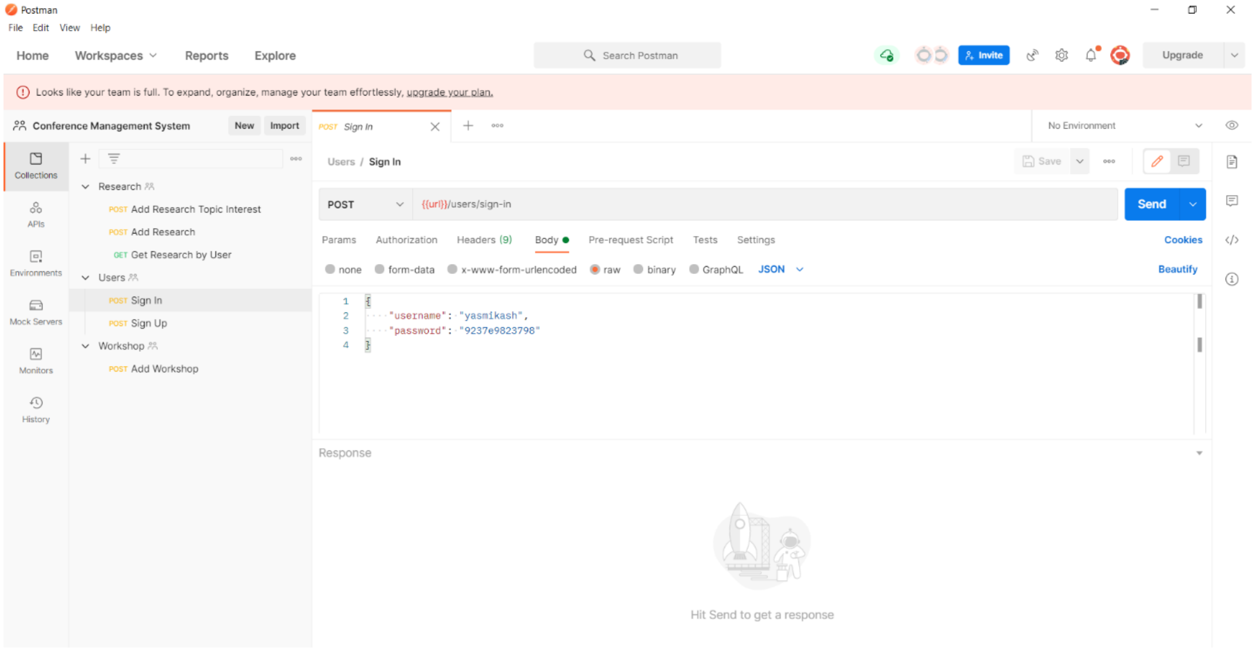
Looks like your team is full. To expand, organize, manage your team effortlessly, upgrade your plan.

Research / Get Research by User

GET {{url}}/research/users/60b22277b310f7301059a5f9

This request does not have a body

Figure 6.3: get research by user api end point



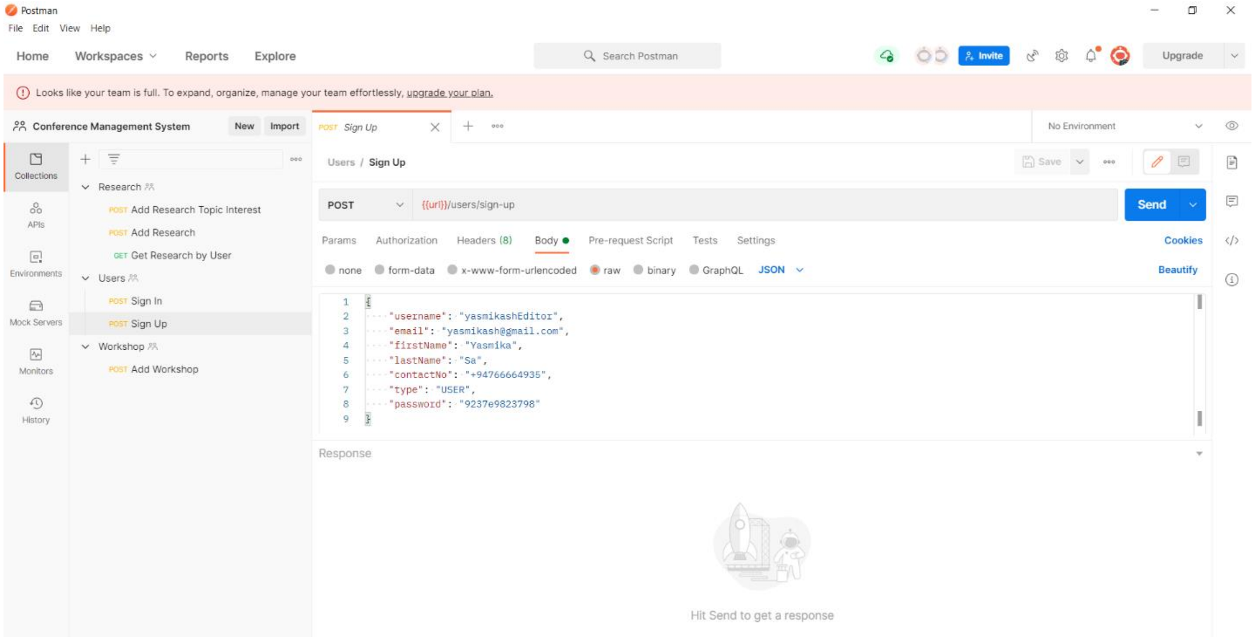
Looks like your team is full. To expand, organize, manage your team effortlessly, upgrade your plan.

Users / Sign In

POST {{url}}/users/sign-in

```
{
    "username": "yasmikash",
    "password": "9237e9823798"
}
```

Figure 6.4: user sign in api end point

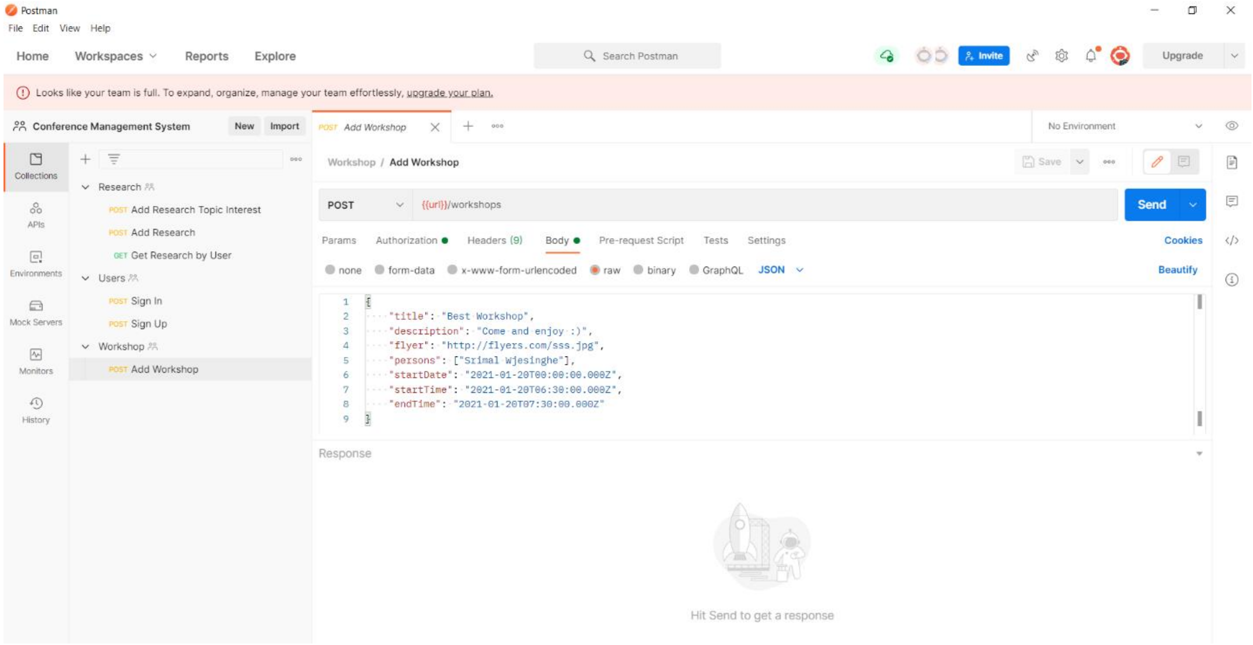

Postman
File Edit View Help
Home Workspaces Reports Explore
Search Postman
Invite
Upgrade
Looks like your team is full. To expand, organize, manage your team effortlessly, upgrade your plan.
Conference Management System New Import
Collections
APIs
Environments
Mock Servers
Monitors
History
Research
POST Add Research Topic Interest
POST Add Research
GET Get Research by User
Users
POST Sign In
POST Sign Up
Workshop
POST Add Workshop
POST Sign Up
No Environment
Users / Sign Up
Save
Send
Params Authorization Headers (8) Body Pre-request Script Tests Settings
Cookies
Beautify
Response
Hit Send to get a response

POST {{url}}/users/sign-up

○ none ○ form-data ○ x-www-form-urlencoded ● raw ○ binary ○ GraphQL JSON

```
{
    "username": "yasmikashEditor",
    "email": "yasmikash@gmail.com",
    "firstName": "Yasmika",
    "lastName": "Sa",
    "contactNo": "+94766664935",
    "type": "USER",
    "password": "9237e9823798"
}
```

Figure 6.5: user sign up api end point


Postman
File Edit View Help
Home Workspaces Reports Explore
Search Postman
Invite
Upgrade
Looks like your team is full. To expand, organize, manage your team effortlessly, upgrade your plan.
Conference Management System New Import
Collections
APIs
Environments
Mock Servers
Monitors
History
Research
POST Add Research Topic Interest
POST Add Research
GET Get Research by User
Users
POST Sign In
POST Sign Up
Workshop
POST Add Workshop
POST Add Workshop
No Environment
Workshop / Add Workshop
Save
Send
Params Authorization Headers (9) Body Pre-request Script Tests Settings
Cookies
Beautify
Response
Hit Send to get a response

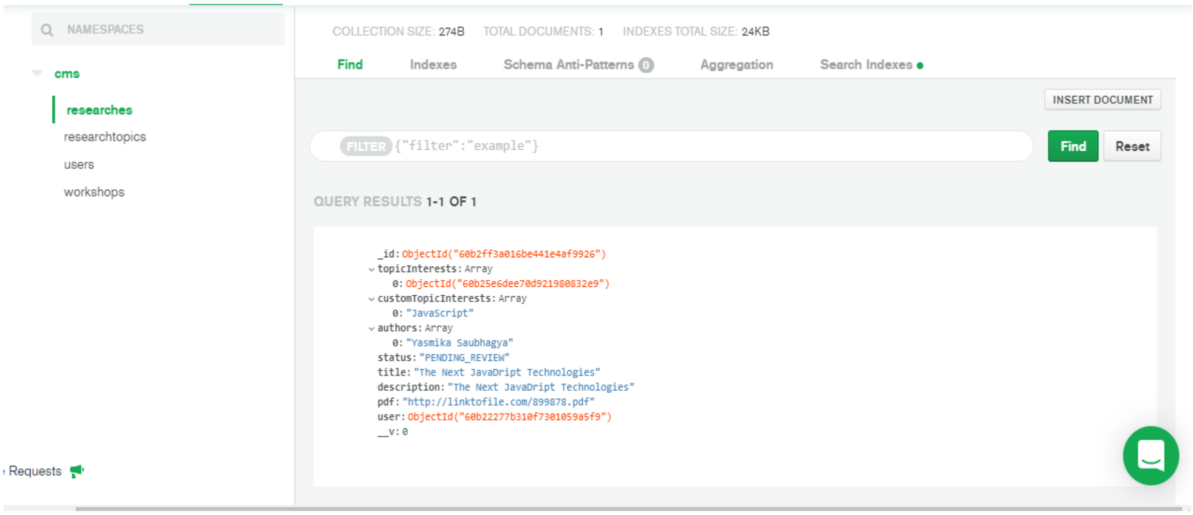
POST {{url}}/workshops

○ none ○ form-data ○ x-www-form-urlencoded ● raw ○ binary ○ GraphQL JSON

```
{
    "title": "Best Workshop",
    "description": "Come and enjoy :)",
    "flyer": "http://flyers.com/sss.jpg",
    "persons": ["Srimal Wjesinghe"],
    "startDate": "2021-01-20T00:00:00.000Z",
    "startTime": "2021-01-20T06:30:00.000Z",
    "endTime": "2021-01-20T07:30:00.000Z"
}
```

Figure 6.6: add workshop api end point

7. Mongo DB Collections



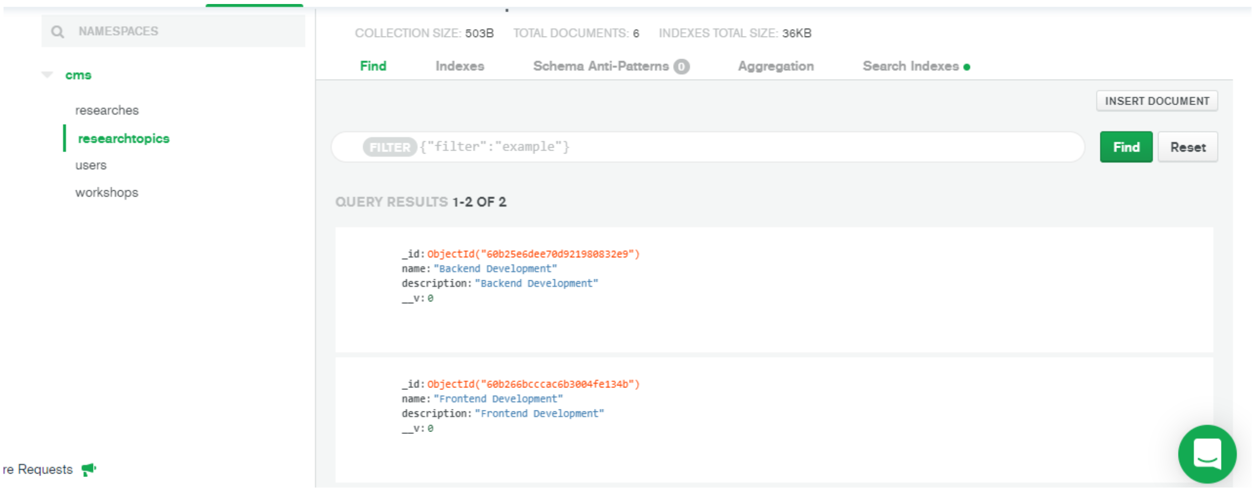
COLLECTION SIZE: 274B TOTAL DOCUMENTS: 1 INDEXES TOTAL SIZE: 24KB

{"filter":"example"}

QUERY RESULTS 1-1 OF 1

```
  _id: ObjectId("60b2ff3a016be441e4af9926")
⌄ topicInterests: Array
    0: ObjectId("60b25e6dee70d921980832e9")
⌄ customTopicInterests: Array
    0: "JavaScript"
⌄ authors: Array
    0: "Yasmika Saubhagya"
  status: "PENDING_REVIEW"
  title: "The Next JavaDript Technologies"
  description: "The Next JavaDript Technologies"
  pdf: "http://linktofile.com/899878.pdf"
  user: ObjectId("60b22277b310f7301059a5f9")
  __v: 0
```

Figure 7.1:researches mongo db collection



COLLECTION SIZE: 503B TOTAL DOCUMENTS: 6 INDEXES TOTAL SIZE: 36KB

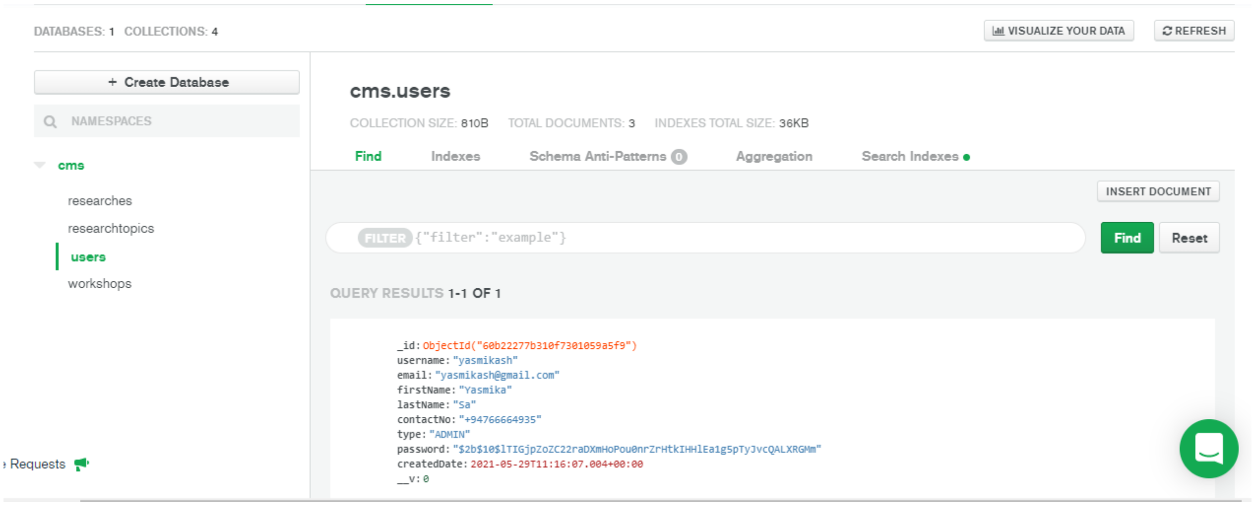
{"filter":"example"}

QUERY RESULTS 1-2 OF 2

```
_id: ObjectId("60b25e6dee70d921980832e9")
name: "Backend Development"
description: "Backend Development"
__v: 0
```

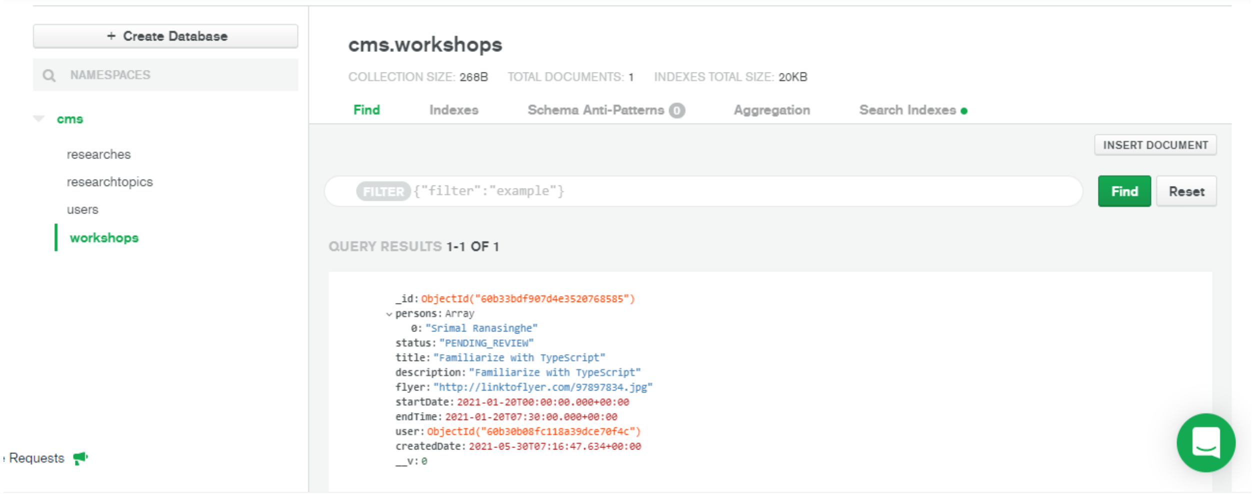
```
_id: ObjectId("60b266bcccac6b3004fe134b")
name: "Frontend Development"
description: "Frontend Development"
__v: 0
```

Figure 7.2: research topics mongo db collection



```
_id: ObjectId("60b22277b310f7301059a5f9")
username: "yasmikash"
email: "yasmikash@gmail.com"
firstName: "Yasmika"
lastName: "Sa"
contactNo: "+94766664935"
type: "ADMIN"
password: "$2b$10$lTIGjpZoZC22raDXmHoPou0nrZrHtkIHHlEa1g5pTyJvcQALXRGMm"
createdDate: 2021-05-29T11:16:07.004+00:00
__v: 0
```

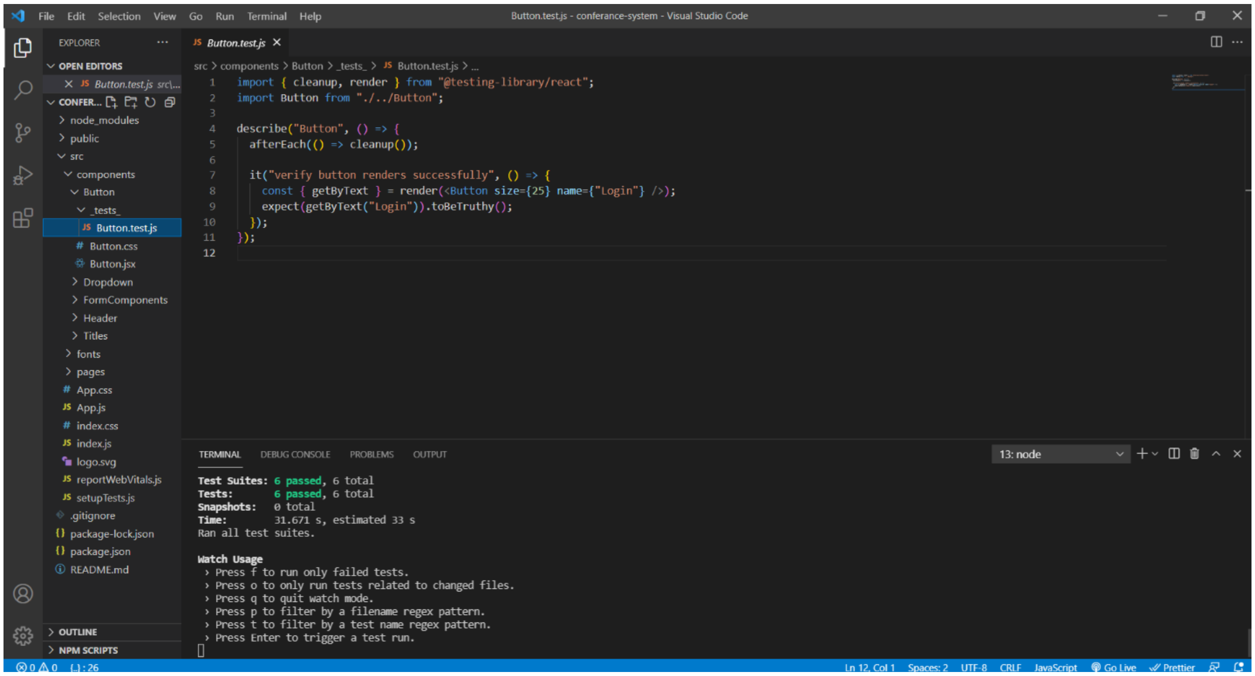
Figure 7.3: users mongo db collection



```
  _id: ObjectId("60b33bdf907d4e3520768585")
⌄ persons: Array
    0: "Srimal Ranasinghe"
  status: "PENDING_REVIEW"
  title: "Familiarize with TypeScript"
  description: "Familiarize with TypeScript"
  flyer: "http://linktoflyer.com/97897834.jpg"
  startDate: 2021-01-20T00:00:00.000+00:00
  endTime: 2021-01-20T07:30:00.000+00:00
  user: ObjectId("60b30b08fc118a39dce70f4c")
  createdDate: 2021-05-30T07:16:47.634+00:00
  __v: 0
```

Figure 7.4: workshops mongo db collection

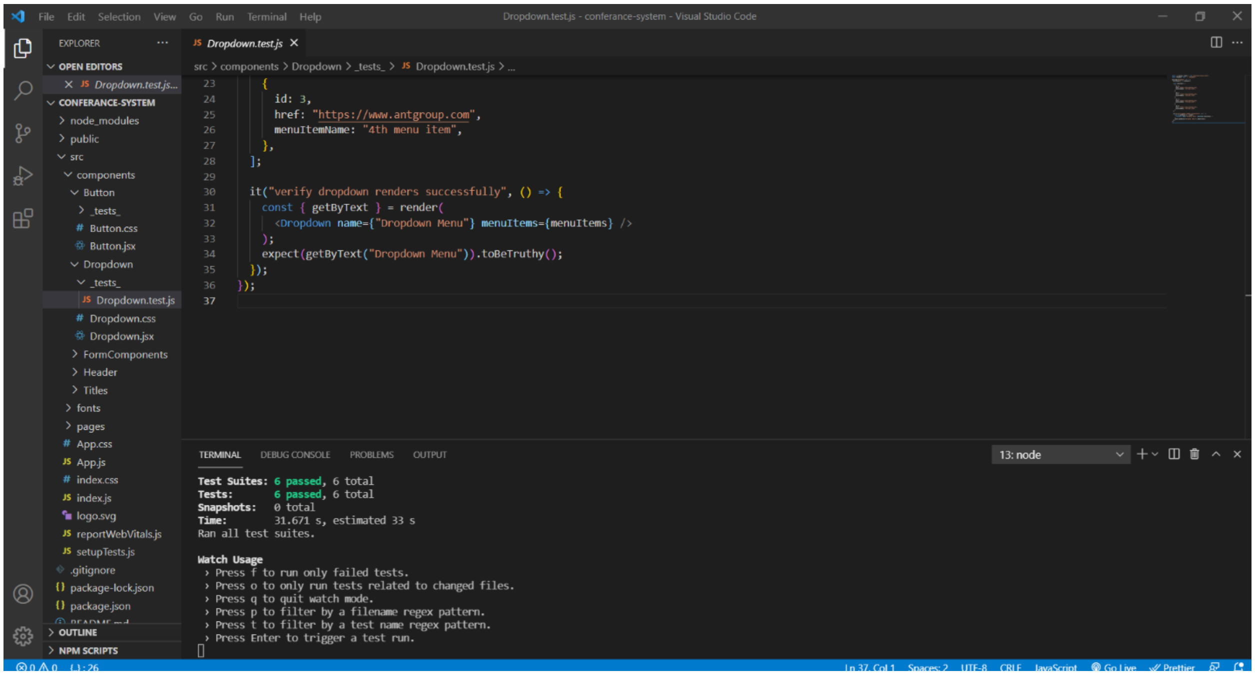
8. Test Cases



```javascript
import { cleanup, render } from "@testing-library/react";
import Button from "./../Button";

describe("Button", () => {
  afterEach(() => cleanup());

  it("verify button renders successfully", () => {
    const { getByText } = render(<Button size={25} name={"Login"} />);
    expect(getByText("Login")).toBeTruthy();
  });
});

```

```
Test Suites: 6 passed, 6 total
Tests:       6 passed, 6 total
Snapshots:   0 total
Time:        31.671 s, estimated 33 s
Ran all test suites.

Watch Usage
 › Press f to run only failed tests.
 › Press o to only run tests related to changed files.
 › Press q to quit watch mode.
 › Press p to filter by a filename regex pattern.
 › Press t to filter by a test name regex pattern.
 › Press Enter to trigger a test run.
```

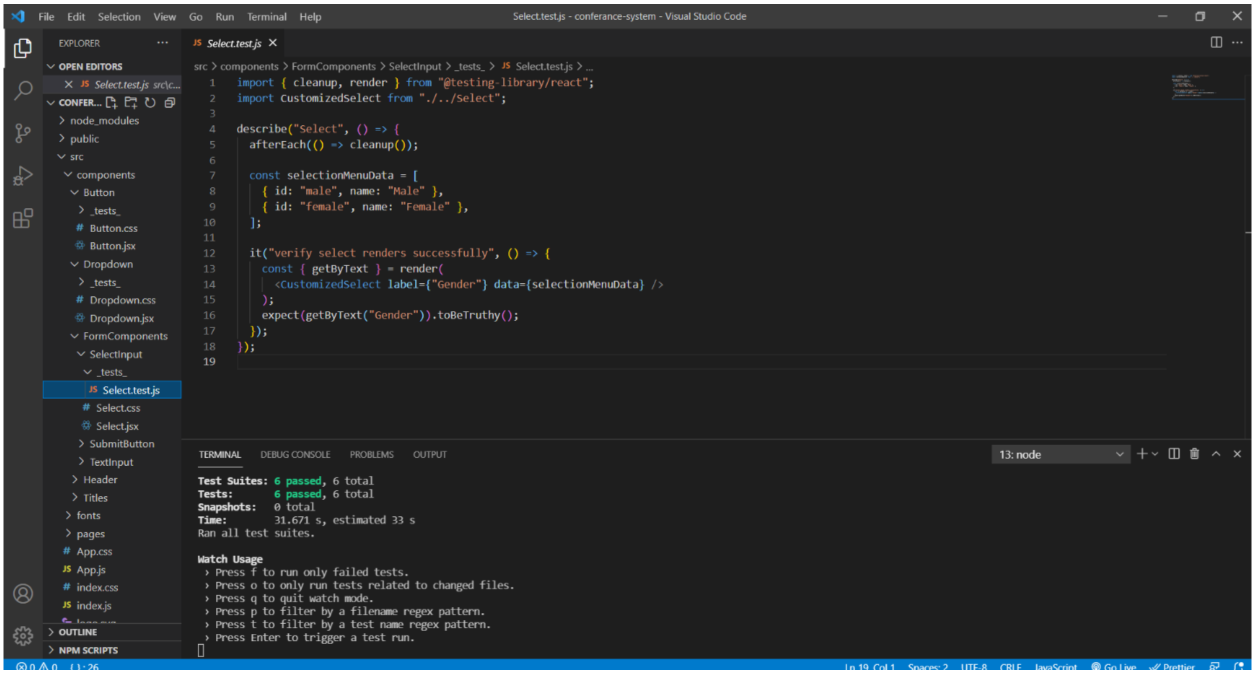
Figure 8.1: test case number one



```javascript
    {
      id: 3,
      href: "https://www.antgroup.com",
      menuItemName: "4th menu item",
    },
  ];

  it("verify dropdown renders successfully", () => {
    const { getByText } = render(
      <Dropdown name={"Dropdown Menu"} menuItems={menuItems} />
    );
    expect(getByText("Dropdown Menu")).toBeTruthy();
  });
});

```

```
Test Suites: 6 passed, 6 total
Tests:       6 passed, 6 total
Snapshots:   0 total
Time:        31.671 s, estimated 33 s
Ran all test suites.

Watch Usage
 › Press f to run only failed tests.
 › Press o to only run tests related to changed files.
 › Press q to quit watch mode.
 › Press p to filter by a filename regex pattern.
 › Press t to filter by a test name regex pattern.
 › Press Enter to trigger a test run.
```

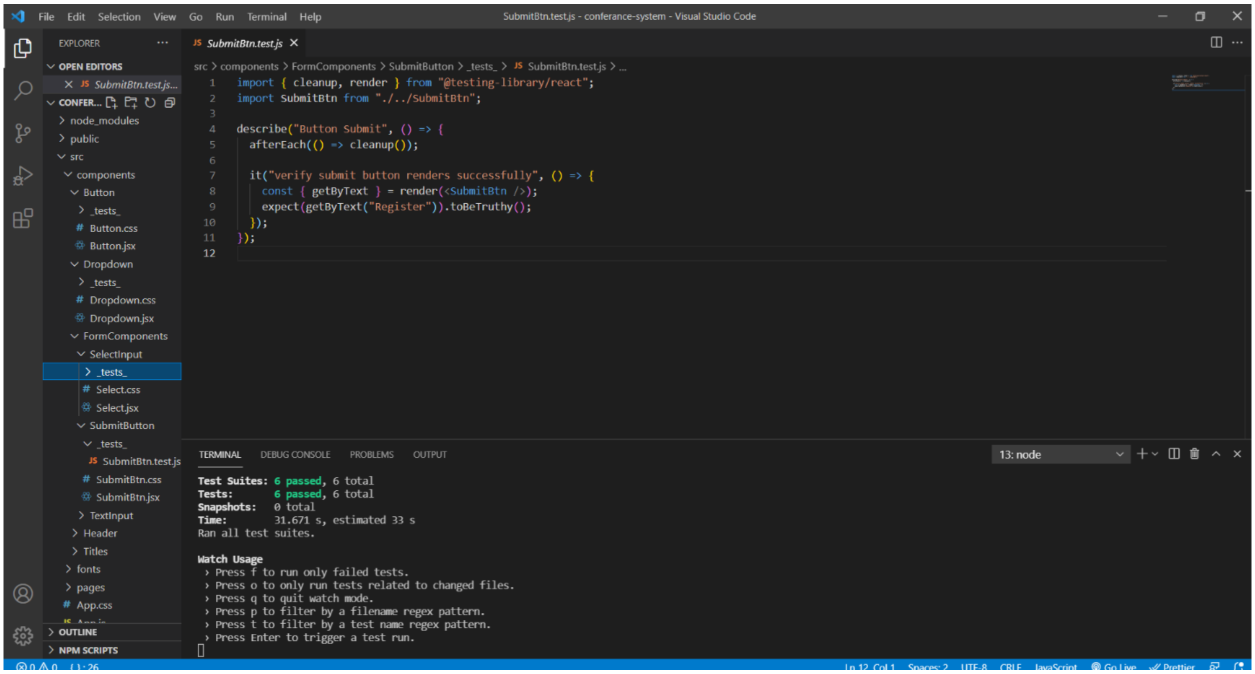
Figure 8.2: test case number two


File Edit Selection View Go Run Terminal Help
Select.test.js - conferance-system - Visual Studio Code
EXPLORER
OPEN EDITORS
Select.test.js
src > components > FormComponents > SelectInput > _tests_ > Select.test.js > ...
TERMINAL DEBUG CONSOLE PROBLEMS OUTPUT
13: node
OUTLINE
NPM SCRIPTS
Ln 19, Col 1 Spaces: 2 UTF-8 CRLF JavaScript Go Live Prettier

```javascript
import { cleanup, render } from "@testing-library/react";
import CustomizedSelect from "./../Select";

describe("Select", () => {
  afterEach(() => cleanup());

  const selectionMenuData = [
    { id: "male", name: "Male" },
    { id: "female", name: "Female" },
  ];

  it("verify select renders successfully", () => {
    const { getByText } = render(
      <CustomizedSelect label={"Gender"} data={selectionMenuData} />
    );
    expect(getByText("Gender")).toBeTruthy();
  });
});
```

```
Test Suites: 6 passed, 6 total
Tests:       6 passed, 6 total
Snapshots:   0 total
Time:        31.671 s, estimated 33 s
Ran all test suites.

Watch Usage
 › Press f to run only failed tests.
 › Press o to only run tests related to changed files.
 › Press q to quit watch mode.
 › Press p to filter by a filename regex pattern.
 › Press t to filter by a test name regex pattern.
 › Press Enter to trigger a test run.
```

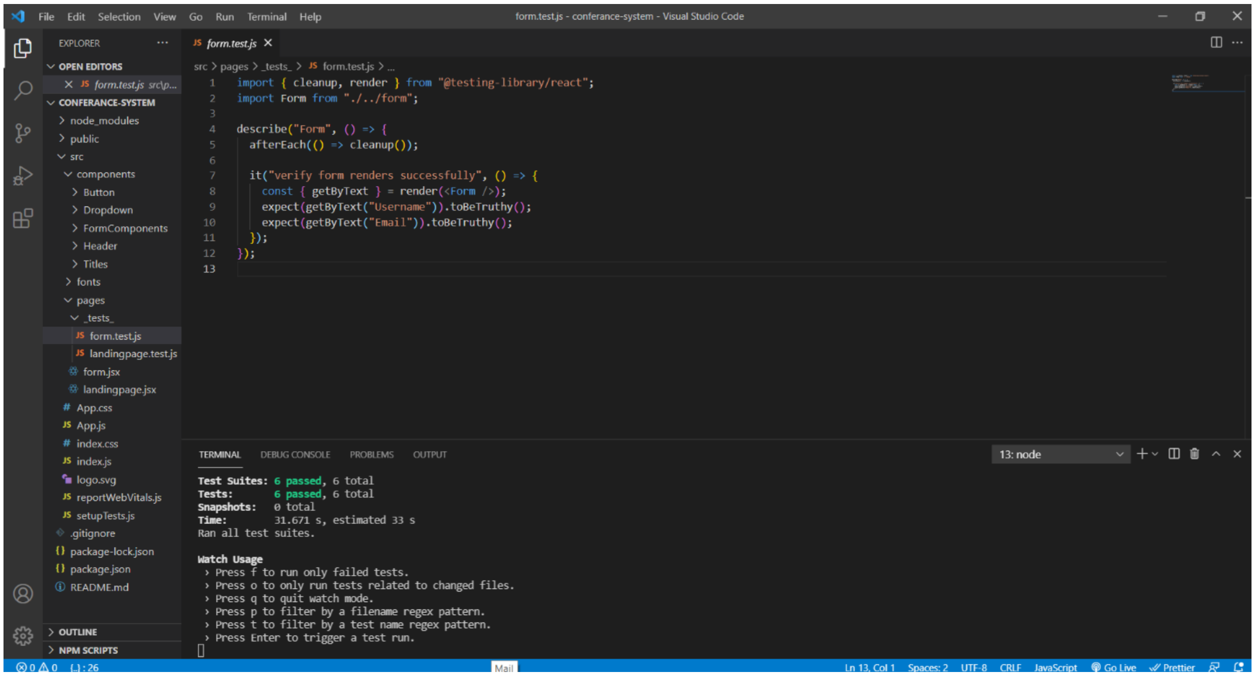
Figure 8.3: test case number three


File Edit Selection View Go Run Terminal Help
SubmitBtn.test.js - conferance-system - Visual Studio Code
EXPLORER
OPEN EDITORS
SubmitBtn.test.js
src > components > FormComponents > SubmitButton > _tests_ > SubmitBtn.test.js > ...
TERMINAL DEBUG CONSOLE PROBLEMS OUTPUT
13: node
OUTLINE
NPM SCRIPTS
Ln 12, Col 1 Spaces: 2 UTF-8 CRLF JavaScript Go Live Prettier

```javascript
import { cleanup, render } from "@testing-library/react";
import SubmitBtn from "./../SubmitBtn";

describe("Button Submit", () => {
  afterEach(() => cleanup());

  it("verify submit button renders successfully", () => {
    const { getByText } = render(<SubmitBtn />);
    expect(getByText("Register")).toBeTruthy();
  });
});
```

```
Test Suites: 6 passed, 6 total
Tests:       6 passed, 6 total
Snapshots:   0 total
Time:        31.671 s, estimated 33 s
Ran all test suites.

Watch Usage
 › Press f to run only failed tests.
 › Press o to only run tests related to changed files.
 › Press q to quit watch mode.
 › Press p to filter by a filename regex pattern.
 › Press t to filter by a test name regex pattern.
 › Press Enter to trigger a test run.
```

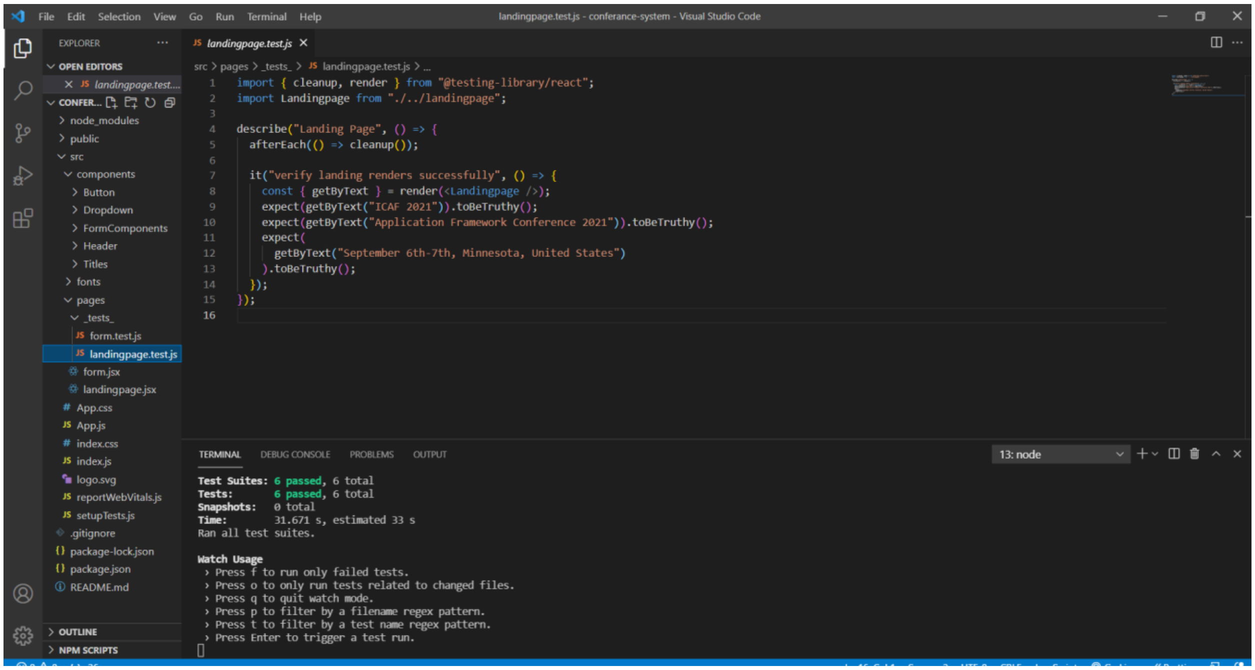
Figure 8.4: test case number four



```javascript
import { cleanup, render } from "@testing-library/react";
import Form from "./../form";

describe("Form", () => {
  afterEach(() => cleanup());

  it("verify form renders successfully", () => {
    const { getByText } = render(<Form />);
    expect(getByText("Username")).toBeTruthy();
    expect(getByText("Email")).toBeTruthy();
  });
});
```

```
Test Suites: 6 passed, 6 total
Tests:       6 passed, 6 total
Snapshots:   0 total
Time:        31.671 s, estimated 33 s
Ran all test suites.

Watch Usage
 › Press f to run only failed tests.
 › Press o to only run tests related to changed files.
 › Press q to quit watch mode.
 › Press p to filter by a filename regex pattern.
 › Press t to filter by a test name regex pattern.
 › Press Enter to trigger a test run.
```

Figure 8.5: test case number five



```javascript
import { cleanup, render } from "@testing-library/react";
import Landingpage from "./../landingpage";

describe("Landing Page", () => {
  afterEach(() => cleanup());

  it("verify landing renders successfully", () => {
    const { getByText } = render(<Landingpage />);
    expect(getByText("ICAF 2021")).toBeTruthy();
    expect(getByText("Application Framework Conference 2021")).toBeTruthy();
    expect(
      getByText("September 6th-7th, Minnesota, United States")
    ).toBeTruthy();
  });
});
```

```
Test Suites: 6 passed, 6 total
Tests:       6 passed, 6 total
Snapshots:   0 total
Time:        31.671 s, estimated 33 s
Ran all test suites.

Watch Usage
 › Press f to run only failed tests.
 › Press o to only run tests related to changed files.
 › Press q to quit watch mode.
 › Press p to filter by a filename regex pattern.
 › Press t to filter by a test name regex pattern.
 › Press Enter to trigger a test run.
```

Figure 8.6: test case number six