SE3020
Distribute Systems
3rd Year, 1st Semester

Assignment 2

Online train ticket reservation

Assignment Report

Submitted to

Sri Lanka Institute of Information Technology

In partial fulfillment of the requirements for the

Bachelor of Science Special Honors Degree in Information Technology

Submitted By: IT17137560 (R.S. Najeeb)

Table of Contents

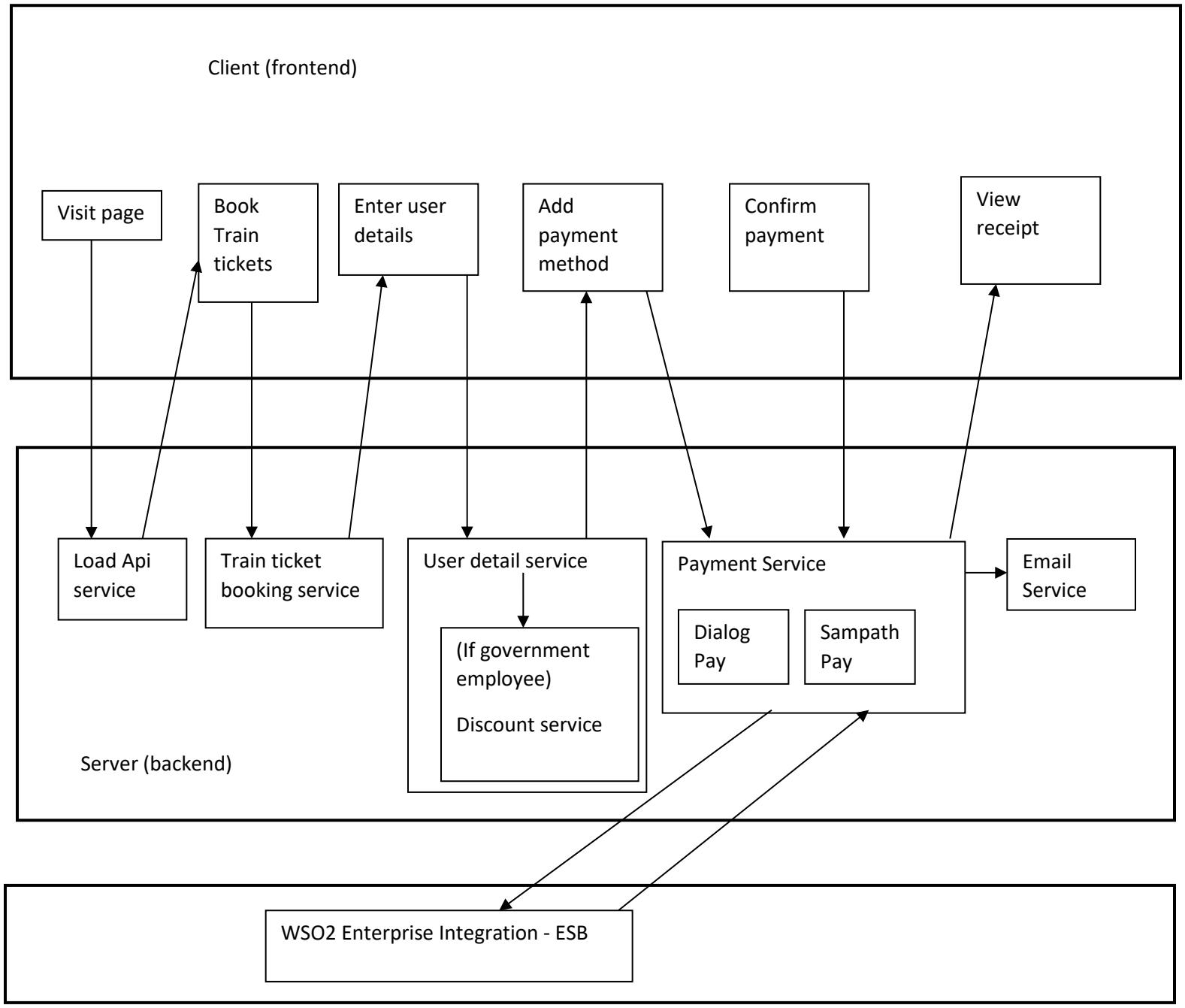
Introduction

The online ticket reservation application is built using different technologies. Front end client is developed using ReactJs and the backend server-side is developed using NodeJS, and MongoDB. The API can be tested via loopback API explorer services. For mails Node mailer is module is imported

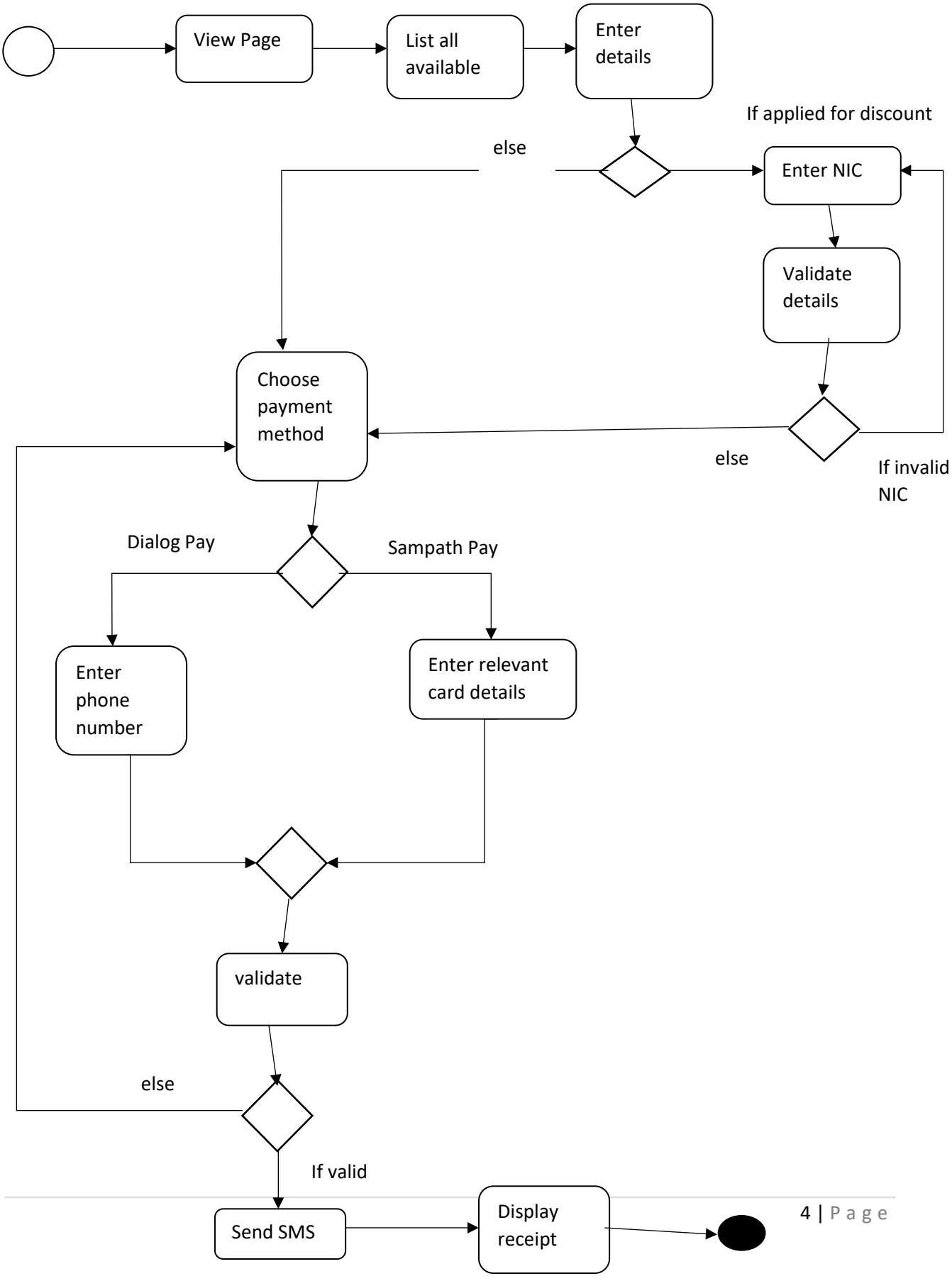
The user doesn't have to login or sign up to the system to use the application. As an when the application is loaded the train schedules are shown and a booking can be made. This train schedules are taken via a restful API named 'train-api' after a schedule is chosen the user is directed to a page to enter his/her details and make the payment either by card or dialog bill payment. Once the payment is made the receipt an email is sent to the user and if payments are done by the dialog bill an SMS is sent to the mobile.

Diagrams

Architectural Diagram



Flow Chart

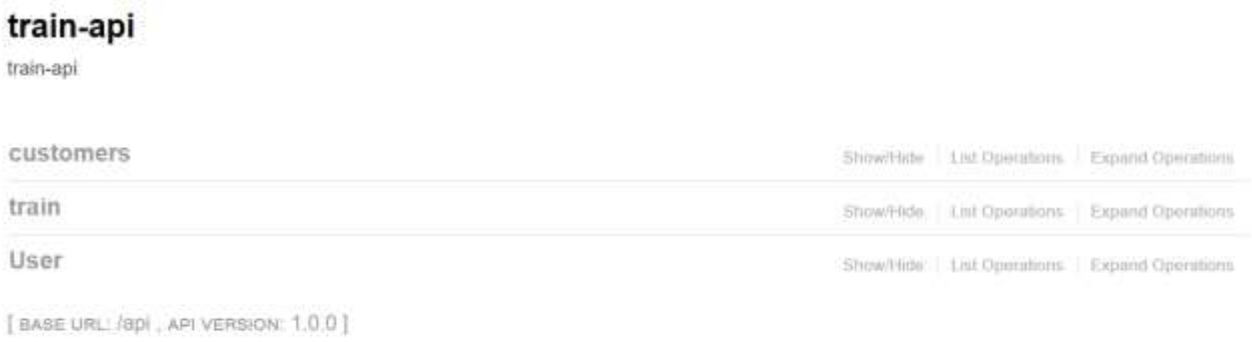

View Page
List all available
Enter details
If applied for discount
else
Enter NIC
Validate details
Choose payment method
else
If invalid NIC
Dialog Pay
Sampath Pay
Enter phone number
Enter relevant card details
validate
else
If valid
Send SMS
Display receipt

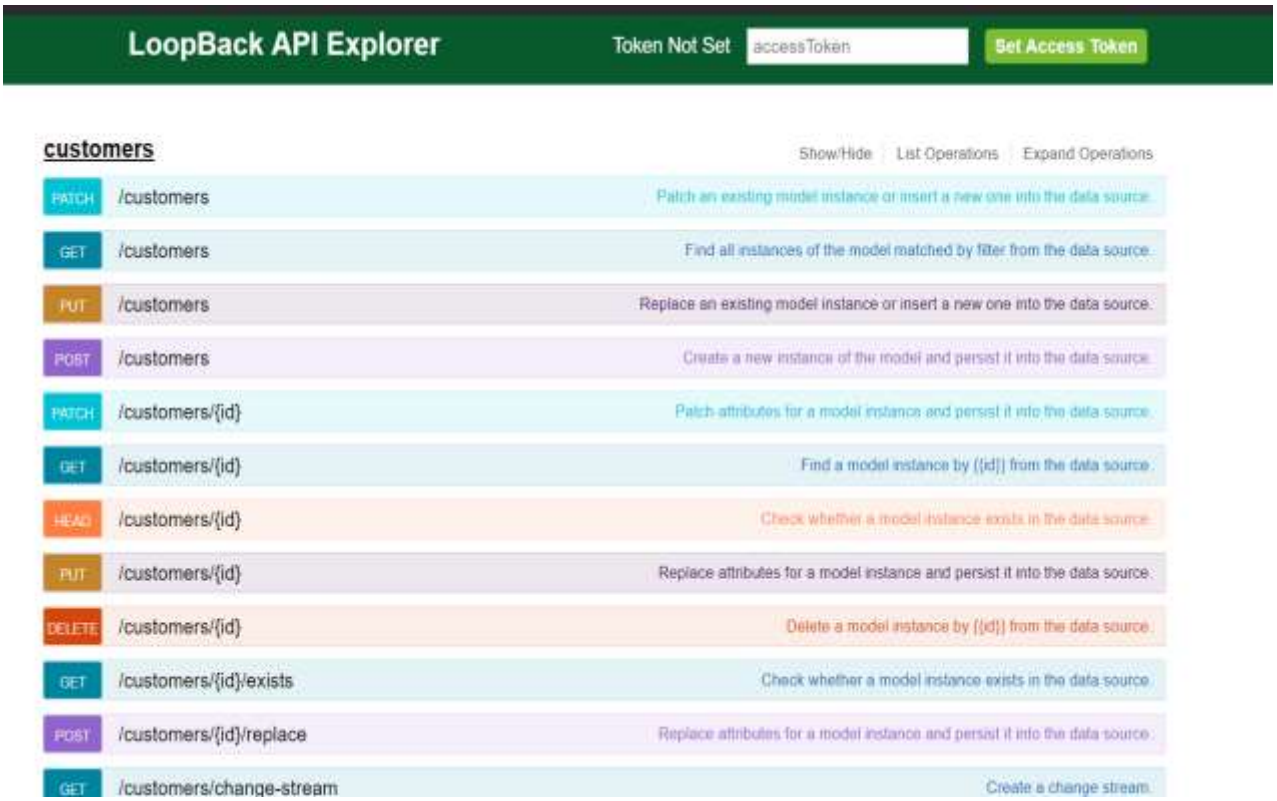
Authentication and Security

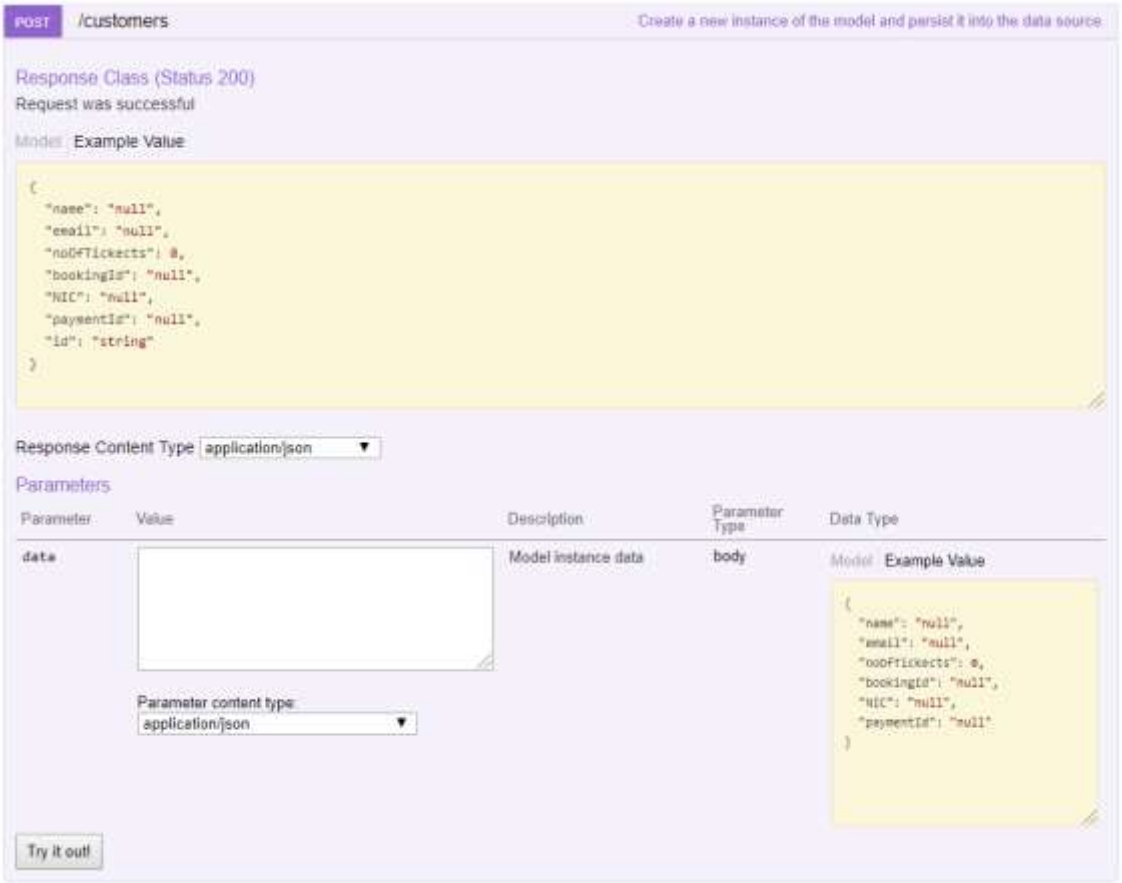
The user's card details are validated with the Card API, mobile number is validated through Phone API and the user is check whether he is an actual government employee and if not proper error messages are shown.

Testing the REST API

The train details, user details all can be tested with loopback service that is attached and all the server calls can be tested

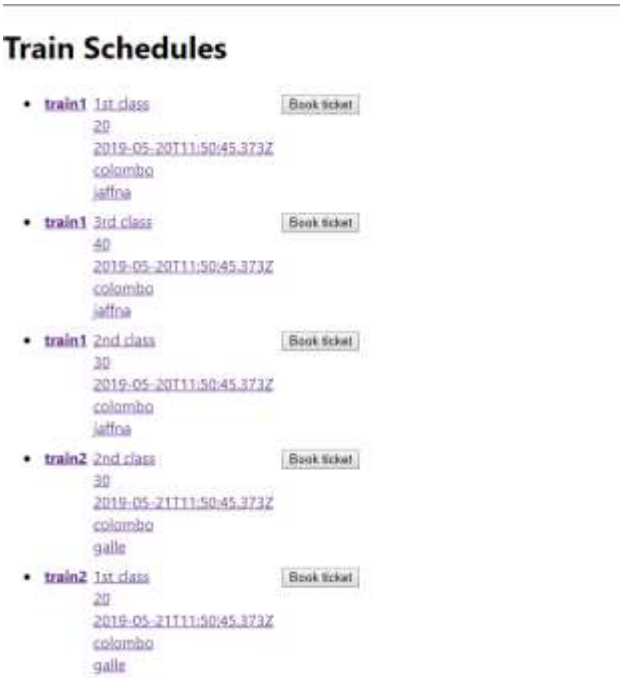


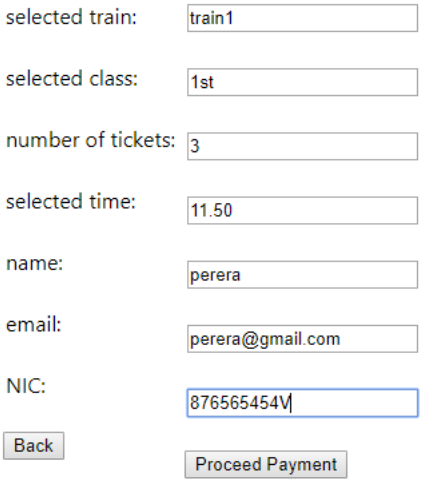

LoopBack API Explorer
Token Not Set
accessToken
Set Access Token
customers
Show/Hide
List Operations
Expand Operations
PATCH /customers
Patch an existing model instance or insert a new one into the data source.
GET /customers
Find all instances of the model matched by filter from the data source.
PUT /customers
Replace an existing model instance or insert a new one into the data source.
POST /customers
Create a new instance of the model and persist it into the data source.
PATCH /customers/{id}
Patch attributes for a model instance and persist it into the data source.
GET /customers/{id}
Find a model instance by {{id}} from the data source.
HEAD /customers/{id}
Check whether a model instance exists in the data source.
PUT /customers/{id}
Replace attributes for a model instance and persist it into the data source.
DELETE /customers/{id}
Delete a model instance by {{id}} from the data source.
GET /customers/{id}/exists
Check whether a model instance exists in the data source.
POST /customers/{id}/replace
Replace attributes for a model instance and persist it into the data source.
GET /customers/change-stream
Create a change stream.

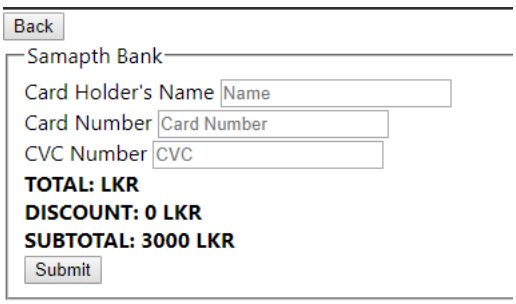

POST /customers
Create a new instance of the model and persist it into the data source.
Response Class (Status 200)
Request was successful
Model Example Value
{
"name": "null",
"email": "null",
"noOfTickects": 0,
"bookingId": "null",
"NIC": "null",
"paymentId": "null",
"id": "string"
}
Response Content Type application/json
Parameters
Parameter
Value
Description
Parameter Type
Data Type
data
Model instance data
body
Model Example Value
{
"name": "null",
"email": "null",
"noOfTickects": 0,
"bookingId": "null",
"NIC": "null",
"paymentId": "null"
}
Parameter content type
application/json
Try it out!

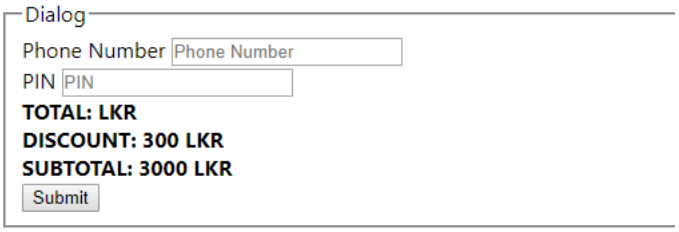
```
[
  {
    "name": "train1",
    "class": "1st class",
    "seats": 20,
    "availableSeats": 20,
    "bookedSeats": 0,
    "time": "2019-05-20T11:50:45.373Z",
    "price": 1000,
    "startStation": "colombo",
    "endStation": "jaffna",
    "id": "5ce294e2beb02e375471bec7"
  },
  {
    "name": "train1",
    "class": "3rd class",
    "seats": 40,
    "availableSeats": 40,
    "bookedSeats": 0,
    "time": "2019-05-20T11:50:45.373Z",
    "price": 200,
    "startStation": "colombo",
    "endStation": "jaffna",
    "id": "5ce29526beb02e375471bec9"
  },
  {
    "name": "train1",
    "class": "2nd class",
    "seats": 30,
    "availableSeats": 30,
    "bookedSeats": 0,
    "time": "2019-05-20T11:50:45.373Z",
    "price": 500,
    "startStation": "colombo",
    "endStation": "jaffna",
```

UI Design Screenshots









Code

The code can be accessed through https://github.com/rushdanajeeb/Assignment2.git

Frontend

App.js

```
import React from 'react';
import './App.css';
import Main from './components/Main';

const App = () => (
  <div>
    <Main/>
  </div>
)

export default App;
```

index.js

```
import React from 'react';
import ReactDOM from 'react-dom';
import './index.css';
import App from './App';
import * as serviceWorker from './serviceWorker';
import {BrowserRouter} from 'react-router-dom';


ReactDOM.render(
  <BrowserRouter>
  <App />
  </BrowserRouter>,
  document.getElementById('root')
);

// If you want your app to work offline and load faster, you can change
// unregister() to register() below. Note this comes with some pitfalls.
// Learn more about service workers: https://bit.ly/CRA-PWA
serviceWorker.unregister();
```

App.css

```
.trainBooking {
    margin-top: 10%;
    margin-bottom: 10%;
    margin-right: 35%;
    margin-left: 35%;
}

.a{
    padding: 5%;
    border: 5px lightsalmon;
    border-style: solid;
}
```

dialogPay.js

```
import React, {Component} from 'react';
import ReactDOM from "react-dom";
import '../App.css';

class DialogPay extends Component {

  constructor(props) {
    super(props);
    this.state = {
      name: this.props.name,
      discount:,
      subtot:,
      total:
    }
  }

  render() {
    var total = JSON.stringify(this.props.total);
    return (
      <div>
        <div >
          <form>
            <fieldset>
              <legend>Dialog</legend>
              <div><label>Phone Number</label> <input
                placeholder="Phone Number"
                type="text"/></div>
              <div><label>PIN</label> <input
                placeholder="PIN" type="text"/></div>
```

```
                    <div><label><b>TOTAL: {this.props.total} LKR</b></label>
                    </div>
                    <div><label><b>DISCOUNT: {this.state.discount} LKR</b></label></div>
                    <div><label><b>SUBTOTAL: {this.state.subtot} LKR</b></label></div>
                    <button onClick={() => this.home(document.getElementById("phone").value,
document.getElementById("pin").value, total, document.getElementById("loyalpoints").value)}>Submit
                    </button>
                  </fieldset>
                </form>
              </div>
            </div>
        )
    }
}

export default DialogPay;
```

Home.js

```
import React, {Component} from 'react';
import axios from 'axios/index';
import TrainItem from './TrainItem'
import '../App.css';

class Home extends Component{
    constructor(){
        super();
        this.state = {
            trains:[]
        }
    }

    componentWillMount() {
        this.showTrains();
    }

    showTrains(){
        axios.get('http://localhost:3000/api/trains')
            .then(
                response =>{
                    this.setState({trains: response.data},
                        () =>{
                        console.log(this.state);
                        })
```

```
            }
        )
    }

    render() {
        const trains = this.state.trains.map((train, i) =>{
            return(
                //<li className={"collection-item"}>{train.name}</li>

                <TrainItem key={train.id} item={train}/>
            )
        })
        return(
            <div>
                <h1>Train Schedules</h1>
                <ul className={"collection"}>
                    {trains}
                </ul>

            </div>
        )
    }
}

export default Home;
```

Main.js

```
import React from 'react';
import {Switch,Route} from 'react-router-dom';
import Home from './Home';
import TrainDetails from './TrainDetails';
import EditTrain from './EditTrain';
import ViewDetails from "./viewDetails";

const Main = () =>(
    <main>
        <Switch>
            <Route extract path={'/'} component={Home}/>
            <Route extract path={'/trains/view'} component={ViewDetails}/>
            <Route extract path={'/trains/:id'} component={TrainDetails}/>
            <Route extract path={'/trains/edit/:id'} component={EditTrain}/>
        </Switch>
    </main>
```

```
);

export default Main;
```

makePayment.js

```
import React, { Component } from "react";
import ReactDOM from "react-dom";
import ViewDetails from "./viewDetails";
import DialogPay from "./dialogPay";
import SampathPay from "./sampathPay";

class MakePayment extends Component {

   dialogpay = function (e) {
      ReactDOM.render(<DialogPay />, document.getElementById('root'));
   };

   sampathpay = function (e) {
      ReactDOM.render(<SampathPay />, document.getElementById('root'));
   };

   render() {
      return (
         <div className={"trainBooking"}>
            <div className={"a"}> Enter Payment method<br/><br/>
               <table>
                  <td>
                     <tr>
                        <input type={"radio"} id={"mobile"}  onClick={() => this.dialogpay()}/>Mobile Pay
                        <input type={"radio"} id={"sampath"} onClick={() => this.sampathpay()}/>Sampath Pay
                     </tr>
                     <br/><br/>
                     {/*<tr><button >Next</button></tr>*/}
                  </td>
               </table>
            </div>
         </div>
      );
   }
}


export default MakePayment;
```

sampathPay.js

```
import React, {Component} from 'react';
import ReactDOM from "react-dom";
import '../App.css';

class SampathPay extends Component {
   constructor(props) {
      super(props);
      this.state = {
         name: this.props.name,
         points: this.props.points,
         email: this.props.email,
         discount: 0,
         subtot: 3000,
         tot:2700
      }
   }

   home = function (name, cardNumber, cvc, total, points) {
      console.log(name + "--" + cardNumber + "--" + cvc + "--" + total);
      var subTotal = parseFloat(this.state.subtot);
      var data = {
         "email": this.state.email,
         "name": name,
         "cardNumber": cardNumber,
         "cvc": cvc,
         "total": total,
         "subtotal": subTotal
      };
      // ReactDOM.render(<App name={this.state.name} points={newPoints} email={this.state.email}/>,
document.getElementById('root'));
   }

   back() {
      // ReactDOM.render(<App name={this.state.name} points={this.state.points}
email={this.state.email}/>, document.getElementById('root'));
   }

   render() {
      var total = JSON.stringify(this.props.total);
      return ( /* *Sampath payment interface */ <div className="container">
         <div>
```

```
            <button type="submit" onClick={() => {
              this.back()
            }}>Back
            </button>
          </div>
          <div>
            <div >
              <form>
                <fieldset>
                  <legend>Samapth Bank</legend>
                  <div><label>Card Holder's Name</label> <input
                    placeholder="Name"
                    type="text"/></div>
                  <div><label>Card Number</label> <input
                    placeholder="Card Number" type="text"/></div>
                  <div ><label>CVC Number</label> <input
                    placeholder="CVC"
                    type="text"/></div>
                  <div><label><b>TOTAL: {this.props.total} LKR</b></label>
                  </div>
                  <div><label><b>DISCOUNT: {this.state.discount} LKR</b></label></div>
                  <div><label><b>SUBTOTAL: {this.state.subtot} LKR</b></label></div>
                  <button onClick={() => this.home(document.getElementById("name").value,
document.getElementById("cardNumber").value, document.getElementById("cvc").value, total,
document.getElementById("loyalpoints").value)}>Submit
                  </button>
                </fieldset>
              </form>
            </div>
          </div>
        </div>)
    }
}
export default SampathPay;
```

TrainDetails.js

```
import React,{Component} from 'react';
import axios from 'axios';
import {Link} from 'react-router-dom';

class TrainDetails extends Component{
  constructor(props){
```

```
    super(props);
    this.state = {
      details:''
    }
}

componentWillMount() {
    this.showTrainsById();
}

showTrainsById(){
    let trainId = this.props.match.params.id;
    axios.get('http://localhost:3000/api/train/${trainId}')
      .then(response =>{
        this.setState({details: response.data}
        , () =>{
          console.log(this.state)
          })
      })
      .catch(err => console.log(err));
}

onDelete(){
    let trainId = this.state.details.id;
    axios.get("http://localhost:3000/api/train/${trainId}")
      .then(response =>{
        this.props.history.push('/');
      })
      .catch(err => console.log(err));
}

render(){
    return(
      <div>
        <Link classnName={"btn grey"} to={"/"}>Back</Link>
        <h1>{this.state.details.name}</h1>
        <ul className={"collection"}>
          <li className={"collection-item"}>Price:{this.state.details.price}</li>
        </ul>

        <Link className={"btn"} to={`/trains/edit/${this.state.details.id}`}>Edit</Link>
        <button onClick={this.onDelete.bind(this)} className="bnt red light">Delete</button>
      </div>
    )
```

```
  }
}

export default TrainDetails;
```

trainItem.js

```
import React, {Component} from 'react';
import {Link} from 'react-router-dom';
import '../App.css';
import ReactDOM from "react-dom";
import ViewDetails from './viewDetails';

class TrainItem extends Component {
  constructor(props) {
    super(props);
    this.state = {
      item: props.item
    }
  }

  booking = function (e) {
    ReactDOM.render(<ViewDetails />, document.getElementById('root'));
  };

  render() {
    return (
      <li className={"collection-item"}>
        <table>
          <th><Link to={`/trains/${this.state.item.id}`}>{this.state.item.name}</Link></th>

          <td>
            <tr>
              <Link to={`/trains/${this.state.item.id}`}>{this.state.item.class}</Link></tr>
            <tr>
              <Link to={`/trains/${this.state.item.id}`}>{this.state.item.availableSeats}</Link>
            </tr>
            <tr>
              <Link to={`/trains/${this.state.item.id}`}>{this.state.item.time}</Link>
            </tr>
            <tr>
              <Link to={`/trains/${this.state.item.id}`}>{this.state.item.startStation}</Link>
            </tr>
            <tr>
```

```
                    <Link to={`/trains/${this.state.item.id}`}>{this.state.item.endStation}</Link>
                  </tr>
                </td>

                <td>
                  <tr><button onClick={() => this.booking()}>Book ticket</button></tr>
                </td>

              </table>


            </li>


        )
    }
}

export default TrainItem;
```

viewDetails.js

```
import React, { Component } from "react";
import { Button, FormGroup, FormControl } from "react-bootstrap";
import "../App.css";
import ReactDOM from "react-dom";
import Home from "./Home";
import MakePayment from "./makePayment";
import FormLabel from "react-bootstrap/FormLabel";

class ViewDetails extends Component {
  constructor(props) {
    super(props);

    this.state = {
      email: "",
      name: "",
      noOfTickects:" ",
      NIC:" ",
      class:" ",
      time:" "
    };
  }
```

```
  back = function (e) {
    ReactDOM.render(<Home />, document.getElementById('root'));
  };

  next = function (e) {
    ReactDOM.render(<MakePayment />, document.getElementById('root'));
  };

  validateForm() {
    return this.state.email.length > 0 && this.state.name.length > 0 && this.state.name.length < 50 &&
this.state.noOfTickects !== 0 && this.state.NIC.length > 0 && this.state.NIC.length < 10;
  }

  handleChange = event => {
    this.setState({
      [event.target.id]: event.target.value
    });
  }

  handleSubmit = event => {
    event.preventDefault();
  }

  render() {
    return (
      <div className="View">
        <form onSubmit={() => this.next()}>
          <table>
            <td>
              <tr>selected train:</tr>
              <br/>
              <tr>selected class:</tr>
              <br/>
              <tr>number of tickets:</tr>
              <br/>
              <tr>selected time:</tr>
              <br/>
              <tr>name:</tr>
              <br/>
              <tr>email:</tr>
              <br/>
              <tr>NIC:</tr>
              <br/>
              <Button block bsSize="large" type="submit" onClick={() => this.back()}>Back</Button>
```

```
                </td>

                <td>
                  <tr><input value={this.state.name} onChange={this.handleChange} type="text"/></tr>
                  <br/>
                  <tr><input value={this.state.class} onChange={this.handleChange} type="text"/></tr>
                  <br/>
                  <tr><input value={this.state.noOfTickects} onChange={this.handleChange}
type="number"/></tr>
                  <br/>
                  <tr><input value={this.state.time} onChange={this.handleChange}
type="datetime"/></tr>
                  <br/>
                  <tr><input onChange={this.handleChange} type="text"/></tr>
                  <br/>
                  <tr><input onChange={this.handleChange} type="text"/></tr>
                  <br/>
                  <tr><input onChange={this.handleChange} type="text"/></tr>
                  <br/>
                  <Button block bsSize="large" enable={this.validateForm()} onClick={() => this.next()}
type="submit">Proceed Payment</Button>
                </td>
              </table>
            </form>
          </div>
      );
    }
}

export default ViewDetails;
```

backend

server.js

```
// Copyright IBM Corp. 2016. All Rights Reserved.

// Node module: loopback-workspace

// This file is licensed under the MIT License.

// License text available at https://opensource.org/licenses/MIT


'use strict';
```

```
var loopback = require('loopback');
var boot = require('loopback-boot');

var app = module.exports = loopback();

app.start = function() {
  // start the web server
  return app.listen(function() {
    app.emit('started');
    var baseUrl = app.get('url').replace(/\/$/, '');
    console.log('Web server listening at: %s', baseUrl);
    if (app.get('loopback-component-explorer')) {
      var explorerPath = app.get('loopback-component-explorer').mountPath;
      console.log('Browse your REST API at %s%s', baseUrl, explorerPath);
    }
  });
};

// Bootstrap the application, configure models, datasources and middleware.
// Sub-apps like REST API are mounted via boot scripts.
boot(app, __dirname, function(err) {
  if (err) throw err;

  // start the server if `$ node server.js`
  if (require.main === module)
    app.start();
});
```

dataSource.json

```
{
  "db": {
    "host": "localhost",
    "port": 27017,
    "url": "",
    "database": "train",
    "password": "",
    "name": "db",
    "user": "",
    "useNewUrlParser": true,
    "connector": "mongodb"
  }
}
```

authentication.js

```
// Copyright IBM Corp. 2016. All Rights Reserved.
// Node module: loopback-workspace
// This file is licensed under the MIT License.
// License text available at https://opensource.org/licenses/MIT

'use strict';

module.exports = function enableAuthentication(server) {
  // enable authentication
  server.enableAuth();
};
```

root.js

```
// Copyright IBM Corp. 2016. All Rights Reserved.
// Node module: loopback-workspace
// This file is licensed under the MIT License.
// License text available at https://opensource.org/licenses/MIT

'use strict';
```

```
module.exports = function(server) {
  // Install a `/` route that returns server status
  var router = server.loopback.Router();
  router.get('/', server.loopback.status());
  server.use(router);
};
```

Customers.js

```
'use strict';

module.exports = function(Customers) {

};
```

Customer.json

```
{
  "name": "customers",
  "base": "PersistedModel",
  "idInjection": true,
  "options": {
    "validateUpsert": true
  },
  "properties": {
    "name": {
      "type": "string",
      "required": true,
      "default": "null"
    },
    "email": {
      "type": "string",
      "required": true,
      "default": "null"
    },
    "noOfTickects": {
      "type": "number",
      "required": true
    },
    "bookingId": {
      "type": "string",
      "required": true,
      "default": "null"
    },
    "NIC": {
      "type": "string",
```

```
      "default": "null"
    },
    "paymentId": {
      "type": "string",
      "required": true,
      "default": "null"
    }
  },
  "validations": [],
  "relations": {},
  "acls": [],
  "methods": {}
}
```

Train.js

```
'use strict';


module.exports = function(Train) {


};
```

Train.json

```
{
  "name": "train",
  "base": "PersistedModel",
  "idInjection": true,
  "options": {
    "validateUpsert": true
  },
  "properties": {
    "name": {
      "type": "string",
      "required": true,
      "default": "null"
    },
    "class": {
      "type": "string",
      "required": true,
      "default": "null"
    },
    "seats": {
      "type": "number",
      "required": true,
      "default": 0
    },
    "availableSeats": {
      "type": "number",
      "required": true,
      "default": 0
    },
```

```
    "bookedSeats": {
      "type": "number",
      "required": true,
      "default": 0
    },
    "time": {
      "type": "date",
      "required": true
    },
    "price": {
      "type": "number",
      "required": true,
      "default": 0
    },
    "startStation": {
      "type": "string",
      "required": true,
      "default": "null"
    },
    "endStation": {
      "type": "string",
      "required": true,
      "default": "null"
    }
  },
  "validations": [],
  "relations": {},
  "acls": [],
  "methods": {}
}
```

References

https://codehutt.blogspot.com/2019/05/how-to-build-restful-api-with-react-and.html?m=1