



Sri Lanka Institute of Information Technology

# **Online Pharmacy System**

**Programming Assignment report-**

**System Programming and Designing Concepts**

**(IT301)**

Submitted by:

**IT16015722– P.M.E Alawaththa**

Date of submission:15-10-2018

## **Table of Contents**

Introduction.....	3
High Level Architectural Diagram.....	4
System Workflow & Scenario Execution.....	8
Authentication and Security Mechanism.....	11
Appendix.....	13
(Frontend, Backend, client, DBSchema )	

## **Introduction**

The 'Online Pharmacy System' web application is implemented using the technologies such as ReactJS for the frontend (client side),and NodeJS for the backend (server side) and Monogdb as the database. The web application sends an automatically generate emails to the client who orders it.

The user should signup to use the services provided by this application. After successful signup the user is able to login into his/her account and can search for available pharmacy items such as medicines and add it to the shopping cart and the total of the items will be displayed in a real-time. Once the user is done with shopping the items, he or she can purchase the selected items using sampath payment gateway and also with the dialog number and when the purchase is completed the user will receive loyalty points which is calculated based on the total amount and the user can cash out those points at any time. The user will also get an email about the purchase that was made.

The selection of medicine is implemented in the frontend and as the backend functionality, the user management or handling is been done. I have used php storm to implement this system.

## High Level Architecture Diagrams

### Sequence Diagram

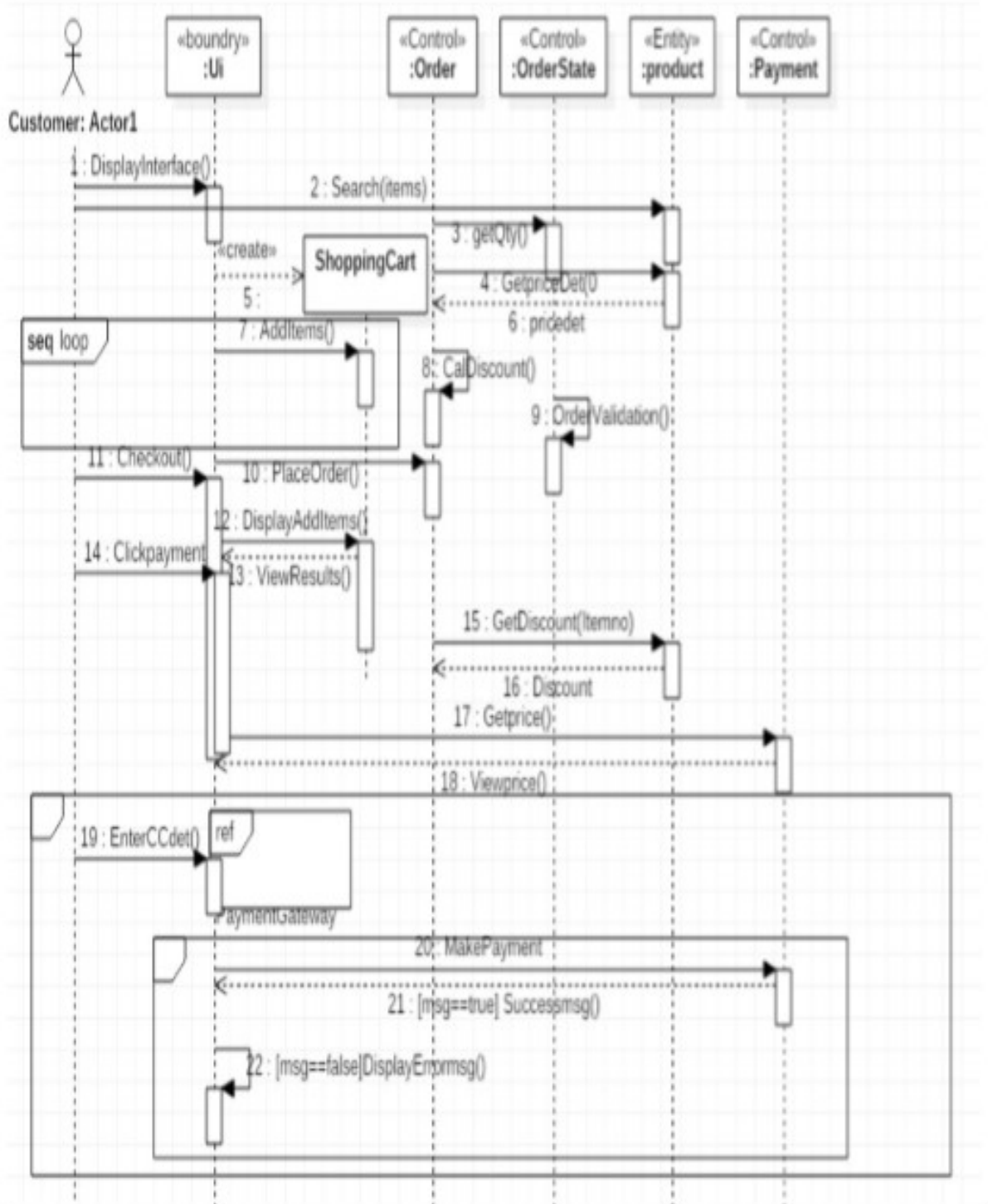


Fig1

## Reference of payment Gateway

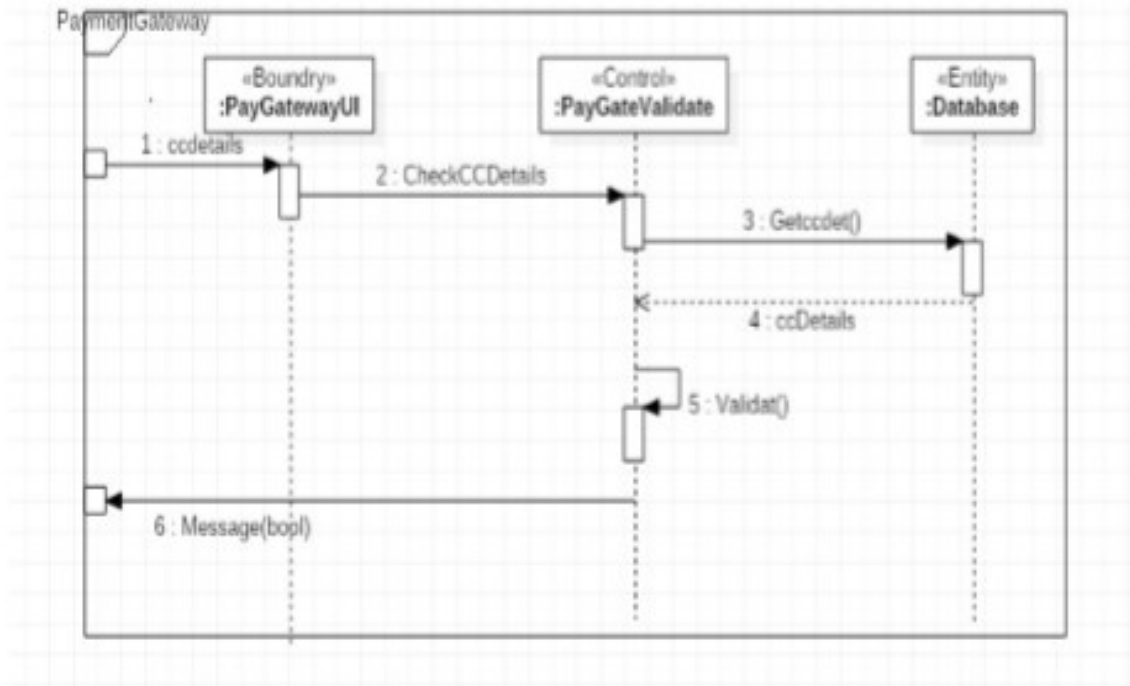


Fig 2

Here in the sequence diagram in Fig 1 shows how the user has the ability to Display the UI screen and order any product that user wants. Then the order can be settled through the Payment gateways available, we have used in this system the Sampath gateway system and dialog payment. The credit card details validation will reply a message to the user to confirming the order.

The Fig 2 shows the reference diagram how the payment is processed using the payment gateway methods available.

In Fig3 I have shown how a normal user has to face the system as new user the signup process by filling the signup page them redirect into the functionalities.

## Activity Diagram

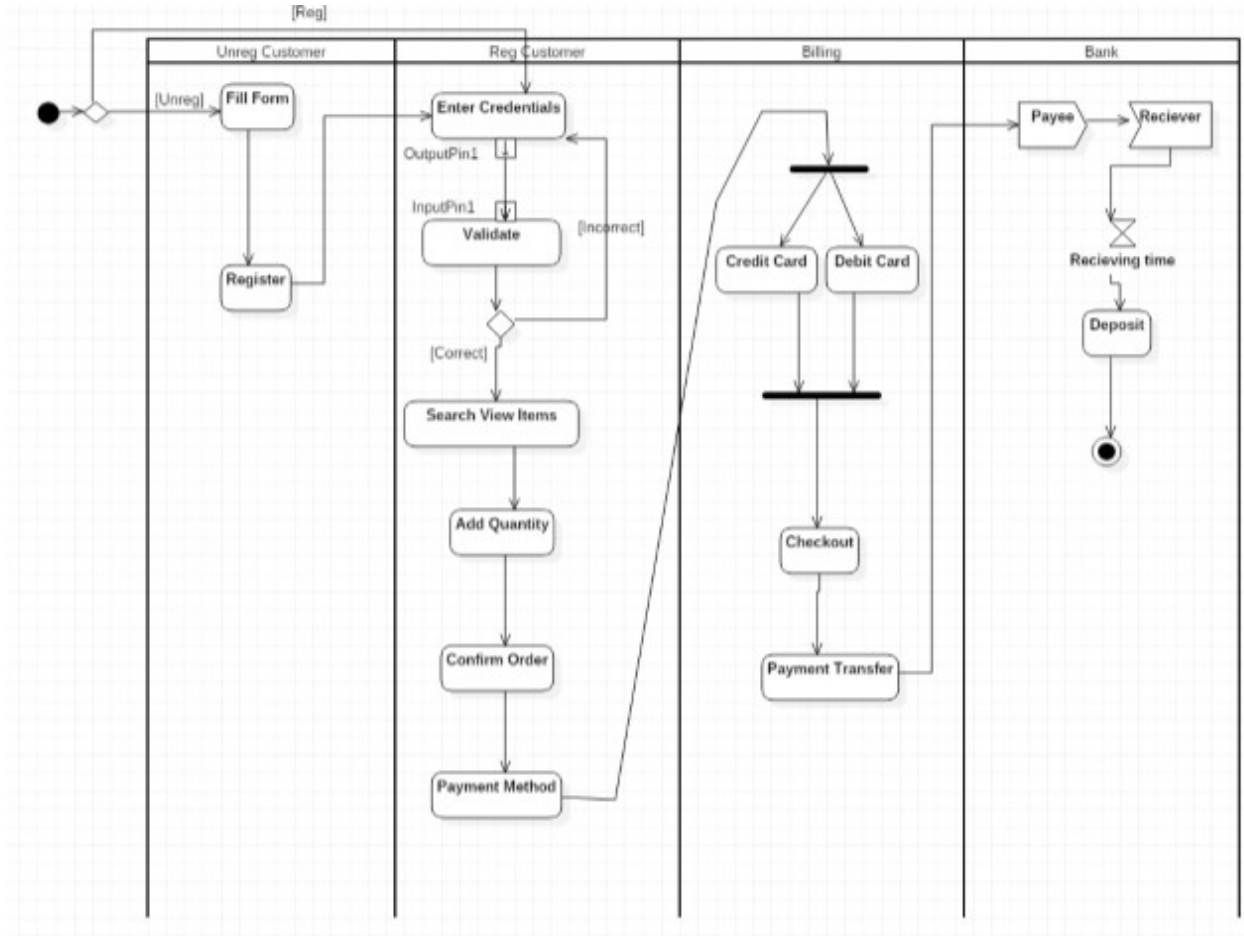


Fig 3

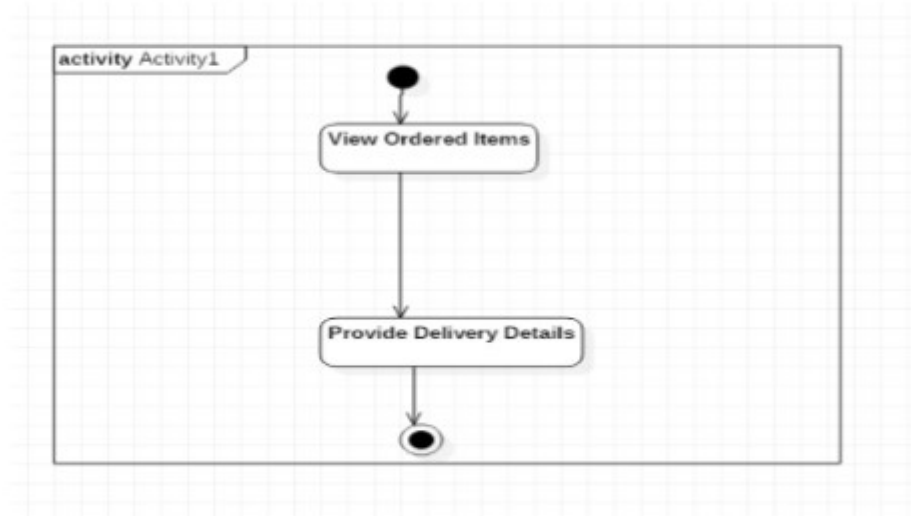


Fig 4 Call activity for confirm order

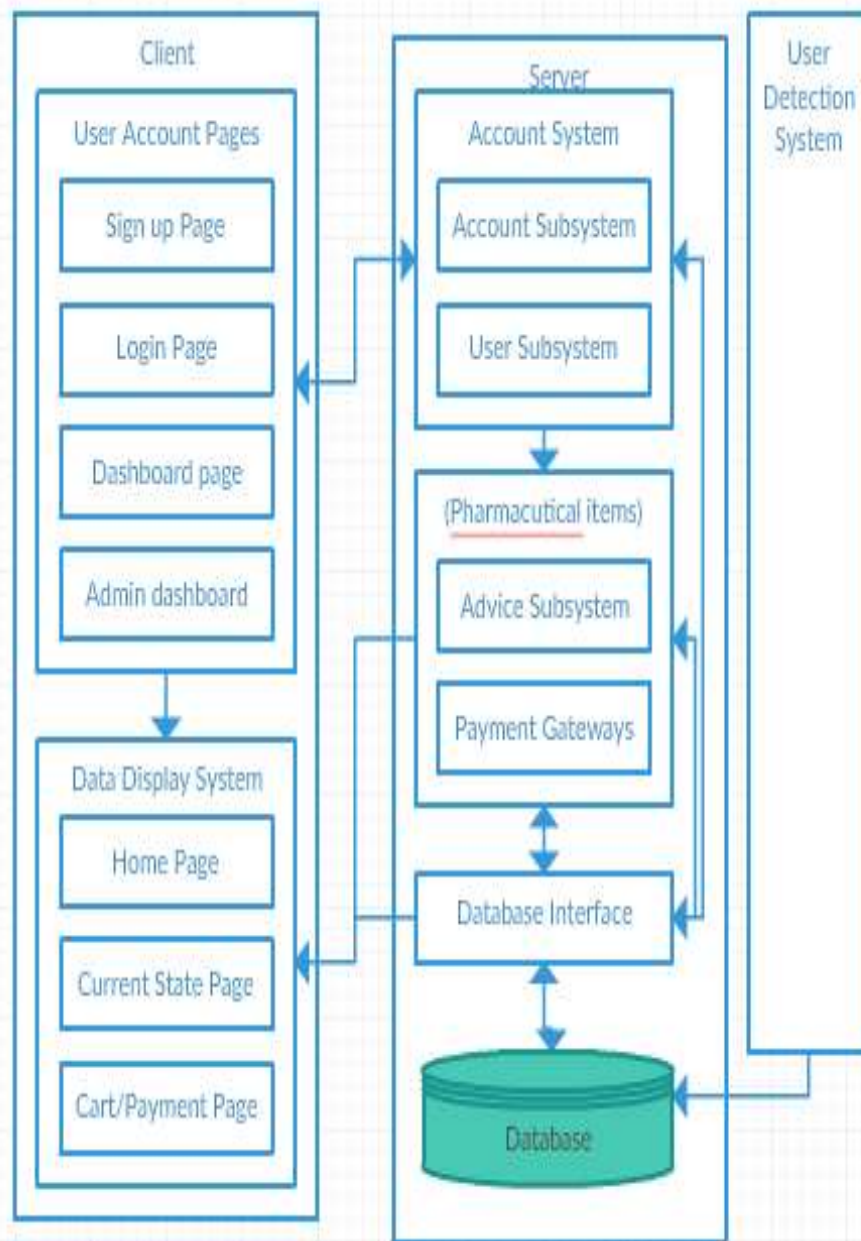


Fig 5

Figure 5 shows, High Level Architecture diagram & how the system work flows with databases and the available components.

## **System Workflow & Scenario Execution**

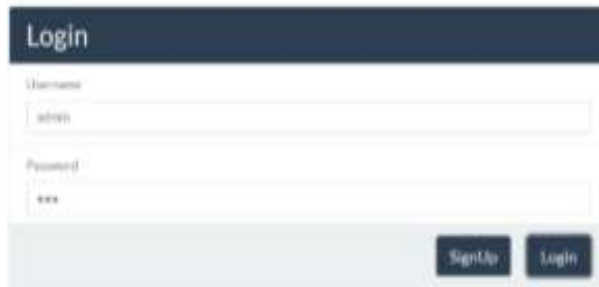
A login form with a dark blue header containing the word "Login". Below the header, there are two input fields: "Username" with the text "admin" and "Password" with masked characters "xxx". At the bottom right, there are two buttons: "SignUp" and "Login".

Fig 6

The Login screen that is displayed for the user into log the system. The Screen that every user has to be displayed.

A quantity selection modal. It has a title bar with a close button. The main content area has a label "Select Quantity :", a text input field containing "1", and a message "To add more than 5, click again". At the bottom, there are two buttons: "Close" and "Add to Cart". The background is dimmed, showing a list of items with prices like "Rs.45.00", "Rs.30.00", and "Rs.40.00".

Fig7

Quantity selection screen from the medicines available any user can select the number of any quantities. And proceed to the add cart function.



The screenshot shows a web application interface for payments. At the top, there is a dark blue header with a logo on the left and a 'Logout' button on the right. Below the header, a light gray bar contains the text 'Home > Payments'. The main content area is titled 'Payments' and displays a summary of the transaction. It shows 'Total = Rs.1240.00', 'Unsettled Check-out = Rs.0.00', and a bold 'Sub Total = Rs.1240.00'. Below this summary, there is a section titled 'Payments via' with two buttons: 'Mobile' and 'Credit Card'.

Fig 8

The screen(fig 8) that used to be displayed when the payment process is in progress by chosen medicine via pharmacy items. It displays the user to pay using mobile as well as the credit card payment and can logout if the user needs.

The screenshot shows a modal form titled 'Payments via Sampath Credit Card' overlaid on a blurred background of the 'Payments' screen. The form includes the Sampath Bank logo and the text 'Sampath Bank' in Sinhala. It contains four input fields: 'Credit Card No.', 'CVC No.', 'Holder's Name', and 'Amount'. The 'Amount' field is pre-filled with 'Rs. 1240.00'. At the bottom right of the form, there are two buttons: 'Close' and 'Do Payment'.

Fig 9

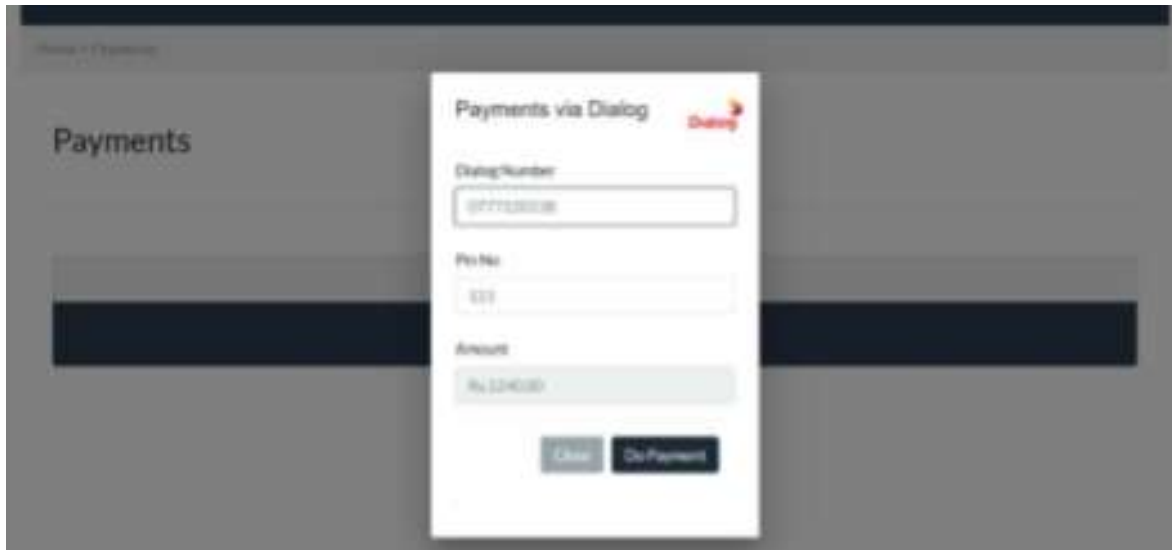


Fig 10

Here I have used the payment gateway options via sampath gateway, and dialog pay which generates a sms to the phone number. In figure 9 & 10 it shows how the system uses and make system to operate using payment methods.

## Authentication and Security Mechanism

Below I have stated the security mechanisms available for the system when logged in.

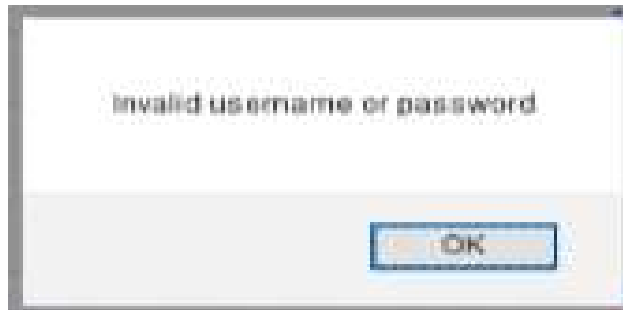


Fig 10

The displaying of the popup to enhance the user enters correct password or username entry.

```
class SignUp extends Component {  
  
  signup() {  
    const pss1 =  
document.getElementById('password1').value;  
    const pss2 =  
document.getElementById('password2').value;  
  
    if( pss1.length>0 && pss1 === pss2){  
      const body = {  
        name:document.getElementById('name').value,  
  
mobileNo:document.getElementById('mobile').value,  
  
email:document.getElementById('email').value,  
  
username:document.getElementById('username').value,  
  
password:document.getElementById('password1').value,  
        loyaltyPoints:0,  
        type:document.getElementById('type').value  
      };  
      console.log(pss1.length,pss2,body);  
  
      axios.post(API.Get_All_Customers,body).then(function  
(response) {  
        if(response.status==200) {
```

```

        window.alert('Welcome to Online
pharmacy')
        localStorage.setItem('logged', 'ok');
        localStorage.setItem('user',
response.data.name);
        localStorage.setItem('type',
response.data.type);
        localStorage.setItem('id',
response.data.id || response.data._id);
        localStorage.setItem('points',
response.data.loyaltyPoints);
        console.log()
        response.data.type === 'admin' ?
window.location.href = '/Admin' : window.location.href =
'/Dashboard';

    }
    else window.alert('cannot create :(')
  }).catch(function (err) {
    window.alert('cannot create :(')
  });
}
else window.alert('Invalid username or password ');
}
}

```

Users are identified uniquely by their email addresses and once a user logs in his/her username will be displayed in the signup page. The above code implies how the system uses them. For the specific user these type of security mechanisms are been done.

## Appendix

App.js

```
const express = require('express');
const app = express();

const bodyParser = require('body-parser');
app.use(bodyParser.json());

const PORT = process.env.port || 3002 || 3001 || 3000;

// Server startup
app.listen(PORT, function (err) {
  if(err) {
    console.log(err);
    process.exit(-1);
  }
  console.log('Pharmacy company server is running on port ' + PORT);
});

const pharmacy = [{name:"Panadol", price:45},
  {name:"Piriton",price:30},
  {name:"Amoxilin",price:40},
  {name:"Vitamin",price:40},
  {name:"Peroxide",price:340}];

const customers = [{
  id:1,
  name:"Malith Alawaththa",
  type:"admin",
  mobileNo:"0718220691",
  username:"admin",
  password:"123",
  email:"malitha.emedia@gmail.com",
  loyaltyPoints:0},
  {
    id:2,
    name:"Lionel Messi",
    type:"user",mobileNo:"0777234543",
    username:"user",
    password:"123",
    email:"messi@gmail.com",
```

```

        loyaltyPoints:5
    }]);

const payments = [];

const dialog = [{phoneNo:"0771534567",pin:1234}];
const sampath =
[{{creditCardNo:1234,cvc:123,holdersName:'ds'}}];

// To resolve cross origin issues
app.use(function (req, res, next) {
    res.header('Access-Control-Allow-Origin', "*");
    res.header('Access-Control-Allow-Methods',
'GET,PUT,POST,DELETE');
    res.header('Access-Control-Allow-Headers', 'Content-
Type');
    next();
});

// REST APIs

// Customers service
app.get('/customers', function(req, res) {
    res.status(200).send(customers);

});

// Customer get by id
app.get('/customers/:id', function (req,res) {
    const customer = customers.find(c => c.id ===
parseInt(req.params.id));

    if(!customer) res.status(404).send('cannot find the
customer');

    else res.status(200).send(customer);
});

// POST a customer
app.post('/customers', function (req, res) {

    if(!req.body) {
        res.status(404).send('cannot find the customer');
        return;
    }

```

```

    }
    const customer = {
      id: customers.length + 1,
      name: req.body.name,
      type: req.body.type,
      mobileNo: req.body.mobileNo,
      username: req.body.username,
      password: req.body.password,
      email: req.body.email,
      loyaltyPoints: "0"
    };

    customers.push(customer);
    res.status(200).send(customer);
    fs.writeFile('./Files/Customers.txt', JSON(customer),
function (err) {
    if (err) throw err;
    console.log('Saved!');
  });
});

// Check Login

app.post('/customers/login', function (req, res) {
  if(!req.body) {
    res.status(404).send('cannot find the customer');
    return;
  }

  const customer = customers.find(c => (c.username ===
req.body.username) && (c.password === req.body.password));

  if(!customer) res.status(404).send('cannot find the
customer');

  else res.status(200).send(customer);

});

```

```
// Pharmacy Service
```

```
app.get('/pharmacy', function(req, res) {
  res.status(200).send(pharmacy);
});

app.get('/pharmacy/:id', function (req,res) {
  const pharmacy = pharmacy.find(c => c.id ===
parseInt(req.params.id));

  if(!medicine) res.status(404).send('cannot find
medic');

  else res.status(200).send(medicine);
});

app.post('/pharmacy', function (req, res) {
  if(!req.body) {
    res.status(400).send('Bad request');
    return;
  }
  const medic = {
    name: req.body.name,
    price: req.body.price
  };

  pharmacy.push(medic);
  res.status(200).send(medic);
});
```

```
// Dialog Payment Gateway
```

```
app.post('/dialog', function (req, res) {
  if(!req.body) {
    console.log('invalid');
    return;
  }

  const holder = dialog.find(c => (c.phoneNo ===
req.body.phoneNo) && (c.pin === parseInt(req.body.pin)));

  if(!holder) res.status(404).send('Not Found');

  else {
    const body = {
      amount:req.body.amount,
```



```

        userId:req.body.userId,
        date:new Date()
    };
    payments.push(body);
    res.status(200).send('payments are success');

}
console.log(holder);
});

//Sampath Payment Gateway
app.post('/sampath', function (req, res) {
    if(!req.body) {
        console.log('invalid');
        return;
    }

    const holder = sampath.find(c => (c.creditCardNo ===
    parseInt(req.body.creditCardNo)) &&
    (c.cvc ===
    parseInt(req.body.cvc)) &&
    (c.holdersName ===
    (req.body.holdersName));

    if(!holder) res.status(404).send('Not Found');

    else {
        const body = {
            amount:req.body.amount,
            userId:req.body.userId,
            date:new Date()
        };
        payments.push(body);
        res.status(200).send('payments are success');

    }
    console.log(holder);
});

```

dbApi.js

```
const express = require('express');
const app = express();

const bodyParser = require('body-parser');
app.use(bodyParser.json());

const PORT = process.env.port || 3002 || 3001 || 3000;

const mongoose1 = require('./DBSchema/CustomerSchema');
const mongoose2 = require('./DBSchema/MedicSchema');
const mongoose3 =
require('./DBSchema/PaymentGateways/Dialog');
const mongoose4 =
require('./DBSchema/PaymentGateways/Sampath');
const mongoose5 = require('./DBSchema/PaymentSchema');

const CustomerSchema = mongoose1.model('Customer');
const MedicSchema = mongoose2.model('Medicine');
const dialogSchema = mongoose3.model('dialog');
const sampathSchema = mongoose4.model('sampath');
const PaymentSchema = mongoose5.model('Payment');

const Mongo_DB_Base_URL = "mongodb://127.0.0.1:27017";

// Server
app.listen(PORT, function (err) {
  if(err) {
    console.log(err);
    process.exit(-1);
  }
  console.log('Pharmacy server is running on port ' +
PORT);
});

// To resolve cross origin issues
app.use(function (req, res, next) {
  res.header('Access-Control-Allow-Origin', "*");
  res.header('Access-Control-Allow-Methods',
'GET,PUT,POST,DELETE');
  res.header('Access-Control-Allow-Headers', 'Content-
Type');
  next();
});
```

```

// Db connection
mongoose1.connect(Mongo_DB_Base_URL+"/Online Pharmacy",
function (err) {
    if(err) {
        console.log(err);
        process.exit(-1);
    }
    console.log('Pharmacy Db is connected');
});

mongoose2.connect(Mongo_DB_Base_URL+"/Medicine", function
(err) {
    if(err) {
        console.log(err);
        process.exit(-1);
    }
    console.log('Pharmacy Db is connected');
});

mongoose3.connect(Mongo_DB_Base_URL+"/dialog", function
(err) {
    if(err) {
        console.log(err);
        process.exit(-1);
    }
    console.log('connected to Dialog Payment gateway ...');
});

mongoose4.connect(Mongo_DB_Base_URL+"/sampath", function
(err) {
    if(err) {
        console.log(err);
        process.exit(-1);
    }
    console.log('Connected to Sampath Bank Payment
gateway...');
});

mongoose5.connect(Mongo_DB_Base_URL+"/Payments", function
(err) {
    if(err) {
        console.log(err);
        process.exit(-1);
    }
    console.log('Payments DB is connected');
});

```

```

});

// REST API Customers

app.get('/customers',function(req, res) {
    CustomerSchema.find().exec().then(function (customers)
    {
        res.status(200).send(customers);
    }).catch(function (err) {
        res.status(400).send('Bad Reques');
    });
});

app.get('/customers/:id',function(req, res) {

    CustomerSchema.find({_id:req.params.id}).exec().then(function (customers) {
        res.status(200).send(customers);
    }).catch(function (err) {
        res.status(400).send('Bad Reques');
    });
});

app.post('/customers', function (req, res) {

    const data = new CustomerSchema(req.body);
    console.log(req.body);
    data.save().then(function (data) {
        res.status(200).send(data);
    }).catch(function () {
        res.status(400).send({message:'Bad request'});
    });
});

app.post('/customers/login', function (req, res) {
    if(!req.body) {
        res.status(404).send('cannot find the customer');
        return;
    }

    CustomerSchema.find({username:req.body.username,password:req.body.password}).exec().then(function (customers) {
        res.status(200).send(customers[0]);
    }).catch(function (err) {
        res.status(404).send('cannot find the customer');
    });
});

```

```

    });

});

app.put('/customers/:id', function (req, res) {
    CustomerSchema.update({_id:req.params.id},
    req.body).then(function (customer) {
        res.status(200).send({customer: customer, message:
        'updated'});
    }).catch(function () {
        res.status(400).send({message:'Bad request'});
    });
});

app.delete('/customers/:id', function (req, res) {
    CustomerSchema.remove({_id:
    req.params.id}).then(function () {
        res.status(200).send({message:'deleted'});
    }).catch(function () {
        res.status(400).send({message:'Bad request'});
    });
});

// REST API medicine

app.get('/medicine',function(req, res) {
    FoodSchema.find().exec().then(function (pharmacy) {
        res.status(200).send(medicine);
    }).catch(function (err) {
        res.status(400).send('Bad Reques');
    });
});

app.get('/medicine/:name',function(req, res) {

MedicSchema.find({name:req.body.name}).exec().then(function
(food) {
    res.status(200).send(pharmacy);
    }).catch(function (err) {
        res.status(400).send('Bad Reques');
    });
});

app.post('/medicine',function(req, res) {
    const pharmacy = new MedicSchema(req.body);
    pharmacy.save().then(function (pharmacy) {

```

```

        res.status(200).send(medicine);
    }).catch(function (err) {
        res.status(400).send('Bad Reques');
    });
});

app.put('/medicine/:id', function (req, res) {
    CustomerSchema.update({_id:req.params.id},
    req.body).then(function (customer) {
        res.status(200).send({customer: customer, message:
        'updated'});
    }).catch(function () {
        res.status(400).send({message:'Bad request'});
    });
});

app.put('/medicine/:id', function (req, res) {
    MedicSchema.update({_id:req.params.id},
    req.body).then(function (food) {
        res.status(200).send({customer: medicine, message:
        'updated'});
    }).catch(function () {
        res.status(400).send({message:'Bad request'});
    });
});

app.delete('/medicine/:id', function (req, res) {
    MedicSchema.remove({_id: req.params.id}).then(function
    () {
        res.status(200).send({message:'deleted'});
    }).catch(function () {
        res.status(400).send({message:'Bad request'});
    });
});

// Dialog Payment Gateway APIs

app.post('/dialog', function (req, res) {
    if(!req.body) {
        console.log('invalid');
        return;
    }
    const dialog = new dialogSchema({
        amount:req.body.amount,
        phoneNo:req.body.phoneNo,
        date:new Date()
    });

```

```

});

const payment = new PaymentSchema({
  amount:req.body.amount,
  type:req.body.type,
  date:new Date(),
  userid:req.body.userId
});

if(req.body.pin === '1234') {
  dialog.save().then(function () {
    payment.save().then(function () {
      res.status(200).send({message:'payment
success via dialog'});
    }).catch(function () {
      res.status(400).send({message:'Bad
request'});
    });
    // res.status(200).send();
  }).catch(function () {
    res.status(400).send({message:'Bad request'});
  });
}

else res.status(400).send({message:'Bad request'});

});

// Sampath Payment Gateway APIs

app.post('/sampath', function (req, res) {
  if(!req.body) {
    console.log('invalid');
    return;
  }
  const sampath = new sampathSchema({
    amount:req.body.amount,
    holdersName:req.body.holdersName,
    creditCardNo:req.body.creditCardNo,
    date:new Date(),
  });

  const payment = new PaymentSchema({
    amount:req.body.amount,
    type:req.body.type,
    date:new Date(),

```

```

        userid:req.body.userId
    });

    if(req.body.cvc === '123') {
        sampath.save().then(function () {
            payment.save().then(function () {

res.status(200).send({data:payment,message:'payment success
via sampath'}));
                }).catch(function () {
                    res.status(400).send({message:'Bad
request'}));
                });
                //res.status(200).send();
            }).catch(function () {
                res.status(400).send({message:'Bad request'});
            });
        }

        else res.status(400).send({message:'Bad request'});

    });

    // Payments APIs

    app.get('/payments',function (req,res) {
        PaymentSchema.find().exec().then(function (data) {
            res.status(200).send(data);
        }).catch(function (err) {
            console.log(err);
        })
    });
});

```



## DBSchema

### PayemntSchema.js

```
const mongoose = require('mongoose');
const Schema = mongoose.Schema;

const PaymentSchema = new Schema({
  amount: {
    type: String,
    require: true
  },
  type: {
    type: String,
    require: true
  },
  date: {
    type: String,
    require: true
  },
  userid: {
    type: String,
    require: true
  }
});

mongoose.model('Payment', PaymentSchema);
module.exports = mongoose;
```

## Web client

### App.jsx

```
import React, { Component } from 'react';
import {HashRouter, Route} from 'react-router-dom';

import Login from './app/containers/Login';
import Dashboard from './app/containers/Dashboard';
import Payments from './app/containers/PaymentGateway';
import SignUp from './app/containers/SignUp';
import AdminDashboard from
'./app/containers/AdminDashboard';

class App extends Component {
```

```

render() {
  const isLoggedIn = localStorage.getItem('logged');
  const type = localStorage.getItem('type');
  return (
    <HashRouter>
      {
        isLoggedIn === 'ok'
        ?
          type === 'user'
          ?
            <div>
              <Route exact path="/"
component={Dashboard} />
              <Route
path="/Dashboard" component={Dashboard} />
              <Route path="/Payments"
component={Payments} />
            </div>
          :
            <div>
              <Route exact path="/"
component={AdminDashboard} />
              <Route
path="/Dashboard" component={Dashboard} />
              <Route path="/Payments"
`component={Payments} />
              <Route path="/Admin"
component={AdminDashboard} />
            </div>
          :
            <div>
              <Route exact path="/"
component={Login} />
              <Route path="/Login"
component={Login} />
              <Route path="/SignUp"
component={SignUp} />
            </div>
          }
      </HashRouter>
    );
  }
}
export default App;

```

## Components

### Header.js

```
import React, {Component} from 'react';

import logo from '../images/logo.png'

import '../scss/Header.css';

class Header extends Component {

  signOut = () => {
    window.localStorage.clear();
    window.location.href = '/login';
  }

  render() {
    return(
      <div>
        <span><img src={logo} /></span>
        <span>Online Pharmacy</span>

        <span className='ds-logout'>

<span>{localStorage.getItem('user')}</span>
          <button type="button" class="btn btn-
primary" onClick={this.signOut}>logout</button>
        </span>

          <span className="navbar navbar-expand-lg
navbar-dark bg-primary" />
        </div>
      );
    }
}

export default Header;
```

### ButtonBar.js

```
import React, {Component} from 'react';
import {Link} from 'react-router-dom';
import '../scss/ButtonBar.css';

import Payments from '../containers/PaymentGateway';

class ButtonBar extends Component {

  render() {
    let redirect='/Payments';
    this.props.work === 'Payments' ?
    redirect='/Payments' : redirect='/Dashboard';
    return(
      <div className="row">
        <div className="col-md-7 ds-buttonbar">

          </div>
          <div className="col-md-3">
            <Link to={redirect}>
              <button type="button" className="btn
btn-primary btn-lg btn-block">{this.props.work}</button>
            </Link>
          </div>
          <div className="col-md-2 ds-buttonbar">
            </div>
        </div>
      );
    }
}
export default ButtonBar;
```

### Breadcrum.js

```
import React, {Component} from 'react';

class Breadcrumb extends Component {
  render() {
    return(
      <ol className="breadcrumb">
        <li className="breadcrumb-item
active">{this.props.path}</li>
      </ol>
    );
  }
}
```

```
}
```

```
export default Breadcrumb;
```

### AllPharmacyItems.jsx

```
import React, {Component} from 'react';
```

```
import '../.../scss/PharmacyItems.css'
```

```
import PharmacyRaw from './PharmacyRaw';
```

```
import * as API from '../.../Constants/APIs';
```

```
import axios from 'axios';
```

```
class AllPharmacyItems extends Component {
```

```
  constructor(props) {
    super(props);
    this.state = {
      allPharmacy:[],
      isResponseFollFill:false
    }
  }
}
```

```
  componentDidMount() {
    let url = API.Get_All_Pharamacy;
    let pharmacy=this;

    axios.get(url).then(function (response) {
      pharmacy.setState({
        allPharmacy:response.data,
        isResponseFollFill:true
      });
      console.log(response.data);
    }).catch(function (err) {
      console.log(err);
    });
  }
}
```

```
  render() {

    return (
      <div>
        <h4>Pharmacy Items  <i className="fas fa-
```

```

utensils"/></h4>
    {this.props.disable === false ?
      <p> <i className="far fa-star"/> Click
on a Pharmacy Item</p>
      :
      null}
    <div className="ds-content-wrapper">
      <table className="table table-hover ">
        <thead >
          <tr>
            <th scope="col">Medicine</th>
            <th scope="col">Price</th>
          </tr>
        </thead>

        {
          this.state.allPharmacy.map((pharmacy, index) => {
            return (
              <PharmacyRaw key={index}
index={index} pharmacy={pharmacy}

addPharmacyToSelectedPharmacy={this.props.addPharmacyToSele
ctedPharmacy}

disable={this.props.disable}/>
            )
          })
        }
      </table>
    </div>

  </div>
);
}
}

export default AllPharmacyItems;

```

## TotalRows.jsx

```
import React, {Component} from 'react';

class TotalRow extends Component {

  constructor(props) {
    super(props);

    this.state = {

      selectedPharmacy:[{name:'Panadol',quantity:2,price:90.00}],
      total:0.00
    }
  }

  calculateTotal() {
    const selectedPharmacy =
this.props.selectedPharmacy;
    let total = 0;

    for(let i=0; i<selectedPharmacy.length; i++) {
      total = total + selectedPharmacy[i].price;
    }
    localStorage.setItem('total',total);
    return total;
  }

  render() {

    return(
      <div>
        <table className="table table-hover">
          <thead>
            <tr>
              <th scope="col"> </th>
              <th scope="col"> </th>
              <th scope="col"> </th>
              <th scope="col">Total =
Rs.{this.calculateTotal()}.00</th>
            </tr>
          </thead>
        </table>
      </div>
    );
  }
}
```

```
}
```

```
export default TotalRow;
```

SelectItem.js

```
import React, {Component} from 'react';
```

```
import '../.../scss/PharmacyItems.css'
```

```
class SelectedPharmacy extends Component {
```

```
  render() {
    return (
      <div>
        <h4>Food Cart <i className="fas fa-cart-
plus" /></h4>
        <p> <i className="far fa-star"/> Select
from Pharmacy Items</p>
        <div className="ds-content-wrapper">
          <table className="table table-hover">
            <thead>
              <tr>
                <th scope="col">Medicine</th>
                <th scope="col">Quantity</th>
                <th scope="col">Price</th>
              </tr>
            </thead>
            {
this.props.selectedPharmacy.map((pharmacy, index) => {
              return (
                <tbody key={index}>
                  <tr className="table-
active">
                    <th
scope="row">{pharmacy.name}</th>
                    <td>{pharmacy.quantity}</td>
                    <td>Rs.{pharmacy.price}.00</td>
                    <td><i
className="fas fa-trash-alt" style={{cursor:'grab'}}
onClick={ ()=>this.props.deletePharmacy(index) } /></td>

```



```

                </tr>
                <tr className="row"
style={{border:'0px solid white',padding:'5px'}}>

                </tr>

            </tbody>
        )
    })
    }
    </table>
  </div>

  </div>
);
}
}

```

```
export default SelectedPharmacy;
```

#### PharmacyRow.js

```

import React, {Component} from 'react';

import Dialog from '@material-ui/core/Dialog';
import DialogContent from '@material-
ui/core/DialogContent';
import DialogContentText from '@material-
ui/core/DialogContentText';
import DialogTitle from '@material-ui/core/DialogTitle';
import Slide from '@material-ui/core/Slide';

import '../scss/PharmacyItems.css'

function Transition(props) {
  return <Slide direction="up" {...props} />;
}

class PharmacyRaw extends Component {

  constructor(props) {
    super(props);
    this.state = {
      allFood: [{name:'Panadol',price:45.00},
{name:'Piriton',price:30.00}],

```

```

        open: false
    }
}

handleClickOpen = () => {
    this.setState({ open: true });
};

handleClose = () => {
    this.setState({ open: false });
};

addPharmacyToSelectedPharmacy = () => {
    const food = this.props.food;
    const id = "exampleSelect" + this.props.index;
    const quantity =
parseInt(document.getElementById(id).value);

    this.props.addPharmacyToSelectedPharmacy({ name: pharmacy.name,
quantity: quantity, price: (medicine.price * quantity) });
    this.handleClose();
}

render() {
    return (
        <tbody>
            <tr className="ds-medic-column table-active "
onClick={this.handleClickOpen}>
                <th
scope="row">{this.props.pharmacy.name}</th>
                <td>Rs.{this.props.pharmacy.price}.00</td>
            </tr>
            <tr className="row" style={{border: '0px solid
white',padding: '5px'}}>

            </tr>

            {
                this.props.disable === false ?
                <Dialog
                    open={this.state.open}
                    TransitionComponent={Transition}
                    keepMounted
                    onClose={this.handleClose}
                    aria-labelledby="alert-dialog-

```

```

slide-title"
aria-describedby="alert-dialog-
slide-description"
>
<DialogTitle id="alert-dialog-
slide-title">
    <div className="row">
        <div className="col">
            {this.props.pharmacy.name}
        </div>
        <div className="col ds-
dialog">
            <i className="fas fa-
utensils"/>
        </div>
    </div>
</DialogTitle>
<DialogContent>
    <div className="form-group">
        <label>Select Quantity :
</label>
        <select className="form-
control" id={"exampleSelect"+this.props.index}>
            <option
value="1">1</option>
            <option
value="2">2</option>
            <option
value="3">3</option>
            <option
value="4">4</option>
            <option
value="5">5</option>
        </select>
        To add more than 5, click
again on {this.props.pharmacy.name}
    </div>
    <div className="modal-footer">
        <button type="button"
className="btn btn-secondary" data-dismiss="modal"
onClick={this.handleClick}>Close</button>
        <button type="button"
className="btn btn-primary"
onClick={this.addPharmacyToSelectedPharmacy}>Add to

```

```

Cart</button>
                                </div>
                                <DialogContentText id="alert-
dialog-slide-description">
                                    ==Pharmacy==
</DialogContentText>
                                </DialogContent>
                                </Dialog>
                                :
                                null
                                }

                                </tbody>

                                );
                                }
                                }

export default PharmacyRaw;

```

## Containers

### Signup.js

```

import React, {Component} from 'react';

import axios from 'axios';
import * as API from '../Constants/APIs';

import '../scss/Login.css';

class SignUp extends Component {

    signup() {
        const pss1 =
document.getElementById('password1').value;
        const pss2 =
document.getElementById('password2').value;

        if( pss1.length>0 && pss1 === pss2){
            const body = {
                name:document.getElementById('name').value,

mobileNo:document.getElementById('mobile').value,

```

```

email:document.getElementById('email').value,

username:document.getElementById('username').value,

password:document.getElementById('password1').value,
        loyaltyPoints:0,
        type:document.getElementById('type').value
    };
    console.log(pss1.length,pss2,body);

axios.post(API.Get_All_Customers,body).then(function
(response) {
    if(response.status==200) {
        window.alert('Welcome to Online
pharmacy')

        localStorage.setItem('logged','ok');
        localStorage.setItem('user',
response.data.name);
        localStorage.setItem('type',
response.data.type);
        localStorage.setItem('id',
response.data.id || response.data._id);
        localStorage.setItem('points',
response.data.loyaltyPoints);
        console.log()
        response.data.type === 'admin' ?
window.location.href = '/Admin' : window.location.href =
'/Dashboard';

    }
    else window.alert('cannot create :(')
}).catch(function (err) {
    window.alert('cannot create :(')
});
}
else window.alert('Username or Password is
invalid);

}

render() {
    return(

<div className="list-group ds-signUp">

```

```
        <h3 className="list-group-item list-group-item-  
action active">SignUp</h3>
```

```
        <div className="form-group list-group-item  
list-group-item-action">  
            <label className="col-form-  
label">Name</label>  
            <input type="text" className="form-control"  
id="name" />
```

```
            <label className="col-form-label">Mobile  
No.</label>  
            <input type="text" className="form-control"  
id="mobile" />  
        </div>
```

```
        <div className="form-group list-group-item  
list-group-item-action">  
            <label >Email address</label>  
            <input type="email" className="form-  
control" id="email" aria-describedby="emailHelp"  
placeholder="Enter email" />  
            <small id="emailHelp" className="form-text  
text-muted">We'll never share your email with anyone  
else.</small>  
        </div>
```

```
        <div className="form-group list-group-item  
list-group-item-action">  
            <label className="col-form-  
label">Username</label>  
            <input type="text" className="form-control"  
id="username" />  
        </div>
```

```
        <div className="form-group list-group-item  
list-group-item-action">  
            <label>Password</label>  
            <input type="password" className="form-control"  
id="password1"/>  
        </div>
```

```
        <div className="form-group list-group-item  
list-group-item-action">  
            <label>Re-Enter Password</label>  
            <input type="password" className="form-
```

```

control" id="password2" />
    </div>

    <div className="form-group list-group-item
list-group-item-action">
        <label>User Type</label>
        <select className="form-control" id="type"
>
            <option value="admin">Admin</option>
            <option value="user">User</option>
        </select>
        <small id="emailHelp"
style={{color:'red'}}>This is provide to create an admin to
log as admin</small>

    </div>

    <div className="list-group-item list-group-
item-action disabled ds-button">
        <button type="button" className="btn btn-
primary btn-lg ds-button"
onClick={this.signup}>SignUp</button>
    </div>
</div>
);
}
}

export default SignUp;

```

### PaymentGateway.js

```

import React,{Component} from 'react';

import Header from '../components/Header';
import AmountDisplay from
'../components/Payments/AmountDisplay';
import BreadCrumb from '../components/BreadCrumb';
import PaymentSelectionBar from
'../components/Payments/PaymentSelctionBar';

import '../..//scss/Payments.css';

class PaymentGateway extends Component {

```

```

    render() {
      console.log(this.props)
      return (
        <div className="ds-container">
          <Header/>
          <Breadcrumb path={"Home > Payments"}/>
          <div className="ds-payments">
            <h3>Payments</h3>
            <AmountDisplay/>
          </div>
          <PaymentSelectionBar/>
        </div>
      );
    }
  }

export default PaymentGateway;

```

### Login.js

```

import React, {Component} from 'react';
import {Link} from 'react-router-dom';
import * as API from '../Constants/APIs';

import axios from 'axios';

import '../scss/Login.css';

class Login extends Component {

  login() {
    let login = this;

    const username =
document.getElementById('username').value;
    const password =
document.getElementById('password').value;
    const body = {username:username,password:password};

    axios.post(API.LOGIN,body).then(function (response)
    {
      if(response.status === 200) {
        console.log(response);
        localStorage.setItem('logged', 'ok');
        localStorage.setItem('user',
response.data.name);

```



```

        localStorage.setItem('type',
response.data.type);
        localStorage.setItem('id',
response.data.id);
        localStorage.setItem('points',
response.data.loyaltyPoints);
        response.data.type === 'admin' ?
window.location.href = '/Admin' : window.location.href =
'/Dashboard';
    }
    }).catch( function (err) {
        console.log(err);
    });
}

render() {
    return(
        <div className="list-group ds-login">
            <h3 className="list-group-item list-group-
item-action active">
                Login
            </h3>

            <div className="form-group list-group-
item list-group-item-action">
                <label >Username</label>
                <input type="text" className="form-
control" id="username" aria-describedby="emailHelp"/>
            </div>
            <div className="form-group list-group-
item list-group-item-action">
                <label>Password</label>
                <input type="password"
className="form-control" id="password"
placeholder="Password" />
            </div>

            <div className="list-group-item list-group-
item-action disabled ds-button">
                <Link to={"/SignUp"}>
                    <button type="button" className="btn
btn-primary btn-lg ds-button">SignUp</button>
                </Link>
                <button type="button" className="btn
btn-primary btn-lg" onClick={this.login}>Login</button>
            </div>

```

```

        </div>
    );
}
}

export default Login;

```

#### Dashboard.js

```

import React, {Component} from 'react';

import Header from '../components/Header';
import AllPharmacyItems from
'../components/PharmacySelection/AllPharmacyItems';
import SelectedPharmacy from
'../components/PharmacySelection/SelectedPharmacy';
import ButtonBar from '../components/ButtonBar';
import Breadcrumb from '../components/BreadCrumb';
import TotalRow from
'../components/PharmacySelection/TotalRow';

import '../scss/PharmacyItems.css'

class Dashboard extends Component {

    constructor(props) {
        super(props);

        this.state = {
            selectedPharmacy: []
        }

        this.addPharmacyToSelectedPharmacy =
this.addPharmacyToSelectedPharmacy.bind(this);
        this.deletePharmacyFromSelectedPharmacy =
this.deletePharmacyFromSelectedPharmacy.bind(this);
    }

    addPharmacyToSelectedPharmacy(pharmacy) {
        let selectedPharmacy = this.state.selectedPharmacy;
        let previous = null;

        for(let i=0; i<selectedPharmacy.length; i++) {
            if(pharmacy.name === selectedPharmacy[i].name)
            {
                previous = selectedPharmacy[i];
                previous.quantity = previous.quantity +

```

```

    pharmacy.quantity;
    previous.price = previous.price +
    pharmacy.price;
    }

    }

    if(previous === null) {
        selectedPharmacy.push(pharmacy);
    }
    this.setState(selectedPharmacy);
}

deletePharmacyFromSelectedPharmacy (index) {
    const selectedPharmacy =
this.state.selectedPharmacy;

    selectedPharmacy.splice(index,1);

    this.setState(selectedPharmacy);
}

render() {
    return(
        <div className='ds-container'>

            <Header/>
            <Breadcrumb path={'Home'}/>

            <div className="ds-pharmacy-selection">
                <h3>Medic Selection</h3>
                <div className="row">

                    <div className="col-md-5">
                        <AllPharmacyItems
addPharmacyToSelectedPharmacy={this.addPharmacyToSelectedPh
armacy} disable={false}/>
                    </div>

                    <div className="col">
                        <SelectedPharmacy
selectedPharmacy={this.state.selectedPharmacy}
deletePharmacy={this.deletePharmacyFromSelectedPharmacy}/>

```

```

        </div>

        </div>
    </div>
    <div className="row ds-total">
        <div className="col">
            <TotalRow
selectedPharmacy={this.state.selectedPharmacy} />
        </div>
    </div>
    <ButtonBar work={'Payments'} />

    </div>
    );
}
}

```

**export default** Dashboard;

Admins Dashboard.js

```

import React, {Component} from 'react';
import * as API from '../Constants/APIs';
import axios from 'axios';

import Header from '../components/Header';
import AllPharmacyItems from
'../components/PharmacySelection/AllPharmacyItems';
import SelectedPharmacy from
'../components/FoodSelection/SelectedPharmacy';
import ButtonBar from '../components/ButtonBar';
import Breadcrumb from '../components/BreadCrumb';
import TotalRow from
'../components/PharmacySelection/TotalRow';

import '../scss/PharmacyItems.css'

```

```

class Dashboard extends Component {

    constructor(props) {
        super(props);

        this.state = {
            selectedPharmacy: []
        }

        this.addPharmacyToSelectedPharmacy =

```

```

this.addPharmacyToSelectedPharmacy.bind(this);
    this.deletePharmacyFromSelectedPharmacy =
this.deletePharmacyFromSelectedPharmacy.bind(this);
}

addPharmacyToSelectedPharmacy(pharmacy) {
    let selectedPharmacy = this.state.selectedPharmacy;
    let previous = null;

    for(let i=0; i<selectedPharmacy.length; i++) {
        if(pharmacy.name === selectedPharmacy[i].name)
        {
            previous = selectedPharmacy[i];
            previous.quantity = previous.quantity +
pharmacy.quantity;
            previous.price = previous.price +
pharmacy.price;
        }

    }

    if(previous === null) {
        selectedPharmacy.push(pharmacy);
    }
    this.setState(selectedPharmacy);
}

deletePharmacyFromSelectedPharmacy (index) {
    const selectedPharmacy =
this.state.selectedPharmacy;

    selectedPharmacy.splice(index,1);

    this.setState(selectedPharmacy);
}

addPharmacy() {
    let addPharmacy = this;
    const body = {
        name:document.getElementById('name').value,
        price:document.getElementById('price').value
    }
    axios.post(API.Get_All_Pharmacy,
body).then(function (responce) {
        //window.alert(responce.status);
        window.location.reload();
    });
}

```

```

    }).catch(function (err) {
        console.log(err);
    });
}

render() {
    return(
        <div className='ds-container'>

            <Header/>
            <Breadcrumb path={'Admin Dashboard'} />

            <div className="ds-pharmacy-selection">
                <h3>Admin DashBoard</h3>
                <div className="row">

                    <div className="col-md-6">
                        <AllPharmacyItems
addFoodToSelectedPharmacy={this.addPharmacyToSelectedPharmacy}
                        disable={true} />
                    </div>

                    <div className="list-group col">
                        <h3 className="list-group-item
list-group-item-action active">
                            Add Medicine
                        </h3>

                        <div className="form-group
list-group-item list-group-item-action">
                            <label >Pharmacy Item
Name</label>

                            <input type="text"
className="form-control" id="name" aria-
describedby="emailHelp"/>

                            <label >Price Rs.</label>
                            <input type="text"
className="form-control" id="price" aria-
describedby="emailHelp"/>

                        </div>

```

```

                                <div className="list-group-item
list-group-item-action disabled ds-button">
                                <button type="button"
className="btn btn-primary" onClick={this.addPharmacy}>Add
Medicine</button>
                                </div>
                                </div>

                                </div>
                                </div>

                                <ButtonBar work={ 'Customer Dashboard' }/>

                                </div>
                                );
                                }
                                }

export default Dashboard;

```