Requirement: เช่า/ยืมเครื่องดนตรี

เป็นระบบไว้สำาหรับลูกค้าที่ต้องการจะเช่า หรือยืมเครื่องดนตรีไปซ้อมหรือไปเล่นในงานต่างๆ ทั้งในและ นอกสถานที่ โดยการเช้า/ยืมเครื่องดนตรีนั้นลูกค้าจะเข้าระบบแล้วเลือกเครื่องดนตรี เลือกวันและเวลา ที่ต้องการ เมื่อลูกค้าทำาการเลือกเสร็จและยืนยันการเช่าโดยการกรอกข้อมูลยืนยันตัวตนอีกครั้ง เสร็จแล้วระบบจะแสดงราย ละเอียดข้อมูลการเช่า และช่องทางหรือวิธีการชำาระเงินให้กบลูกค้าได้ทราบ

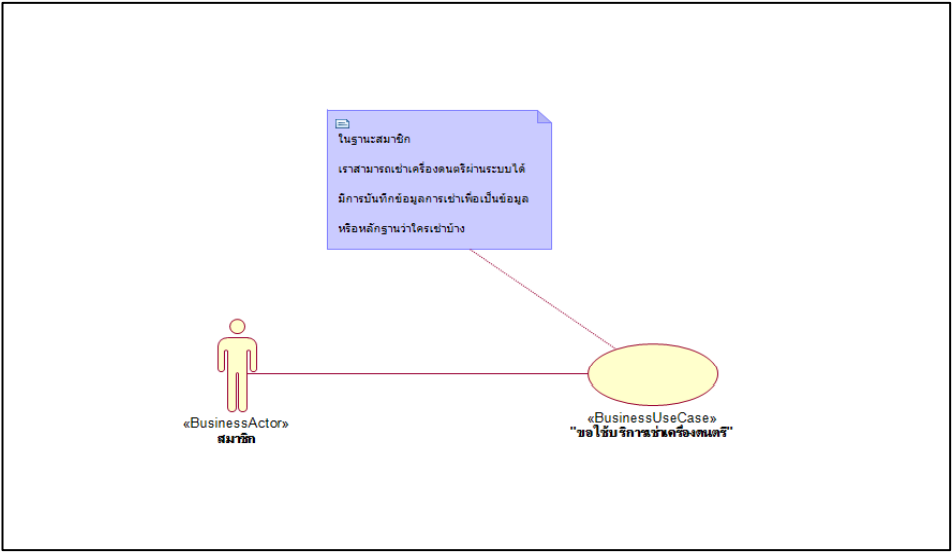
-ในฐานะของ : ลูกค้า

- ฉันต้องการ : เช่า หรือ ยืมเครื่องดนตรี

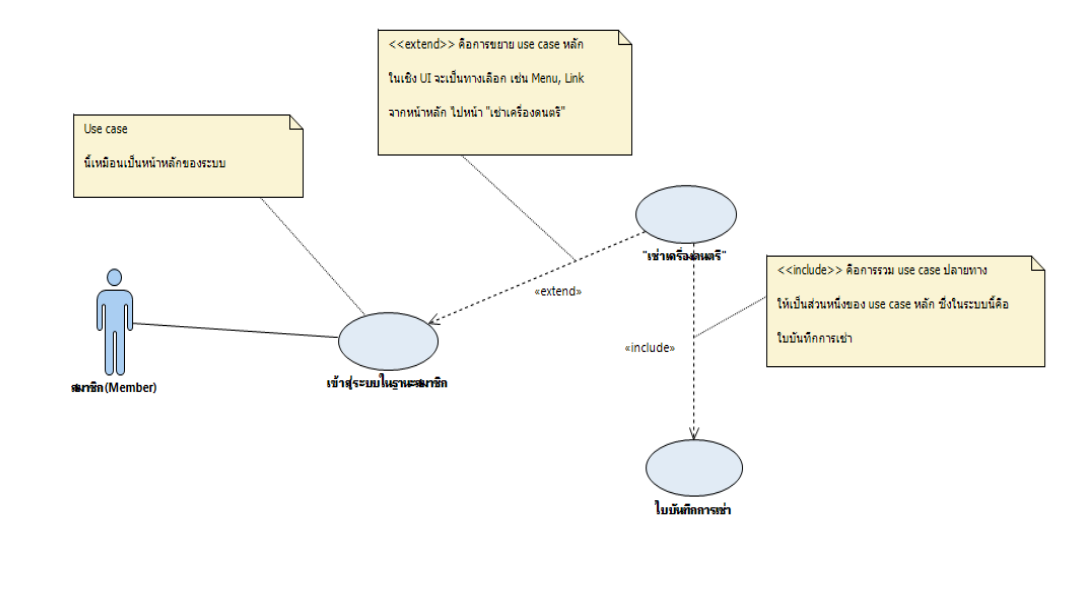
- เพื่อ : - สะดวกและรวดเร็วในการเช่าเครื่องดนตรี - นำเครื่องดนตรีไปซ้อมหรือเล่นในงานต่างๆได้

Output : ใบเสร็จ หรือใบบันทึกรายการ

Businness Diagram



System Diagram



**ไฟล์ หน้า UI .js**

|  |
| --- |
| var React = require('react');  var ReactDOM = require('react-dom');  var ons = require('onsenui');  var Ons = require('react-onsenui');  var client = require('./client');  var page = true;  var page2 = true;  var code = '';  var index = [0,0];  var pro = ['กลองชุด ราคา 500 บาท/วัน','กีตาร์ไฟฟ้า ราคา 450 บาท/วัน','กีตาร์โปร่ง ราคา 450 บาท/วัน','เบส ราคา 450 บาท/วัน','คีย์บอร์ด ราคา 450 บาท/วัน'];  var URL = ["https://www.uppic.org/thumb-0E98\_599FE98A.jpg","https://www.uppic.org/thumb-BF40\_59A6DAF3.jpg","https://www.uppic.org/thumb-A7C3\_59A6DC4B.jpg",  "https://www.uppic.org/thumb-72F6\_59A6DCD8.jpg","https://www.uppic.org/thumb-3483\_59A6DD4B.jpg"];  var index = 1;  var temp = 0;  var MyPage = React.createClass({  getInitialState: function() {  return {  temp:'',  member: '',  date: '',  returndate:'',  musicalIns: [  '01',  '02',  '03',  '04',  '05'  ],  selectedMusicalIns: 'No'    };  },    renderToolbar: function(route, navigator) {    const backButton = route.hasBackButton  ? <Ons.BackButton onClick={this.handleClick.bind(this, navigator)}>Back</Ons.BackButton>  : null;  return (  <Ons.Toolbar>  <div className='left'>{backButton}</div>  <div className='center'>{route.title}</div>  </Ons.Toolbar>    );    },  renderCheckboxRow(row,c) {  return (  <Ons.ListItem key={row} tappable>  <label className='right'>  <Ons.Checkbox  inputId={`checkbox-${row}`}  checked={row === this.state.selectedMusicalIns}  onChange={this.handleMusicalInsChange.bind(this, row)}  />  </label>  <label htmlFor={`checkbox-${pro[c]}`} className = 'center'>  <img src={URL[c]} style={{width: '80',height: '75'}} />&emsp;{pro[c]}  </label>  <label htmlFor={`checkbox-${row}`} className='right'>  </label>  </Ons.ListItem>  );  },    handleClickSave: function(navigator) {  var id = this.state.member  var musical = this.state.selectedMusicalIns  var rentdate = this.state.date  var returndate = this.state.returndate  var price = this.state.selectedMusicalIns  if(price === '01'){  temp = '500';  }else if(price === '02'){  temp = '300';  }else if(price === '03'){  temp = '300';  }else if(price === '04'){  temp = '300';  }else if(price === '05'){  temp = '450';  }  client({method: 'GET', path: '/RentBill/'+id+'/musical/'+musical+'/rentdate/'+rentdate+'/returndate/'+returndate+'/temp/'+temp}).done(  ons.notification.alert('บันทึกสำเร็จ')  )    },    handleClick: function(navigator) {  if(navigator === 1){  ons.notification.confirm('confirm')  .then((response) => {  if (response === 1) {  navigator.popPage();  index--;  }  });  }else{  ons.notification.confirm('คุณต้องการย้อนกลับใช่หรือไม่')  .then((response) => {  if (response === 1) {  navigator.popPage();  index--;  }  });    }  },    handleDateChange(e) {  this.setState({date: e.target.value});  },    handleMemberChange(e) {  this.setState({member: e.target.value});  },    handleReturnDateChange(e) {  this.setState({returndate: e.target.value});  },  handleMusicalInsChange(musicalIns) {  this.setState({selectedMusicalIns: musicalIns});  },    pushPage: function(navigator) {      if (this.state.selectedMusicalIns != 'No'){  navigator.pushPage({  title: `Another page ${index}`,  hasBackButton: true  });  index++;  }  else {  ons.notification.alert('Not Found!');  }  },  renderPage: function(route, navigator) {  var item = this.state.selectedMusicalIns  var r = this.state.date;  if(index==1){  return (  <Ons.Page key={route.title} renderToolbar={this.renderToolbar.bind(this, route, navigator)}>  <section style={{textAlign: 'center'}}>  <p>  <Ons.Input  value={this.state.member}  onChange={this.handleMemberChange}  modifier='underbar'  float  placeholder='กรอกรหัสสมาชิก' />  </p>    <p>  <Ons.Input  value={this.state.date}  onChange={this.handleDateChange}  modifier='underbar'  float  placeholder='วันที่เช่า' />  </p>    <p>  <Ons.Input  value={this.state.returndate}  onChange={this.handleReturnDateChange}  modifier='underbar'  float  placeholder='วันที่ส่งคืน' />  </p>  </section>          <Ons.List  dataSource={this.state.musicalIns}  renderHeader={() => <Ons.ListHeader>เลือกเครื่องดนตรี</Ons.ListHeader>}  renderRow={this.renderCheckboxRow}  />        <section style={{margin: '16px', textAlign: 'center'}}>  <Ons.Button onClick={this.pushPage.bind(this, navigator)}>ถัดไป</Ons.Button>  </section>  </Ons.Page>  );  }else if(index == 2){    var id = this.state.member  var musical = this.state.selectedMusicalIns  var r = this.state.date  var re = this.state.returndate  var price = this.state.selectedMusicalIns  if(price === '01'){  temp = '500';  }else if(price === '02'){  temp = '300';  }else if(price === '03'){  temp = '300';  }else if(price === '04'){  temp = '300';  }else if(price === '05'){  temp = '450';  }  return (  <Ons.Page key={route.title} renderToolbar={this.renderToolbar.bind(this, route, navigator)}>  <Ons.ListHeader>รหัสสมาชิก : {id} </Ons.ListHeader>  <Ons.ListHeader>รหัสเครื่องดนตรี : {musical} </Ons.ListHeader>  <Ons.ListHeader>วันที่เช่า : {r}</Ons.ListHeader>  <Ons.ListHeader>วันที่ส่งคืน : {re}</Ons.ListHeader>  <Ons.ListHeader>ราคา : {temp}</Ons.ListHeader>  <section style={{margin: '16px', textAlign: 'center'}}>  <Ons.Button onClick={this.handleClickSave.bind(this, navigator)}>Save</Ons.Button><br/><br/>  <Ons.Button onClick={this.handleClick.bind(this, 1)}>confirm</Ons.Button>    </section>  </Ons.Page>  );    }  },  render: function() {    return (  <Ons.Navigator  swipeable  renderPage={this.renderPage}  initialRoute={{  title: 'เช่าเครื่องดนตรี',  hasBackButton: false  }}  />    );    }  });  ons.ready(function() {  ReactDOM.render(<MyPage />, document.getElementById('react'));  }); |

**ไฟล์ java ของ entity calss**

|  |
| --- |
| package com.example.demo;  import lombok.Data;  import javax.persistence.Id;  import javax.persistence.Entity;  import javax.persistence.GeneratedValue;  import javax.persistence.ManyToOne;  *@Data*  *@Entity*  public class RentBill {      private @Id      @GeneratedValue  *Long* billId;  *@ManyToOne*      private *MusicalInstrument* musicType;  private *String* member;      private *String* rentdate;      private *String* returndate;      private *String* price;      private RentBill() {}      public RentBill(*MusicalInstrument* *musicType*, *String* *member*,*String* *rentdate*, *String* *returndate*,*String* *price*) {          this.member = member;          this.musicType = musicType;          this.rentdate = rentdate;          this.returndate = returndate;          this.price = price;      }  } |

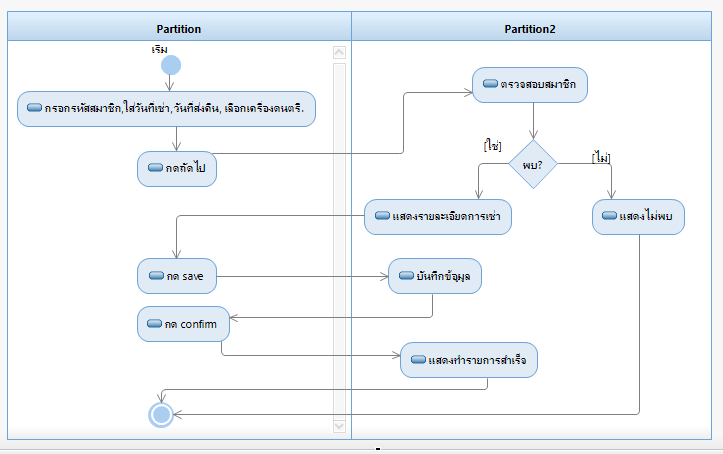
|  |
| --- |
| package com.example.demo;  import javax.persistence.Entity;  import javax.persistence.GeneratedValue;  import javax.persistence.Id;  import lombok.Data;  *@Data*  *@Entity*  public class MusicalInstrument {      private @Id @*GeneratedValue* *Long* id;      private *String* name;      private *Double* price;      private MusicalInstrument () {}      public MusicalInstrument (*String* *name*, *Double* *price*) {          this.name = name;          this.price = price;      }  } |

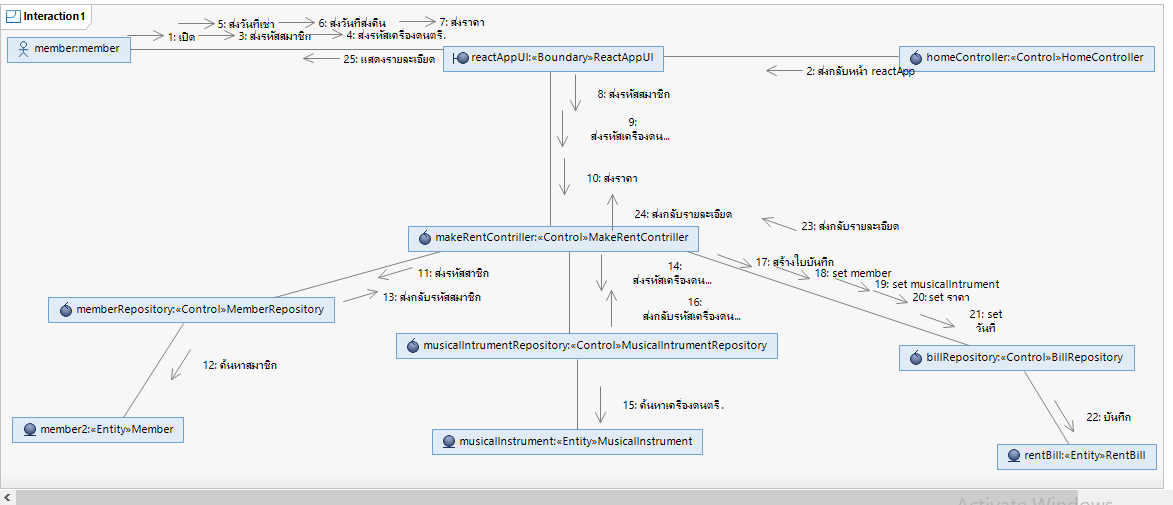
**ไฟล์ java ของ controller**

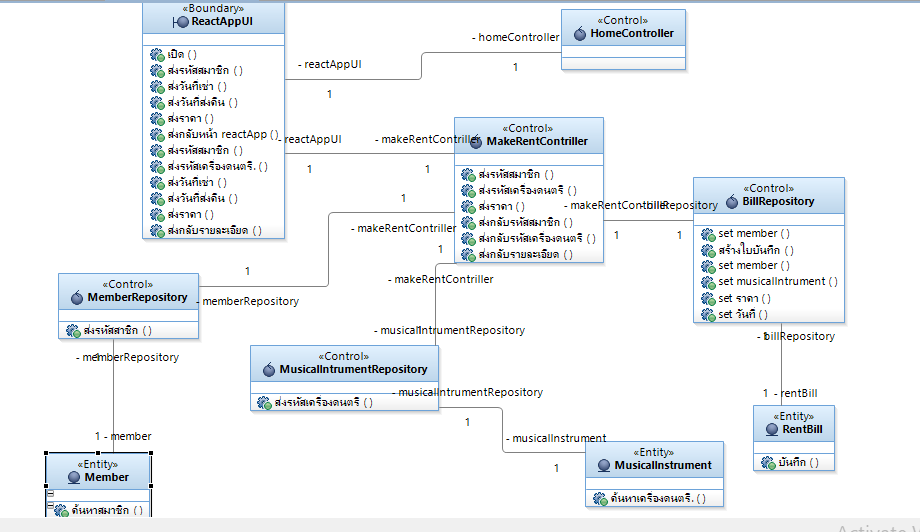
|  |
| --- |
| package com.example.demo;  import org.springframework.beans.factory.annotation.Autowired;  import org.springframework.data.web.PagedResourcesAssembler;  import org.springframework.hateoas.PagedResources;  import org.springframework.http.HttpEntity;  import org.springframework.http.HttpStatus;  import org.springframework.http.MediaType;  import org.springframework.http.ResponseEntity;  import org.springframework.stereotype.Controller;  import org.springframework.web.bind.annotation.\*;  import java.util.List;  import java.util.concurrent.atomic.AtomicLong;  *@Controller*  public class MakeRentController {  *@Autowired*  *RentBillRepository* rentBillRepository;  *@Autowired*  *MusicalInstrumentRepository* musicalInstrumentRepository;  *@ResponseBody*  *@RequestMapping*(path = "/RentBill/{id}/musical/{musical}/rentdate/{rentdate}/returndate/{returndate}/price/{temp}", method = RequestMethod.GET)  public *String* rentbill(*@PathVariable* *String* *id*,*@PathVariable* *Long* *musical*,*@PathVariable* *String* *rentdate*,*@PathVariable* *String* *returndate*,*@PathVariable* *String* *temp*) {  *MusicalInstrument* musicalInstrument = this.musicalInstrumentRepository.findOne(musical);  *RentBill* rentbill = new RentBill(musicalInstrument,id,rentdate,returndate,temp);  this.rentBillRepository.save(rentbill);  return "{\"status\":\"Voted\"}";  }  } |

|  |
| --- |
| package com.example.demo;  import org.springframework.data.repository.CrudRepository;  import java.util.List;  public interface MusicalInstrumentRepository extends *CrudRepository*<*MusicalInstrument*, *Long*> {  } |

|  |
| --- |
| package com.example.demo;  import org.springframework.data.repository.CrudRepository;  import java.util.List;  public interface RentBillRepository extends *CrudRepository*<*RentBill*, *Long*> {  } |

****

****

****

