Information Management and Systems Engineering Milestone 2

Nemanja Srdanovic
Matrikelnummer: 01576891

Adem Mehremic
Matrikelnummer: 01650669

January 2021

Contents

1 Logical / Physical database design 3

1.1 Relational model. 3

1.2 Physical model. 4

2 Web system 6

2.1 Main business use case 6

2.1.1 Login. 6

2.1.2 Execute the purchase. 7

2.2 Elaborate business use case 10

2.2.1 Product sales report per category 10

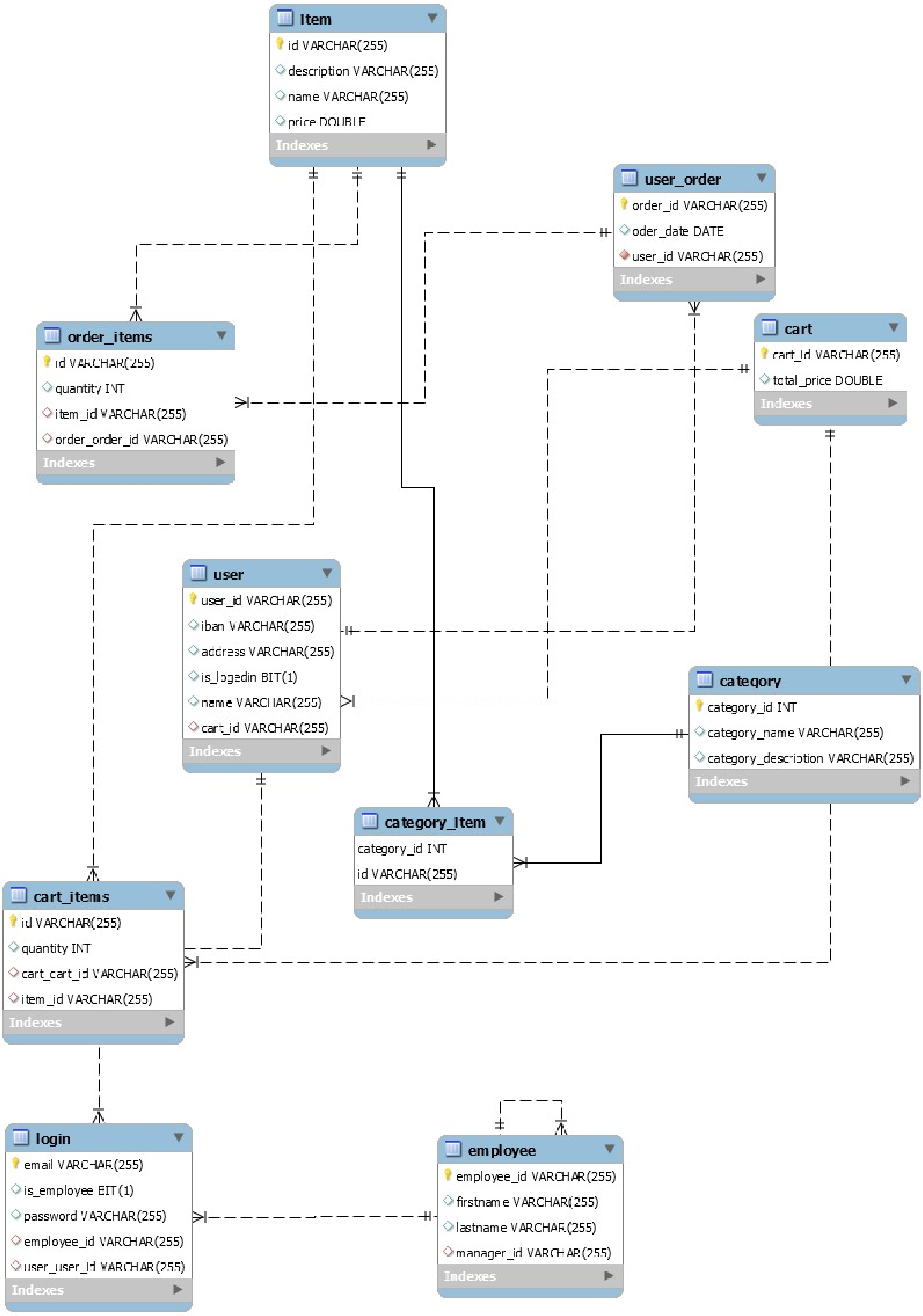
2.2.2 Most sold product 12

3 How to 14

4 Working protocol 15

1 Logical / Physical database design

1.1 Logical model



1.2 Physical model

```
CREATE TABLE IF NOT EXISTS cart (
cart_id VARCHAR(255) NOT NULL,
total_price DOUBLE NULL DEFAULT NULL,
PRIMARY KEY (cart_id)
);

CREATE TABLE IF NOT EXISTS item (
  id VARCHAR(255) NOT NULL,
  description VARCHAR(255) NULL DEFAULT NULL,
  name VARCHAR(255) NULL DEFAULT NULL,
  price DOUBLE NULL DEFAULT NULL,
  PRIMARY KEY (`id`)
);

CREATE TABLE IF NOT EXISTS cart_items (
  id VARCHAR(255) NOT NULL,
  quantity INT NULL DEFAULT NULL,
  cart_cart_id VARCHAR(255) NULL DEFAULT NULL,
  item_id VARCHAR(255) NULL DEFAULT NULL,
  PRIMARY KEY (`id`),
  FOREIGN KEY (`item_id`) REFERENCES  item (id),
  FOREIGN KEY (`cart_cart_id`) REFERENCES cart` (cart_id)
);

CREATE TABLE IF NOT EXISTS category (
  category_id INT NOT NULL AUTO_INCREMENT,
  category_name VARCHAR(255) NULL DEFAULT NULL,
  category_description  VARCHAR(255) NULL DEFAULT NULL,
  PRIMARY KEY (`category_id`)
);

CREATE TABLE IF NOT EXISTS category_item (
  category_id INT NOT NULL,
  id VARCHAR(255) NOT NULL,
  PRIMARY KEY (`category_id`, `id`),
  FOREIGN KEY (id) REFERENCES item` (id),
  FOREIGN KEY (`category_id`) REFERENCES `shopin`.`category` (`category_id`)
);
```

```
CREATE TABLE IF NOT EXISTS  employee (
  employee_id VARCHAR(255) NOT NULL,
  firstname VARCHAR(255) NULL DEFAULT NULL,
  lastname VARCHAR(255) NULL DEFAULT NULL,
  manager_id VARCHAR(255) NULL DEFAULT NULL,
  PRIMARY KEY (`employee_id`),
  FOREIGN KEY (`manager_id`)  REFERENCES employee (employee_id)
);

CREATE TABLE IF NOT EXISTS user (
  user_id VARCHAR(255) NOT NULL,
  iban VARCHAR(255) NULL DEFAULT NULL,
  address VARCHAR(255) NULL DEFAULT NULL,
  is_logedin BIT(1) NULL DEFAULT NULL,
  name VARCHAR(255) NULL DEFAULT NULL,
  cart_id VARCHAR(255) NULL DEFAULT NULL,
  PRIMARY KEY (user_id),
   FOREIGN KEY (cart_id) REFERENCES `shopin`.`cart` (`cart_id`)
);

CREATE TABLE IF NOT EXISTS login` (
  email VARCHAR(255) NOT NULL,
  is_employee BIT(1) NULL DEFAULT NULL,
  password VARCHAR(255) NULL DEFAULT NULL,
  employee_id VARCHAR(255) NULL DEFAULT NULL,
  user_user_id VARCHAR(255) NULL DEFAULT NULL,
  PRIMARY KEY (email),
  FOREIGN KEY (user_user_id) REFERENCES user (user_id),
  FOREIGN KEY (employee_id) REFERENCES employee (employee_id)
);

CREATE TABLE IF NOT EXISTS  user_order (
  order_id VARCHAR(255) NOT NULL,
  oder_date DATE NULL DEFAULT NULL,
  user_id VARCHAR(255) NOT NULL,
  PRIMARY KEY (order_id),
   FOREIGN KEY (`user_id`) REFERENCES user (user_id)
);

CREATE TABLE IF NOT EXISTS  order_items (
  id VARCHAR(255) NOT NULL,
  quantity INT NULL DEFAULT NULL,
  item_id VARCHAR(255) NULL DEFAULT NULL,
  order_order_id VARCHAR(255) NULL DEFAULT NULL,
  PRIMARY KEY (id),
  FOREIGN KEY (order_order_id) REFERENCES user_order (order_id),
  FOREIGN KEY (`item_id`) REFERENCES  item (id)
);
```

2 Web system

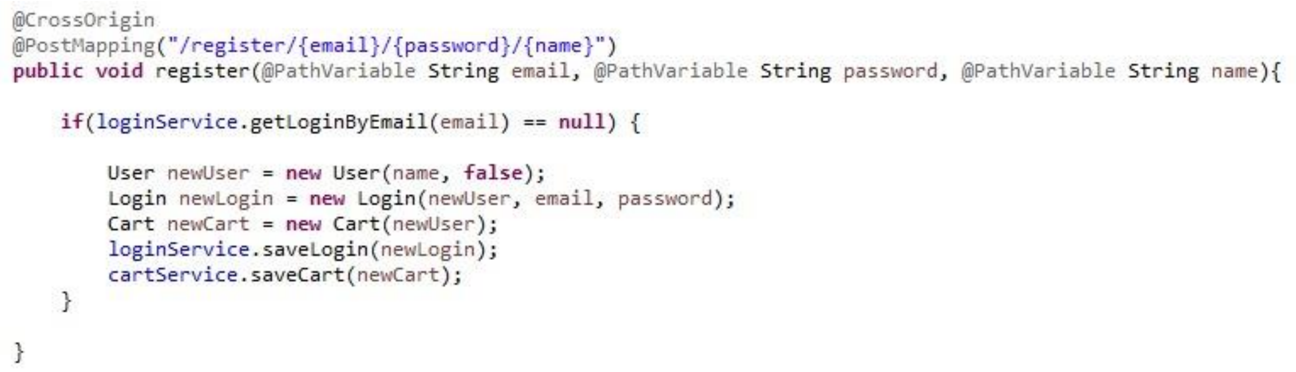
2.1 Main business use case

2.1.1 Login

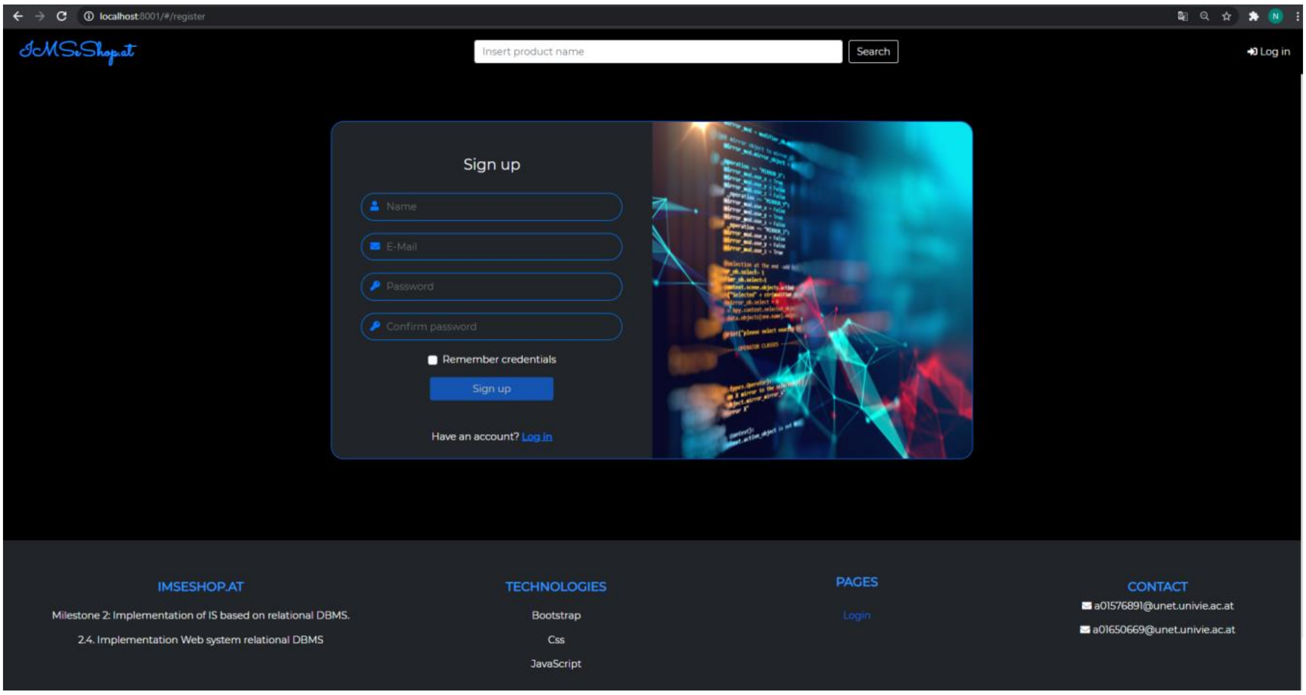
Objective: The customer enters his personal customer area, where he can track his orders, payments etc.

Description: In order to join his customer area, the user has to log in with his username and password. If a person doesn't yet have his login credentials, he must register, whereby if everything fits, he will be directly logged into the customer area. For every other login the customer has to use his login credentials.

Register (Api and Frontend)



```
@CrossOrigin
@PostMapping("/register/{email}/{password}/{name}")
public void register(@PathVariable String email, @PathVariable String password, @PathVariable String name){

    if(loginService.getLoginByEmail(email) == null) {

        User newUser = new User(name, false);
        Login newLogin = new Login(newUser, email, password);
        Cart newCart = new Cart(newUser);
        loginService.saveLogin(newLogin);
        cartService.saveCart(newCart);
    }

}
```



□ Remember credentials

Have an account? Log in

IMSESHOP.AT

Milestone 2: Implementation of IS based on relational DBMS.

2.4. Implementation Web system relational DBMS

TECHNOLOGIES

Bootstrap

Css

JavaScript

PAGES

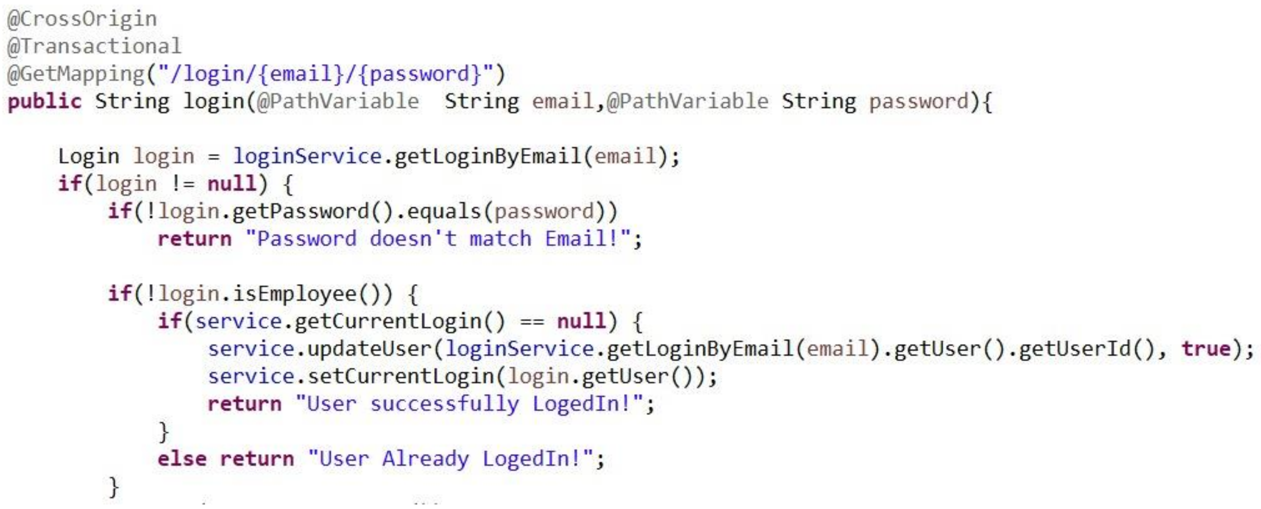
Login

CONTACT

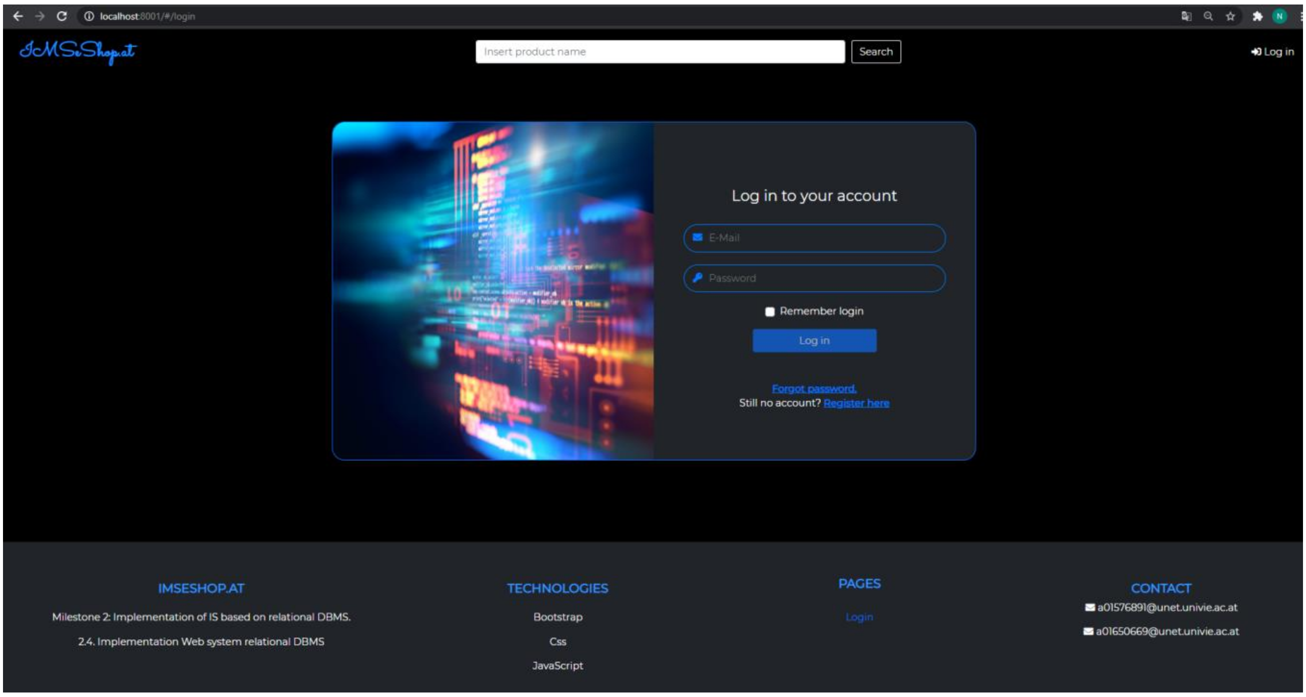
a01576891@unet.univie.ac.at

a01650669@unet.univie.ac.at

Login (Api and Frontend)



```
@CrossOrigin
@Transactional
@GetMapping("/login/{email}/{password}")
public String login(@PathVariable  String email,@PathVariable String password){

    Login login = loginService.getLoginByEmail(email);
    if(login != null) {
        if(!login.getPassword().equals(password))
            return "Password doesn't match Email!";

        if(!login.isEmployee()) {
            if(service.getCurrentLogin() == null) {
                service.updateUser(loginService.getLoginByEmail(email).getUser().getUserId(), true);
                service.setCurrentLogin(login.getUser());
                return "User successfully LogedIn!";
            }
            else return "User Already LogedIn!";
        }
```



Milestone 2: Implementation of IS based on relational DBMS.

2.4. Implementation Web system relational DBMS

a01576891@unet.univie.ac.at

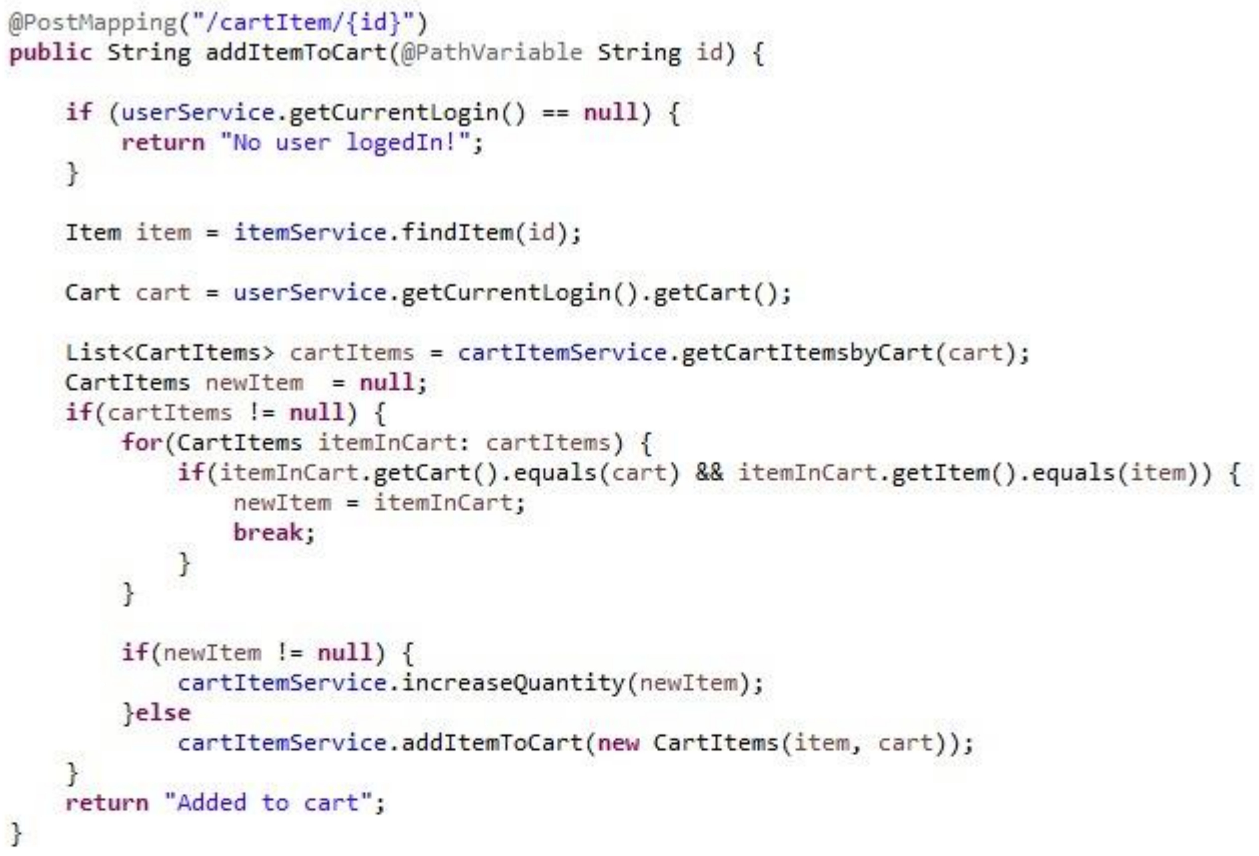
a01650669@unet.univie.ac.at

2.1.2 Execute the purchase

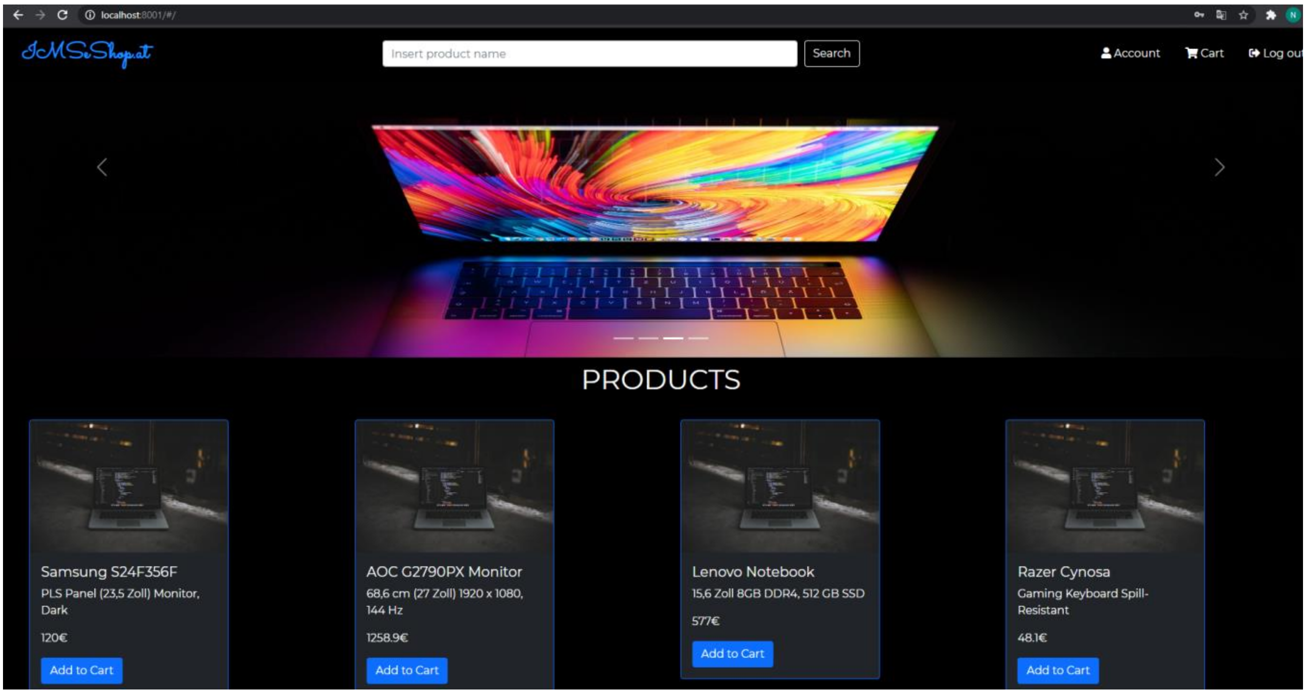
Objective: Purchase the products from the Shopping cart

Description: After the customer finished searching for products that he wants to order he can now proceed to purchase these goods.

Add product to cart (Api and Frontend)



```
@PostMapping("/cartItem/{id}")
public String addItemToCart(@PathVariable String id) {

    if (userService.getCurrentLogin() == null) {
        return "No user logedIn!";
    }

    Item item = itemService.findItem(id);

    Cart cart = userService.getCurrentLogin().getCart();

    List<CartItems> cartItems = cartItemService.getCartItemsbyCart(cart);
    CartItems newItem  = null;
    if(cartItems != null) {
        for(CartItems itemInCart: cartItems) {
            if(itemInCart.getCart().equals(cart) && itemInCart.getItem().equals(item)) {
                newItem = itemInCart;
                break;
            }
        }

        if(newItem != null) {
            cartItemService.increaseQuantity(newItem);
        }else
            cartItemService.addItemToCart(new CartItems(item, cart));
    }
    return "Added to cart";
}
```



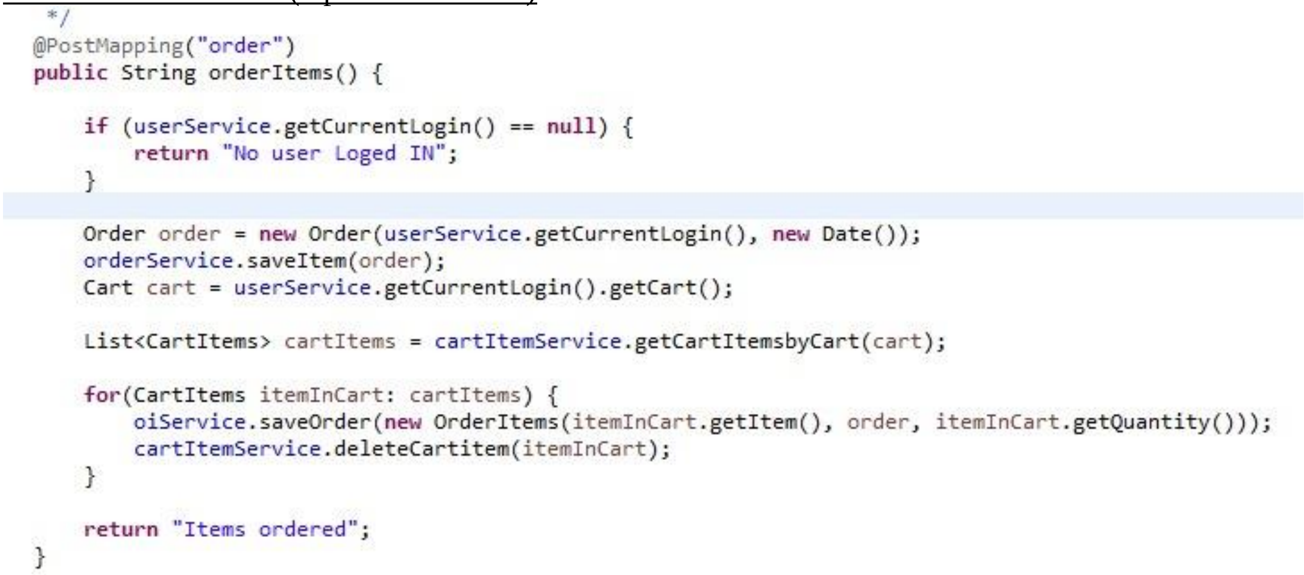
Samsung S24F356F
PLS Panel (23,5 Zoll) Monitor, Dark
120€

AOC G2790PX Monitor
68,6 cm (27 Zoll) 1920 x 1080, 144 Hz
1258.9€

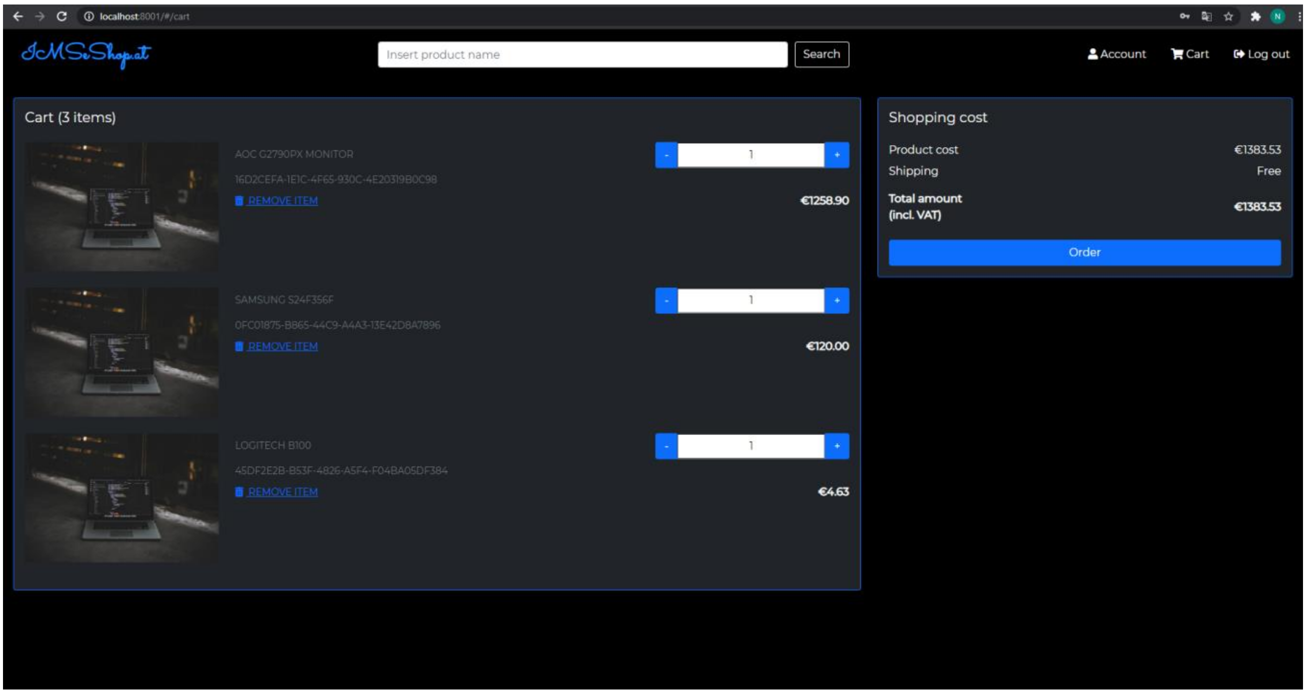
Lenovo Notebook
15,6 Zoll 8GB DDR4, 512 GB SSD
577€

Razer Cynosa
Gaming Keyboard Spill-Resistant
48.1€

Order items from cart (Api and Frontend)



```
 */
@PostMapping("order")
public String orderItems() {

    if (userService.getCurrentLogin() == null) {
        return "No user Loged IN";
    }

    Order order = new Order(userService.getCurrentLogin(), new Date());
    orderService.saveItem(order);
    Cart cart = userService.getCurrentLogin().getCart();

    List<CartItems> cartItems = cartItemService.getCartItemsbyCart(cart);

    for(CartItems itemInCart: cartItems) {
        oiService.saveOrder(new OrderItems(itemInCart.getItem(), order, itemInCart.getQuantity()));
        cartItemService.deleteCartitem(itemInCart);
    }

    return "Items ordered";
}
```



Cart (3 items)

AOC G2790PX MONITOR
16D2CEFA-1E1C-4F65-930C-4E20319B0C98
REMOVE ITEM
- 1 +
€1258.90

SAMSUNG S24F356F
0FC01875-B865-44C9-A4A3-13E42D8A7896
REMOVE ITEM
- 1 +
€120.00

LOGITECH B100
45DF2E2B-B53F-4826-A5F4-F04BA05DF384
REMOVE ITEM
- 1 +
€4.63

Shopping cost

Product cost	€1383.53
Shipping	Free
Total amount (incl. VAT)	**€1383.53**

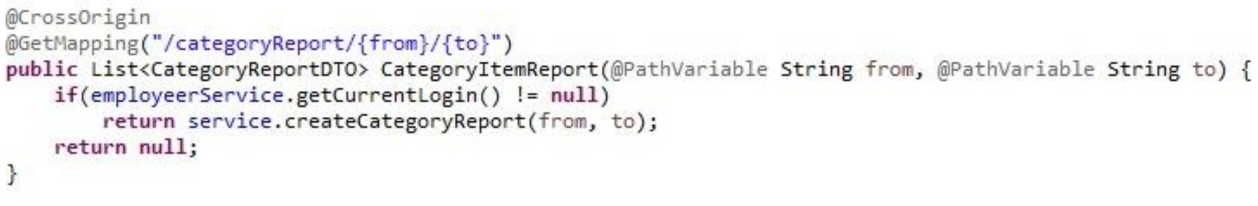
2.2 Elaborate reporting use case

2.2.1 Product sales report per category

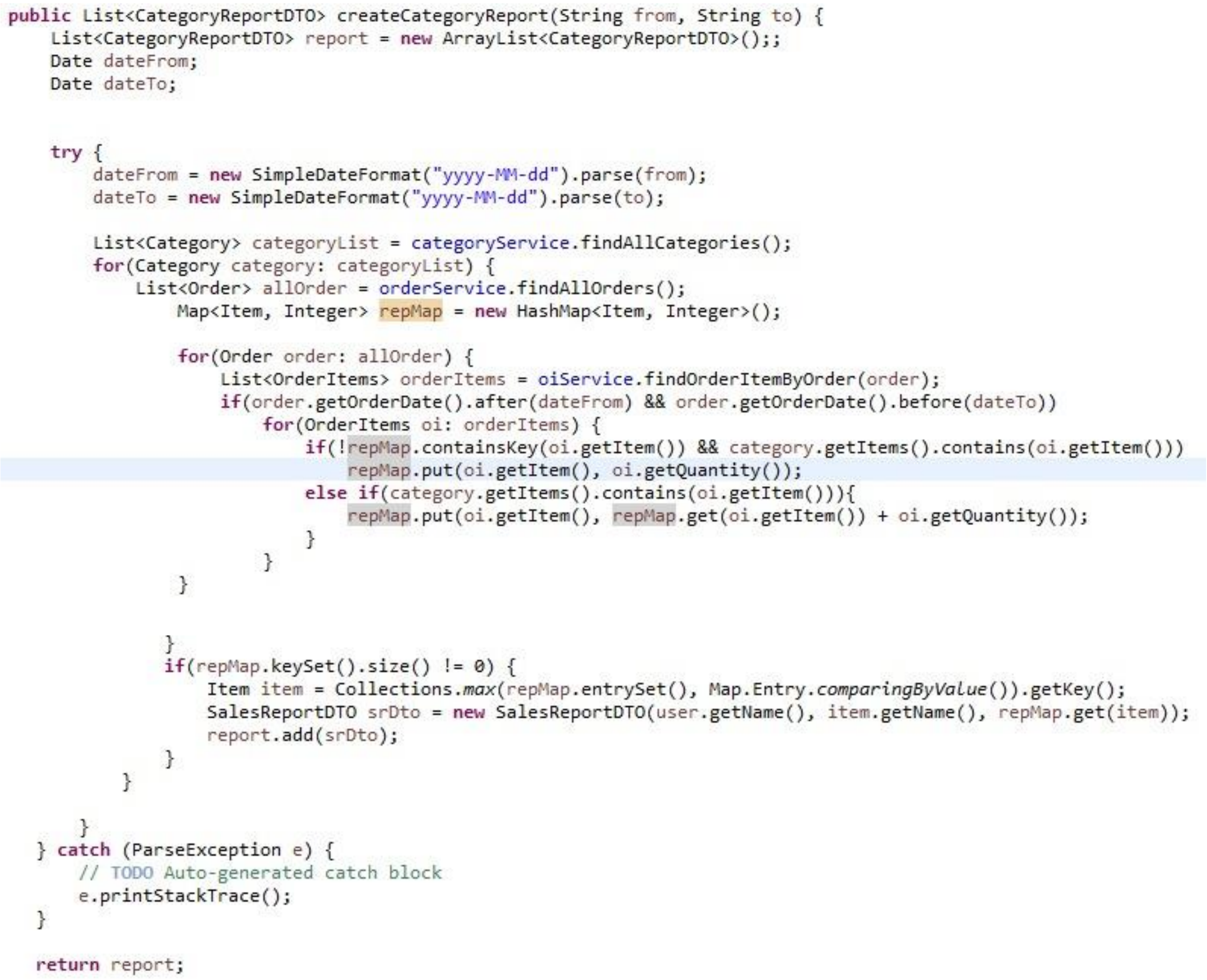
Objective: Be able to see the top selling product per month (order date) per category.

Description**:** Employees can see precise statistic on product sales per category per month.

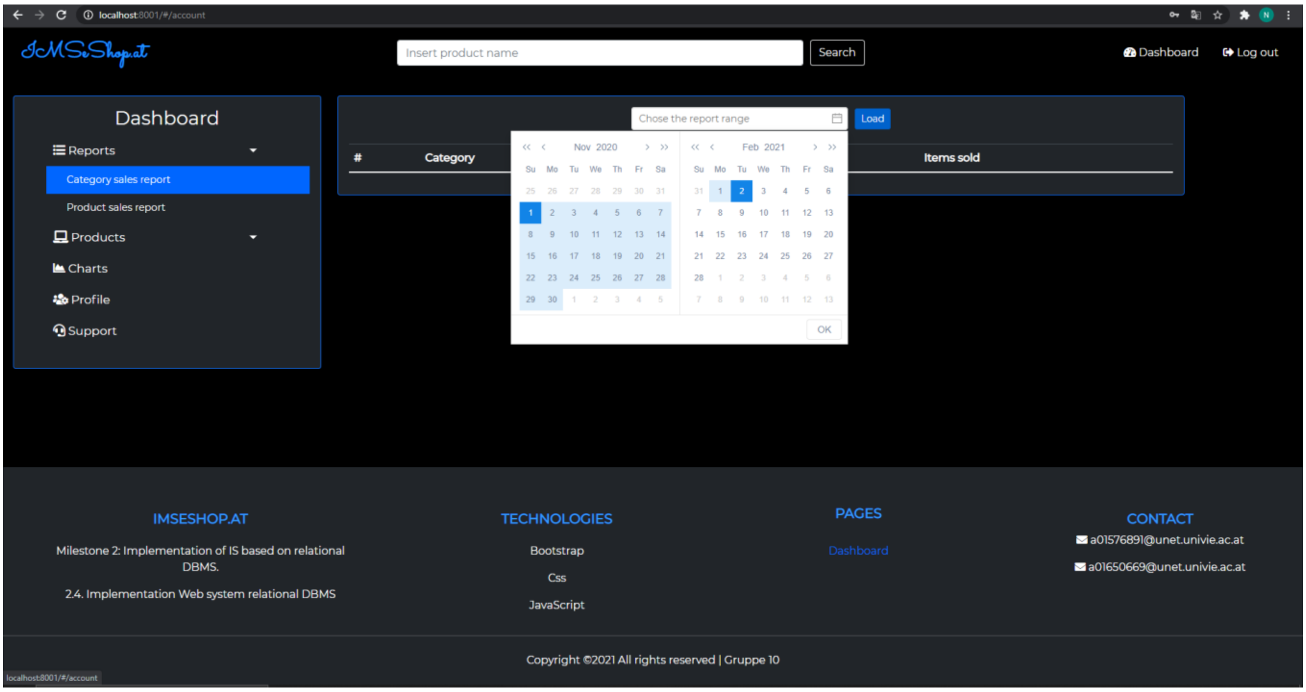
Category report (Api and Frontend)



```
@CrossOrigin
@GetMapping("/categoryReport/{from}/{to}")
public List<CategoryReportDTO> CategoryItemReport(@PathVariable String from, @PathVariable String to) {
    if(employeerService.getCurrentLogin() != null)
        return service.createCategoryReport(from, to);
    return null;
}
```



```
public List<CategoryReportDTO> createCategoryReport(String from, String to) {
    List<CategoryReportDTO> report = new ArrayList<CategoryReportDTO>();;
    Date dateFrom;
    Date dateTo;


    try {
        dateFrom = new SimpleDateFormat("yyyy-MM-dd").parse(from);
        dateTo = new SimpleDateFormat("yyyy-MM-dd").parse(to);

        List<Category> categoryList = categoryService.findAllCategories();
        for(Category category: categoryList) {
            List<Order> allOrder = orderService.findAllOrders();
                Map<Item, Integer> repMap = new HashMap<Item, Integer>();

                for(Order order: allOrder) {
                    List<OrderItems> orderItems = oiService.findOrderItemByOrder(order);
                    if(order.getOrderDate().after(dateFrom) && order.getOrderDate().before(dateTo))
                        for(OrderItems oi: orderItems) {
                            if(!repMap.containsKey(oi.getItem()) && category.getItems().contains(oi.getItem()))
                                repMap.put(oi.getItem(), oi.getQuantity());
                            else if(category.getItems().contains(oi.getItem())){
                                repMap.put(oi.getItem(), repMap.get(oi.getItem()) + oi.getQuantity());
                            }
                        }
                }


                }
                if(repMap.keySet().size() != 0) {
                    Item item = Collections.max(repMap.entrySet(), Map.Entry.comparingByValue()).getKey();
                    SalesReportDTO srDto = new SalesReportDTO(user.getName(), item.getName(), repMap.get(item));
                    report.add(srDto);
                }
            }

        }
    } catch (ParseException e) {
        // TODO Auto-generated catch block
        e.printStackTrace();
    }

    return report;
```



IMSESHOP.AT

Milestone 2: Implementation of IS based on relational DBMS.

2.4. Implementation Web system relational DBMS

TECHNOLOGIES

Bootstrap

Css

JavaScript

PAGES

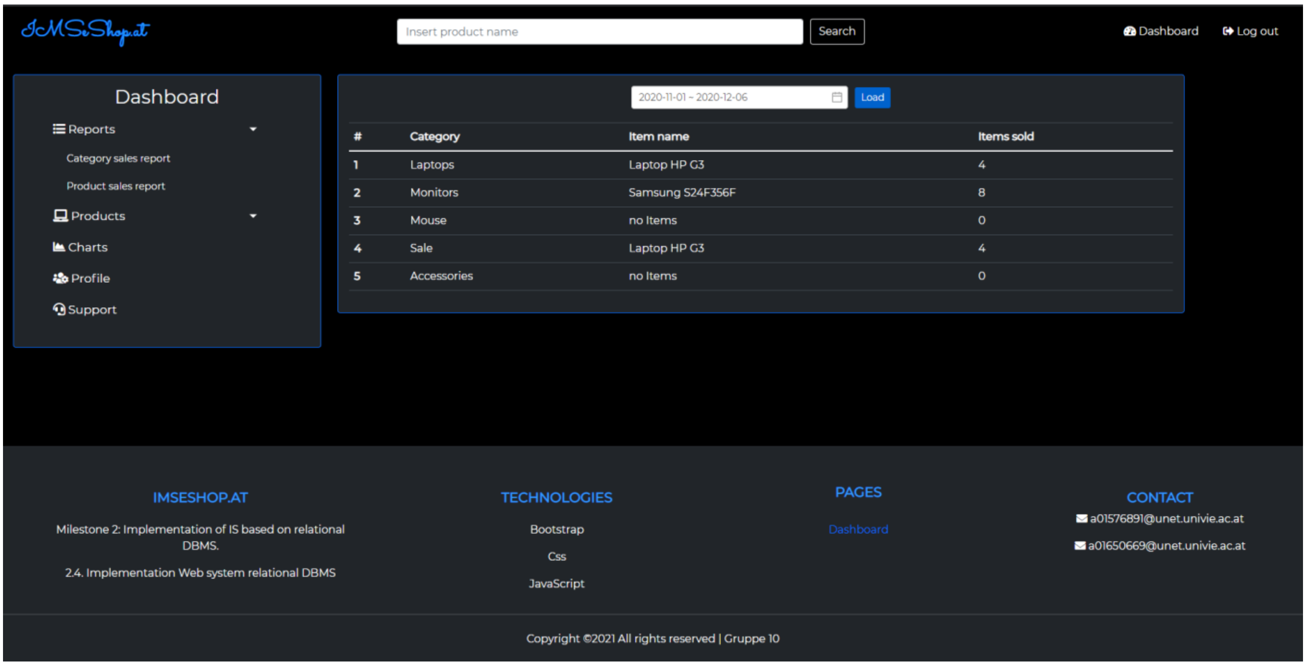
Dashboard

CONTACT

a01576891@unet.univie.ac.at

a01650669@unet.univie.ac.at

Copyright ©2021 All rights reserved | Gruppe 10



#	Category	Item name	Items sold
1	Laptops	Laptop HP G3	4
2	Monitors	Samsung S24F356F	8
3	Mouse	no Items	0
4	Sale	Laptop HP G3	4
5	Accessories	no Items	0

IMSESHOP.AT

Milestone 2: Implementation of IS based on relational DBMS.

2.4. Implementation Web system relational DBMS

TECHNOLOGIES

Bootstrap

Css

JavaScript

PAGES

Dashboard

CONTACT

a01576891@unet.univie.ac.at

a01650669@unet.univie.ac.at

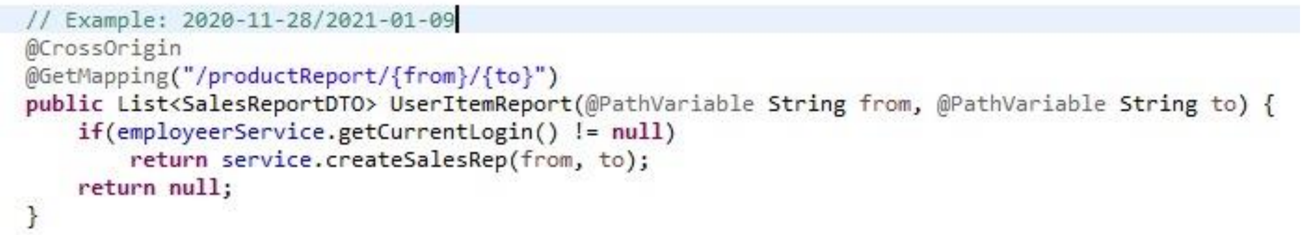
Copyright ©2021 All rights reserved | Gruppe 10

2.2.2 Most sold product

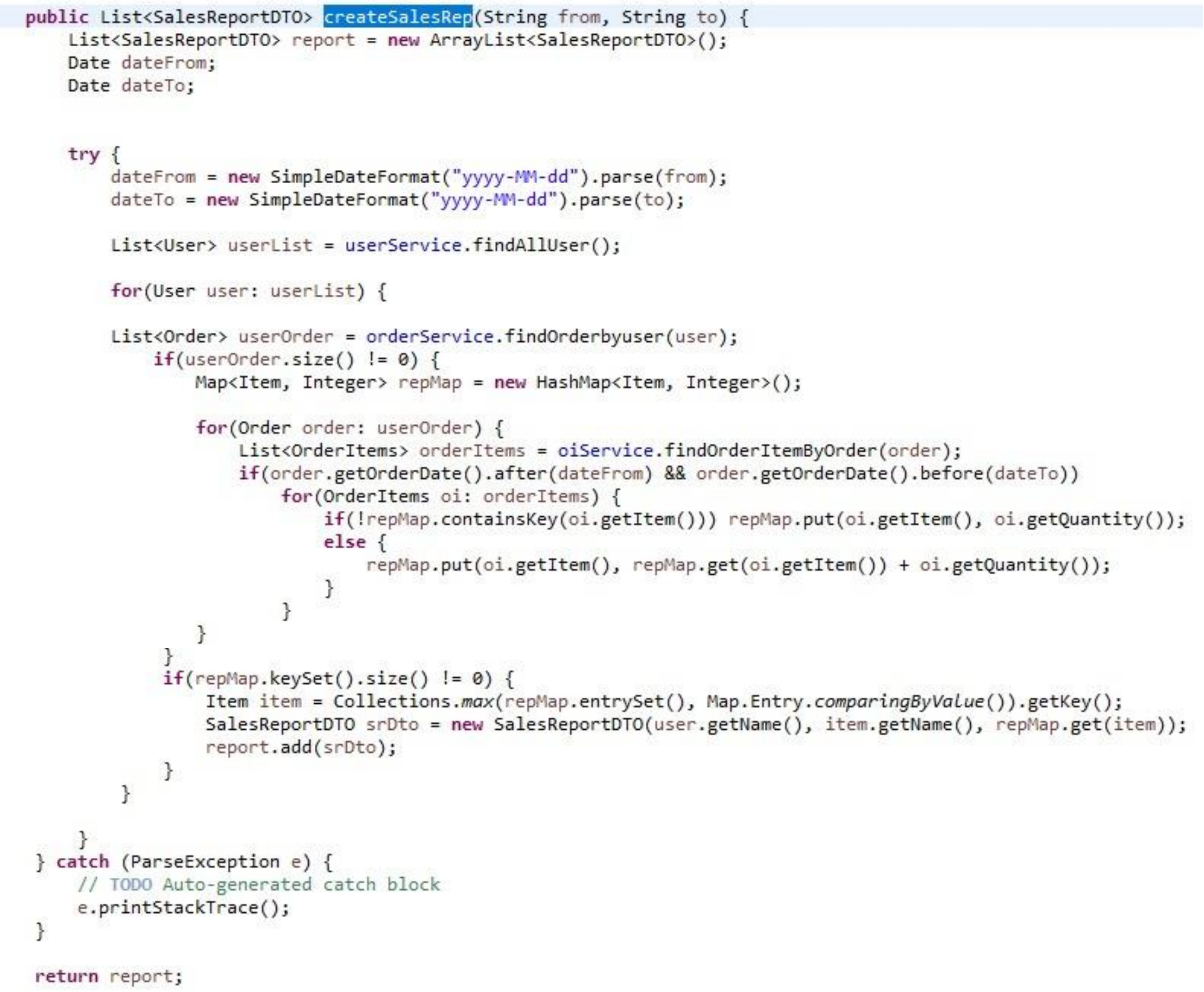
Objective: Find bestselling products pro customer

Description: Administrator can find out what their best-selling Products are to recommend them, and stock more of them.

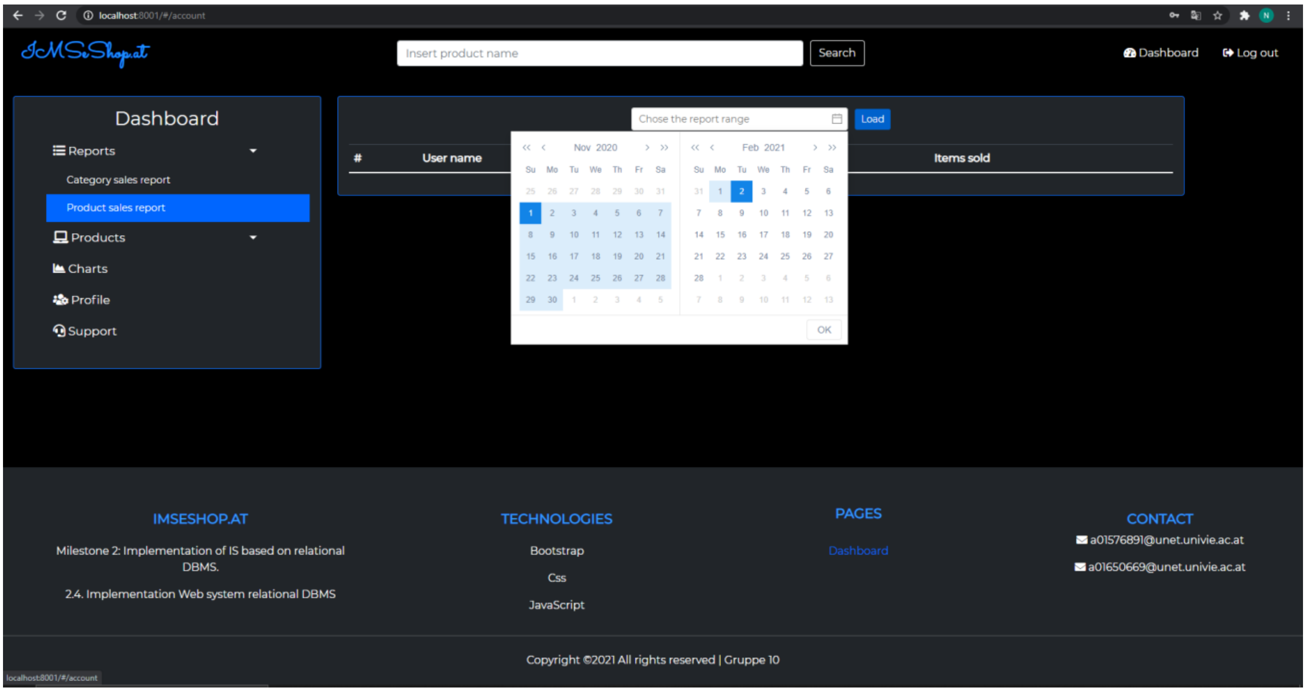
Product report (Api and Frontend)



```
// Example: 2020-11-28/2021-01-09
@CrossOrigin
@GetMapping("/productReport/{from}/{to}")
public List<SalesReportDTO> UserItemReport(@PathVariable String from, @PathVariable String to) {
    if(employeerService.getCurrentLogin() != null)
        return service.createSalesRep(from, to);
    return null;
}
```



```
public List<SalesReportDTO> createSalesRep(String from, String to) {
    List<SalesReportDTO> report = new ArrayList<SalesReportDTO>();
    Date dateFrom;
    Date dateTo;


    try {
        dateFrom = new SimpleDateFormat("yyyy-MM-dd").parse(from);
        dateTo = new SimpleDateFormat("yyyy-MM-dd").parse(to);

        List<User> userList = userService.findAllUser();

        for(User user: userList) {

        List<Order> userOrder = orderService.findOrderbyuser(user);
            if(userOrder.size() != 0) {
                Map<Item, Integer> repMap = new HashMap<Item, Integer>();

                for(Order order: userOrder) {
                    List<OrderItems> orderItems = oiService.findOrderItemByOrder(order);
                    if(order.getOrderDate().after(dateFrom) && order.getOrderDate().before(dateTo))
                        for(OrderItems oi: orderItems) {
                            if(!repMap.containsKey(oi.getItem())) repMap.put(oi.getItem(), oi.getQuantity());
                            else {
                                repMap.put(oi.getItem(), repMap.get(oi.getItem()) + oi.getQuantity());
                            }
                        }
                }
            }
            if(repMap.keySet().size() != 0) {
                Item item = Collections.max(repMap.entrySet(), Map.Entry.comparingByValue()).getKey();
                SalesReportDTO srDto = new SalesReportDTO(user.getName(), item.getName(), repMap.get(item));
                report.add(srDto);
            }
        }

    }
 } catch (ParseException e) {
     // TODO Auto-generated catch block
     e.printStackTrace();
 }

 return report;
```



Su	Mo	Tu	We	Th	Fr	Sa
25	26	27	28	29	30	31
1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30	1	2	3	4	5

Su	Mo	Tu	We	Th	Fr	Sa
31	1	2	3	4	5	6
7	8	9	10	11	12	13
14	15	16	17	18	19	20
21	22	23	24	25	26	27
28	1	2	3	4	5	6
7	8	9	10	11	12	13

#	User name	Items sold

IMSESHOP.AT

Milestone 2: Implementation of IS based on relational DBMS.

2.4. Implementation Web system relational DBMS

TECHNOLOGIES

Bootstrap

Css

JavaScript

PAGES

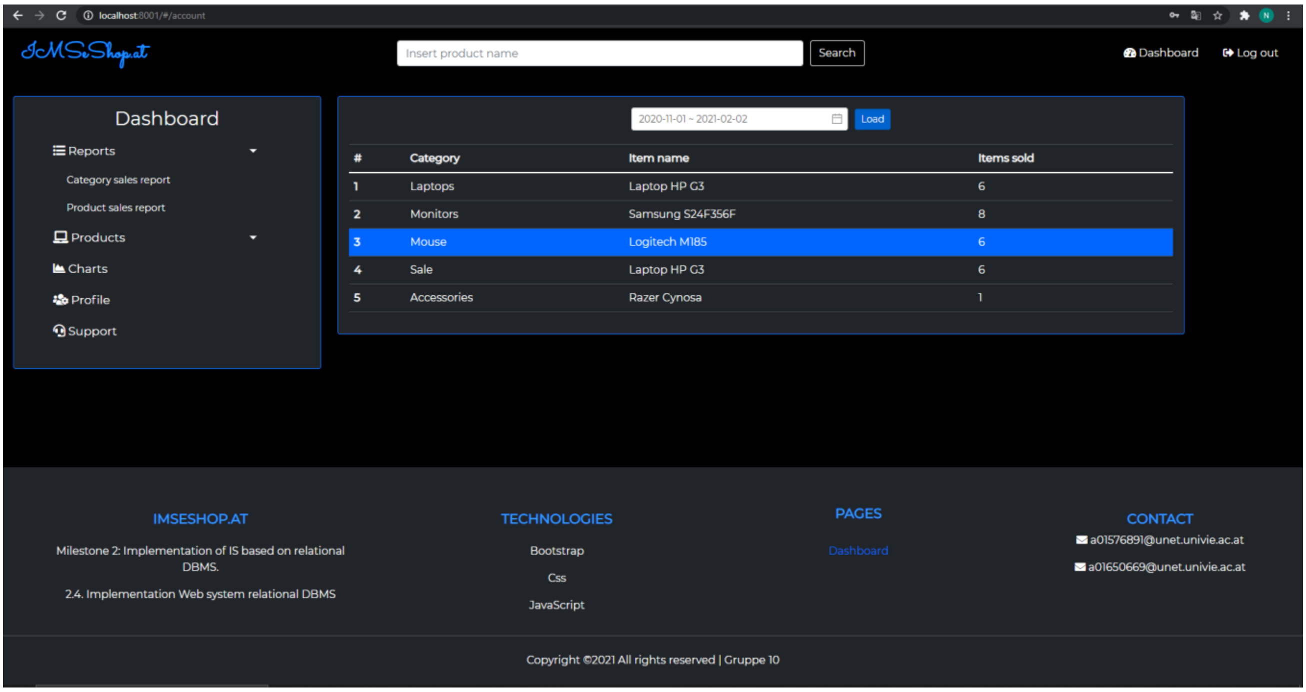
Dashboard

CONTACT

a01576891@unet.univie.ac.at

a01650669@unet.univie.ac.at

Copyright ©2021 All rights reserved | Gruppe 10



2020-11-01 ~ 2021-02-02

#	Category	Item name	Items sold
1	Laptops	Laptop HP G3	6
2	Monitors	Samsung S24F356F	8
3	Mouse	Logitech M185	6
4	Sale	Laptop HP G3	6
5	Accessories	Razer Cynosa	1

IMSESHOP.AT

Milestone 2: Implementation of IS based on relational DBMS.

2.4. Implementation Web system relational DBMS

TECHNOLOGIES

Bootstrap

Css

JavaScript

PAGES

Dashboard

CONTACT

a01576891@unet.univie.ac.at

a01650669@unet.univie.ac.at

Copyright ©2021 All rights reserved | Gruppe 10

3 How to

1. Clone the repo and navigate into the project.
2. run command: docker-compose up –build
3. navigate to: http://localhost:8001/
4. To test the main use cases log in with the credentials:
    - Email: mayo@gmail.com
    - Password: 123456
5. To test the elaborate use cases log in with the credentials:
    - Email: srdjan@gmail.com
    - Password: admin

*In rare the case, that the frontend is faster than the backend and on page start there are no visible items on the home page, just refresh once the page

4 Working protocol

Date, Time	Length in minutes	Task	Result	Responsibility	Remarks
26.12.2020, 10:00	360	Navbar component frontend	functional frontend component	Nemanja Srdanovic	…
27.12.2020 13:00	360	Carousel component frontend	functional frontend component	Nemanja Srdanovic	…
28.12.2020 13:15	360	ProductGrid and Footer component frontend	functional frontend component	Nemanja Srdanovic	…
03.12.2020 14:25	360	User Account and Dashboard components frontend	functional frontend component	Adem Mehremic	…
04.12.2020 11:00	360	Sign in and Sign out components frontend	functional frontend component	Nemanja Srdanovic	…
10.12.2020 11:20	360	Cart, LoggedIn components frontend	functional frontend component	Nemanja Srdanovic	…
11.12.2020 13:15	360	AdministrationController, BuyController,	functional backend implementation	Adem Mehremic, Nemanja Srdanovic	…
26.12.2020 16:15	400	Order, OrderItems, User, ItemController, LoginController,	functional backend implementation	Nemanja Srdanovic	…
27.12.2020	400	CartItems, Category, Employee, Item, Login	functional backend implementation	Adem Mehremic	
28.12.2020	400	Reposytoryes	functional backend implementation	Adem Mehremic	
03.12.2020	400	Services	functional backend implementation	Adem Mehremic	
04.12.2020	400	SalesReportDTO, CategoryReport	functional backend implementation	Adem Mehremic	
10.12.2020	400	Cart, CartDTO	functional backend implementation	Adem Mehremic	