

ДОДАТОК А

Графічний матеріал атестаційної роботи

ГЮИК. 502610.036

(позначення документу)

ЗАТВЕРДЖЕНО

ГЮИК. 502610.036 – ЛУ

Дослідження застосування алгоритмів колаборативної фільтрації в туристичній
установі

Графічний матеріал

ГЮИК. 502610.036 – ЛУ

АРКУШІВ 10

2020 р.

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ХАРКІВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ РАДІОЕЛЕКТРОНІКИ

«ЗАТВЕРДЖУЮ»

керівник атестаційної роботи

проф., д.т.н. Мінухін С. В.

Дослідження застосування алгоритмів колаборативної фільтрації в туристичній
установі

Графічний матеріал

ЛИСТ ЗАТВЕРДЖЕННЯ

ГЮИК. 502610.036 – ЛУ

РОЗРОБИВ:

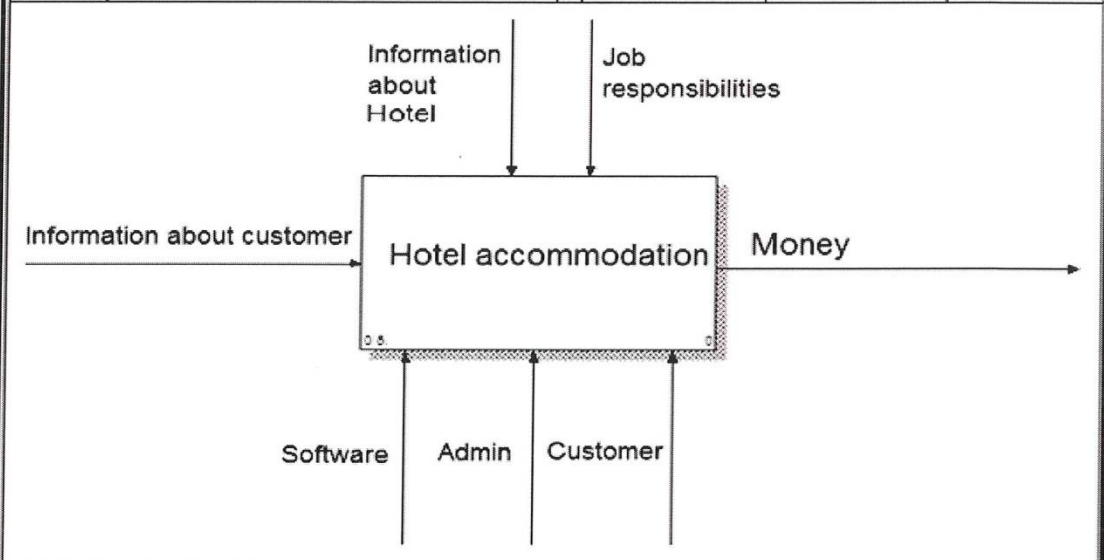
ст. гр. СПРм-18-2

Бугай Д.О.

2020 р.

КОНТЕКСТНА ДІАГРАМА

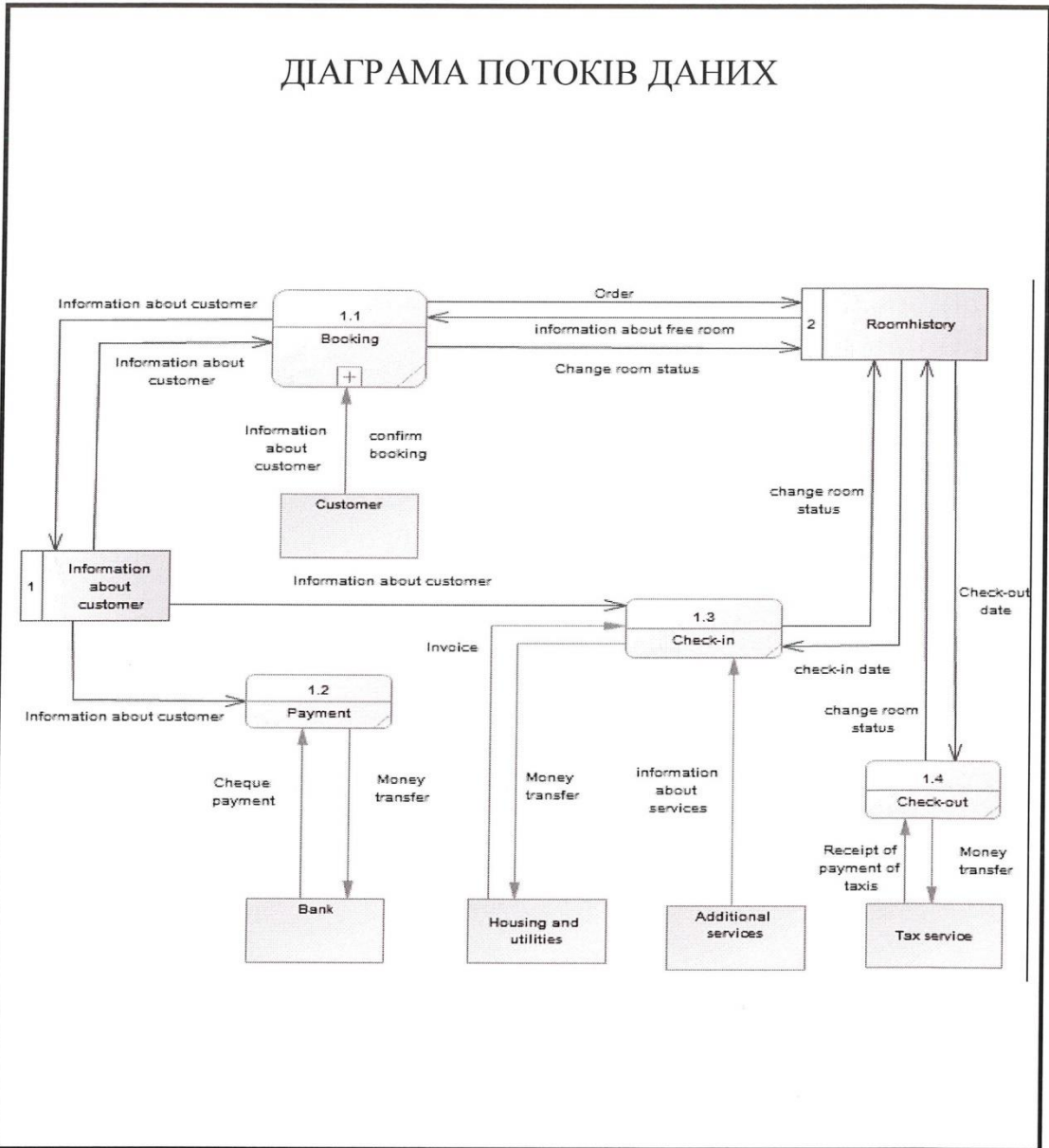
USED AT:	AUTHOR: DaniilBugay PROJECT: Hotel	DATE: 04.04.2016 REV: 28.05.2016	WORKING DRAFT RECOMMENDED PUBLICATION	READER	DATE	CONTEXT: TOP
NOTES: 1 2 3 4 5 6 7 8 9 10						



NODE: A-0	TITLE: Hotel accommodation	NUMBER:
--------------	-------------------------------	---------

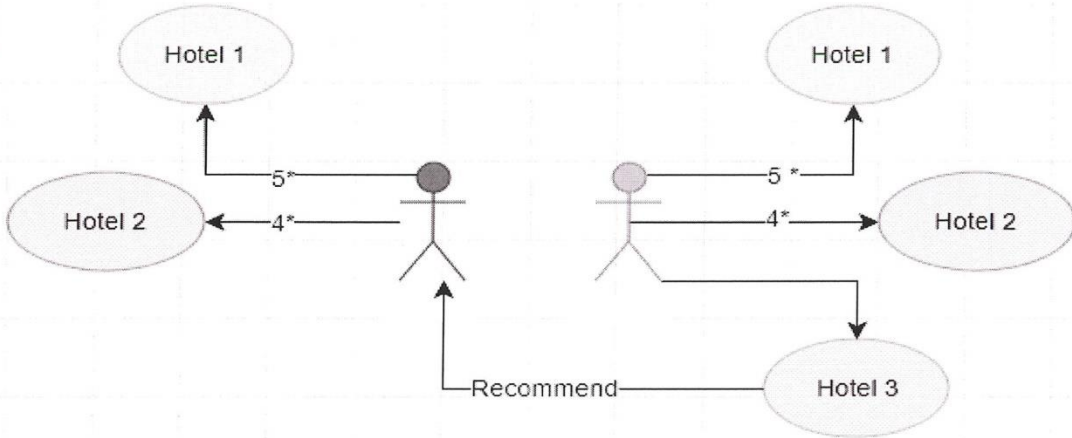
Розроб.	Бугай Д.О.			Контекстна діаграма
Перевір.	Мінухін С. В.	<i>См</i>	22.05	
Н. Контр.	Мінухін С. В.	<i>См</i>	22.05	
Затверд.	Гребеннік І.В.			СПРМ-18-2 СТ
				Лист 1 Листів 1

ДІАГРАМА ПОТОКІВ ДАНИХ



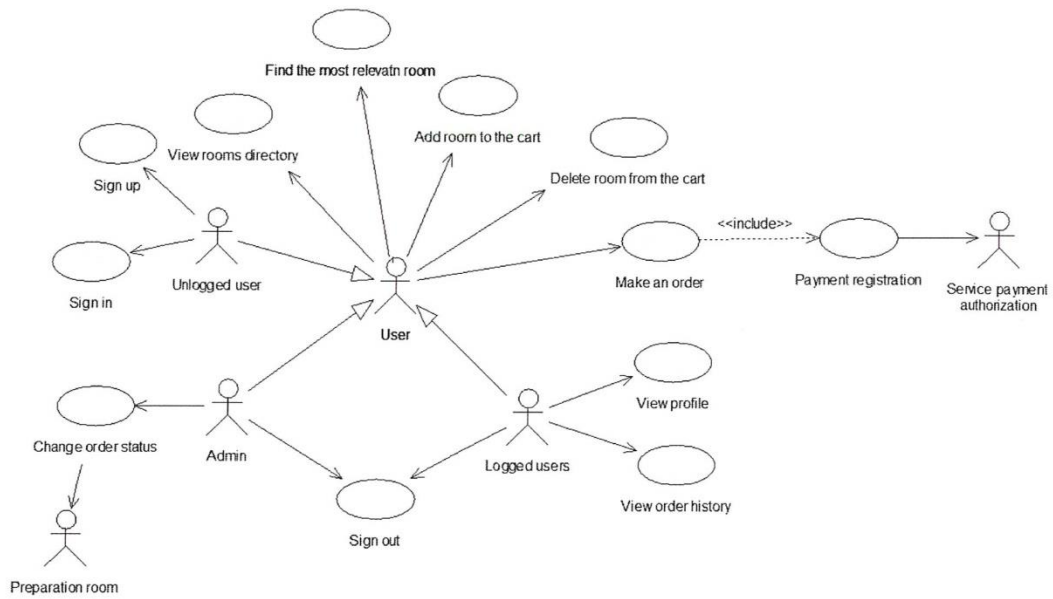
Розроб.	Бугай Д.О.			Діаграма потоків даних	
Перевір.	Мінухін С. В.	<i>СМ</i>	22.05		
Н. Контр.	Мінухін С. В.	<i>СМ</i>	22.05		
				СПРМ-18-2	Лист 1
Затверд.	Гребеннік І.В.			СТ	Листів 1

СХЕМА РЕКОМЕНДАЦІЙ ЗА КОЛЛАБОРАТИВНОЇ ФІЛЬТРАЦІЇ



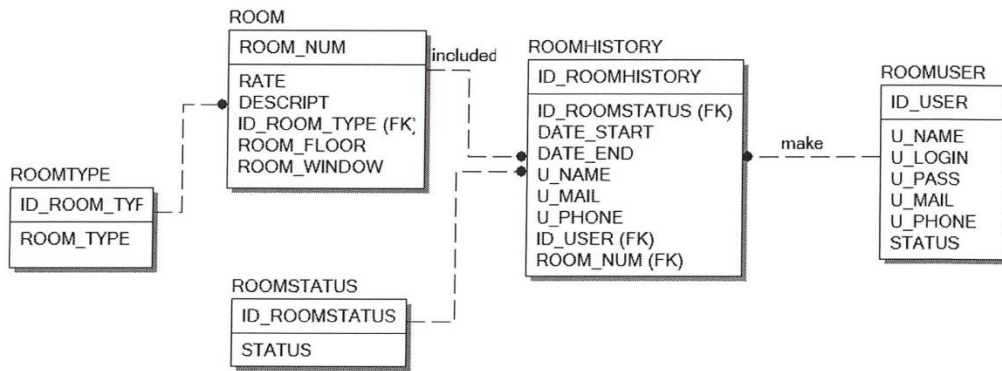
Розроб.	Бугай Д.О.			Схема рекомендацій за колаборативної фільтрації	
Перевір.	Мінухін С. В.	<i>ВМ</i>	22.05		
Н. Контр.	Мінухін С. В.	<i>ВМ</i>	22.05		
				СПРМ-18-2	Лист 1
Затверд.	Гребеннік І.В.			СТ	Листів 1

ДІАГРАМА ВАРІАНТІВ ВИКОРИСТАННЯ



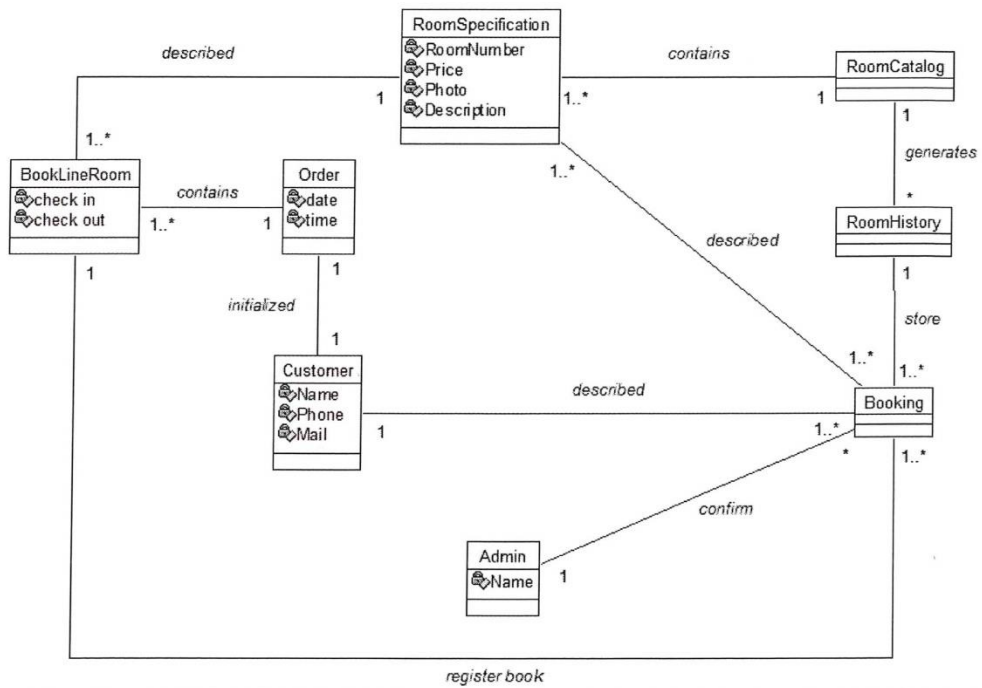
Розроб.	Бугай Д.О.			Діаграма варіантів використання	
Перевір.	Мінухін С. В.	<i>См</i>	22.05		
Н. Контр.	Мінухін С. В.	<i>См</i>	22.05		
				СПРМ-18-2	Лист 1
Затверд.	Гребеннік І.В.			СТ	Листів 1

МОДЕЛЬ ДАНИХ



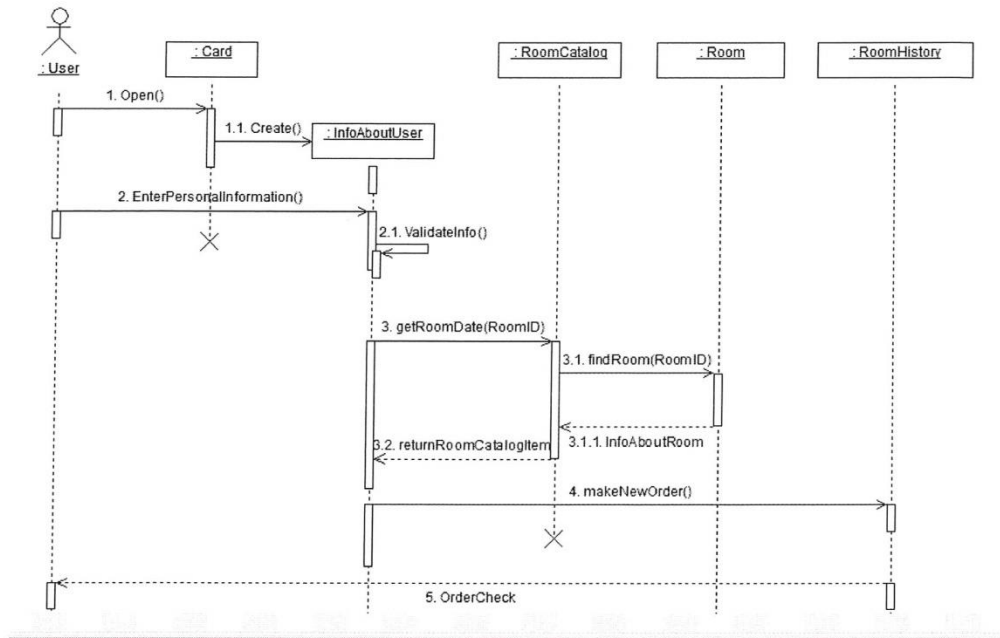
Розроб.	Бугай Д.О.			Модель даних	
Перевір.	Мінухін С. В.	<i>СМ</i>	22.05		
Н. Контр.	Мінухін С. В.	<i>СМ</i>	22.05		
				СПРм-18-2	Лист 1
Затверд.	Гребеннік І.В.			СТ	Листів 1

МОДЕЛЬ ПРЕДМЕТНОЇ ОБЛАСТІ



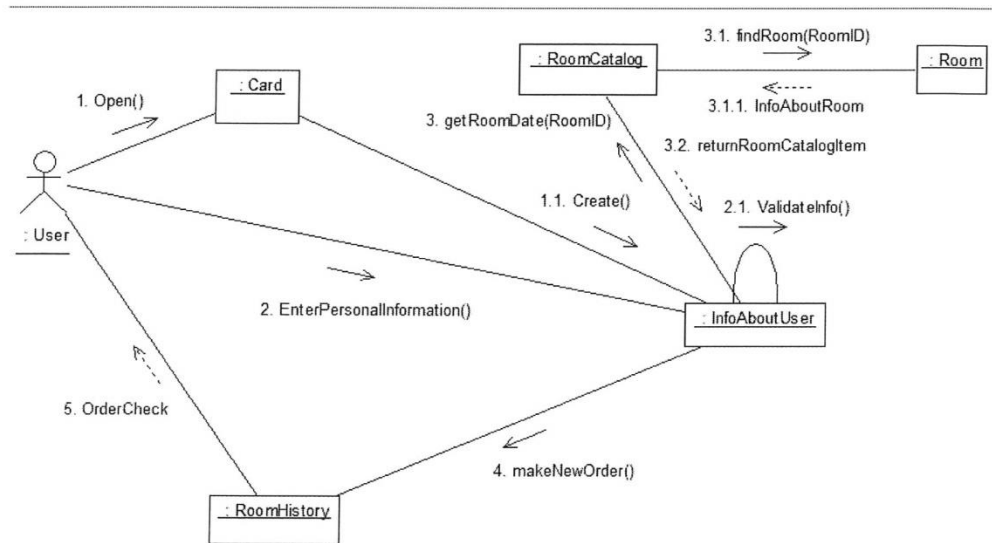
Розроб.	Бугай Д.О.			Модель ПО	
Перевір.	Мінухін С. В.	<i>СМ</i>	22.05		
Н. Контр.	Мінухін С. В.	<i>СМ</i>	22.05		
				СПРМ-18-2	Лист 1
Затверд.	Гребеннік І.В.			СТ	Листів 1

ДІАГРАМА ПОСЛІДОВНОСТІ



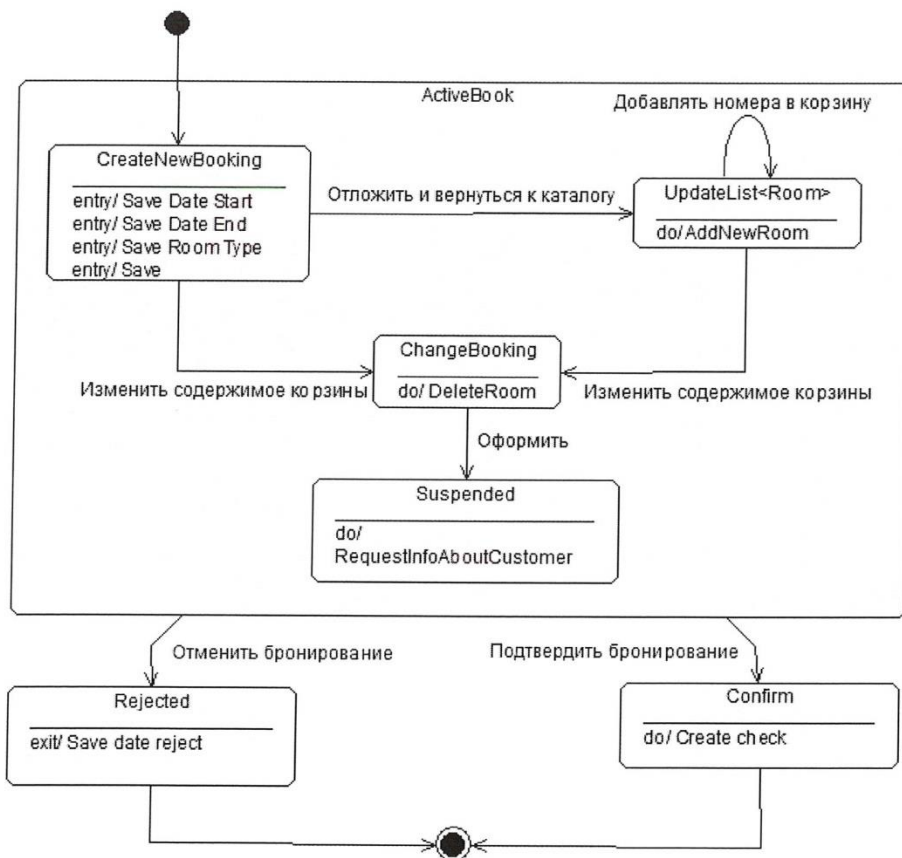
Розроб.	Бугай Д.О.			Діаграма послідовності	
Перевір.	Мінухін С. В.	<i>См</i>	22.05		
Н. Контр.	Мінухін С. В.	<i>См</i>	22.05		
Затверд.	Гребеннік І.В.			СПРМ-18-2 СТ	Лист 1 Листів 1

ДІАГРАМА КООПЕРАЦІЇ



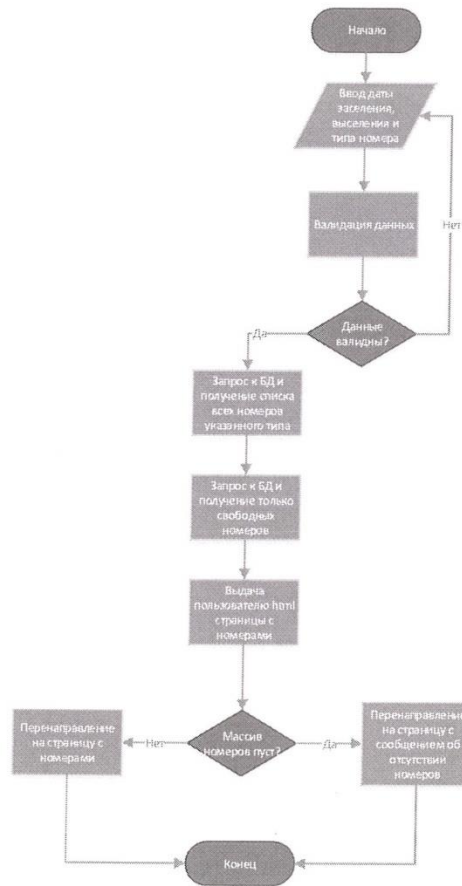
Розроб.	Бугай Д. О.			Діаграма кооперації	
Перевір.	Мінухін С. В.	<i>СВ</i>	22.05		
Н. Контр.	Мінухін С. В.	<i>СВ</i>	22.05		
				СПРМ-18-2	Лист 1
Затверд.	Гребеннік І. В.			СТ	Листів 1

ДІАГРАМА СТАНІВ



Розроб.	Бугай Д.О.			Діаграма станів	
Перевір.	Мінухін С. В.	<i>СМ</i>	22.05		
Н. Контр.	Мінухін С. В.	<i>СМ</i>	22.05		
				СПРМ-18-2	Лист 1
Затверд.	Гребеннік І.В.			СТ	Листів 1

АЛГОРИТМ РОБОТИ ПОШУКУ



Розроб.	Бугай Д.О.			Алгоритм роботи пошуку	
Перевір.	Мінухін С. В.	<i>СЦ</i>	22.05		
Н. Контр.	Мінухін С. В.	<i>СЦ</i>	22.05		
				СПРМ-18-2	Лист 1
Затверд.	Гребеннік І.В.			СТ	Листів 1

ДОДАТОК Б

Текст програми

ГЮИК. 502610.036 – 01 12 01

(позначення документу)

ЗАТВЕРДЖЕНО

ГЮИК. 502610.036 – 01 12 01 – ЛУ

Дослідження застосування алгоритмів колаборативної фільтрації в туристичній
установі

Текст програми

ГЮИК. 502610.036 – 01 12 01 – ЛУ

АРКУШІВ 8

2020 р.

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ХАРКІВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ РАДІОЕЛЕКТРОНІКИ

«ЗАТВЕРДЖУЮ»

керівник атестаційної роботи

проф., д.т.н. Мінухін С. В.

Дослідження застосування алгоритмів колаборативної фільтрації в туристичній
установі

Текст програми

ЛИСТ ЗАТВЕРДЖЕННЯ

ГЮИК. 502610.036 – 01 12 01 – ЛУ

РОЗРОБИВ:

ст. гр. СПРм-18-2

Бугай Д.О.

2020 р.

```

@Controller
public class AdminController {

    @Autowired
    OrderService orderService;

    @RequestMapping(value = "/hotel/newOrders", method = RequestMethod.GET)
    public String showOrders(ModelMap model) {
        List<Order> dataRoomHistory = orderService.loadRoomHistory(1);
        model.addAttribute("dataRoomHistory", dataRoomHistory);
        return "/AdminOrders";
    }

    @RequestMapping(value = "/hotel/changeStatusOrder", method = RequestMethod.POST)
    public String changeStatusOrder(@RequestParam(value = "bConfirm", required = false) String
bConfirm,
                                     @RequestParam(value = "bRejec", required = false) String bRejec,
                                     @RequestParam(value = "selectroom", required = false) String[] roomID) {
        Integer status = 1;
        if (bConfirm != null) {
            status = 2;
        }
        if (bRejec != null) {
            status = 3;
        }
        if (roomID != null) {
            List<Order> dataRoomHistory = orderService.loadRoomHistory(1);
            for (String row : roomID) {
                int currentRowrow = Integer.parseInt(row);
                Order order = dataRoomHistory.get(currentRowrow);
                Room room = new Room(order.getRoomNumber());
                orderService.updateStatusOrder(room, order.getDateStart(), order.getDateEnd(), status);
            }
        }
    }
}

```

```

        return "redirect:/hotel/newOrders";
    }
}

```

```

public enum UserRole {
    CLIENT, ADMIN;

    public static UserRole getRole(Roomuser user) {
        int roleId = Integer.valueOf(user.getStatus());
        return UserRole.values()[roleId];
    }
}

```

```

public String getName() {
    return name().toLowerCase();
}
}

```

```
@Controller
```

```
@SessionAttributes({"user", "userRole", "dataSelectRoom", "dataRoomCatalog", "Attr_DateStart", "Attr_DateEnd"})
```

```
public class OrderController {
```

```
    @Autowired
```

```
    RoomService roomService;
```

```
    @Autowired
```

```
    OrderService orderService;
```

```
    @RequestMapping(value = "/hotel", method = RequestMethod.GET)
```

```
    public String initHotel(Model model) {
```

```
        List<Roomtype> dataRoomType = roomService.loadRoomType();
```

```
        model.addAttribute("dataRoomType", dataRoomType);
```

```
        return "/Index";
```

```
    }
```

```

@RequestMapping(value = "/hotel/loadCard", method = RequestMethod.GET)
public String loadCard(ModelMap model, HttpSession session) {
    List dataSelectRoom = (List) session.getAttribute("dataSelectRoom");
    model.addAttribute("dataSelectRoom", dataSelectRoom);
    return "/Cart";
}

@RequestMapping(value = "/hotelCatalog", method = RequestMethod.GET)
public String userSignUp(@RequestParam(value = "ID_room_type") String ID,
    @RequestParam(value = "Date_start") String dateStart,
    @RequestParam(value = "Date_end") String dateEnd,
    ModelMap model, HttpSession session) {
    boolean startDate = isThisDateValid(dateStart, "dd.MM.yyyy");
    boolean endDate = isThisDateValid(dateEnd, "dd.MM.yyyy");
    if (startDate && endDate) {
        List<Room> dataRoomCatalog = roomService.loadRoomCatalog(ID, dateStart, dateEnd);
        Integer sizeList = dataRoomCatalog.size();
        int flagList = sizeList == 0 ? 0 : 1;
        session.setAttribute("dataRoomCatalog", dataRoomCatalog);
        session.setAttribute("Attr_DateStart", dateStart);
        session.setAttribute("Attr_DateEnd", dateEnd);
        model.addAttribute("aEMPTY_LIST", flagList);
        model.addAttribute("dataRoomCatalog", dataRoomCatalog);
        return "/Catalog";
    } else {
        return "redirect:/hotel";
    }
}

@ModelAttribute("dataSelectRoom")
public List<Room> roomInCard() {
    return new ArrayList<>();
}

```

```

@RequestMapping(value = "/addCard", method = RequestMethod.GET)
public String addCard(@RequestParam(value = "selectroom", required = false) String[] roomID,
HttpSession session) {
    if (roomID == null) {
        return "/Cart";
    } else {
        List dataSelectRoom = (List) session.getAttribute("dataSelectRoom");
        List dataRoomCatalog = (List) session.getAttribute("dataRoomCatalog");
        for (String currentRow : roomID) {
            int row = Integer.parseInt(currentRow);
            Room room = (Room) dataRoomCatalog.get(row);
            if (!dataSelectRoom.contains(room)) {
                dataSelectRoom.add(room);
            }
        }
        session.setAttribute("dataSelectRoom", dataSelectRoom);
        return "/Cart";
    }
}

```

```

@RequestMapping(value = "/createUser", method = RequestMethod.POST)
public String createUser(@RequestParam(value = "UserName") String userName,
        @RequestParam(value = "UserMail") String userMail,
        @RequestParam(value = "UserPhone") String userPhone,
        HttpSession session, HttpServletRequest request) {
    Roomuser user = new
Roomuser().setId(0).setUName(userName).setUMail(userMail).setUPhone(userPhone);
    session.setAttribute("user", user);
    request.setAttribute(View.RESPONSE_STATUS_ATTRIBUTE,
HttpStatus.TEMPORARY_REDIRECT);
    return "redirect:/makeOrder";
}

```

```

@RequestMapping(value = "/makeOrder", method = RequestMethod.POST)
public String makeOrder(@RequestParam(value = "bMakeOrder", required = false) String
makeOrder,
                        @RequestParam(value = "bDelete", required = false) String deleteOrder,
                        @RequestParam(value = "cart_select_room", required = false) String[] currentID,
                        HttpSession session) {
    Roomuser roomuser = (Roomuser) session.getAttribute("user");
    if (deleteOrder != null) {
        if (currentID != null) {
            List dataSelectRoom = (List) session.getAttribute("dataSelectRoom");
            for (String currentRow : currentID) {
                int row = Integer.parseInt(currentRow);
                Iterator room = dataSelectRoom.iterator();
                while (room.hasNext()) {
                    Room currentRoom = (Room) room.next();
                    if (currentRoom.getRoomNum() == row) {
                        room.remove();
                    }
                }
            }
            session.setAttribute("dataSelectRoom", dataSelectRoom);
        }
        return "/Cart";
    }
    if (makeOrder != null && roomuser == null) {
        return "/UserInfo";
    } else {
        int result = -1;
        String dateStart = (String) session.getAttribute("Attr_DateStart");
        String dateEnd = (String) session.getAttribute("Attr_DateEnd");
        List orderRooms = (List) session.getAttribute("dataSelectRoom");
        for (Object orderRoom : orderRooms) {
            Room room = (Room) orderRoom;
            result = orderService.roomHistoryInsert(room, dateStart, dateEnd, roomuser, 1);
        }
    }
}

```

```

    }
    if (result != -1) {
        return "/Order";
    } else {
        return "/Error";
    }
}
}
}

@Controller
@SessionAttributes({"user", "userRole", "dataSelectRoom", "dataRoomCatalog", "Attr_DateStart",
"Attr_DateEnd"})
public class UserController {

    @Autowired
    UserService userService;

    @Autowired
    OrderService orderService;

    @RequestMapping(value = "/hotel/login", method = RequestMethod.GET)
    public String userLogin(@RequestParam(value = "UserLogin") String userLogin,
        @RequestParam(value = "UserPassword") String userPassword,
        ModelMap model, HttpSession session) {
        List rooms = (List) session.getAttribute("dataSelectRoom");
        if (rooms != null) {
            rooms.clear();
            session.setAttribute("dataSelectRoom", rooms);
        }
        List<Roomuser> user = userService.login(userLogin, userPassword);
        if (user.size() != 0) {
            UserRole userRole = UserRole.getRole(user.get(0));

```

```

        session.setAttribute("user", user.get(0));
        session.setAttribute("userRole", userRole);
        return "redirect:/hotel";
    } else {
        model.addAttribute("ERROR_ID", 0);
        return "/Login";
    }
}

```

```

@RequestMapping(value = "/hotel/signUp", method = RequestMethod.POST)
public String userSignUp(@RequestParam(value = "UserName") String userName,
                        @RequestParam(value = "UserLogin") String userLogin,
                        @RequestParam(value = "UserPassword") String userPassword,
                        @RequestParam(value = "UserMail") String userMail,
                        @RequestParam(value = "UserPhone") String userPhone,
                        ModelMap model) {

    int result = userService.regUser(new
Roomuser().setULogin(userLogin).setUPass(userPassword).setUName(userName)
        .setUMail(userMail).setUPhone(userPhone).setStatus("0"));
    if (result != -1) {
        return "/Login";
    } else {
        model.addAttribute("ERROR_ID", 0);
        return "/SignUp";
    }
}

```

```

@RequestMapping(value = "/hotel/logOut", method = RequestMethod.GET)
public String logOut(SessionStatus sessionStatus) {
    sessionStatus.setComplete();
    return "redirect:/hotel";
}

```

```

@RequestMapping(value = "/hotel/home", method = RequestMethod.GET)

```

```

public String showHomePage(HttpSession session, ModelMap model) {
    Roomuser user = (Roomuser) session.getAttribute("user");
    List<Order> listRoomHistory = orderService.loadRommHistory(user);
    model.addAttribute("list_roomHistory", listRoomHistory);
    return "/Home";
}

```

@Repository

```

public class MySQLOrderDAO implements OrderDAO {

```

```

    @PersistenceContext

```

```

    private EntityManager em;

```

```

    @Override

```

```

    public int roomHistoryInsert(Room room, String dateStart, String dateEnd, Roomuser user, Integer
roomStatus) {

```

```

        Date dStart = stringToDate(dateStart, "dd.MM.yyyy");

```

```

        Date dEnd = stringToDate(dateEnd, "dd.MM.yyyy");

```

```

        Roomstatus roomstatus = new Roomstatus(roomStatus);

```

```

        Roomhistory order = new Roomhistory().setRoom(room).setDateStart(dStart).setDateEnd(dEnd)

```

```

            .setIdRoomuser(user.getId()).setUName(user.getUName()).setUMail(user.getUMail())

```

```

            .setUPhone(user.getUPhone()).setIdRoomstatus(roomstatus);

```

```

        em.persist(order);

```

```

        return order.getId();

```

```

    }

```

ДОДАТОК В

«Відомість атестаційної роботи»

ГЮИК. 502610.036 ДЗ

(позначення документу)

