



ФГБОУ ВО Уральский государственный горный
университет

Инженерно-экономический факультет
Кафедра информатики

Курсовой проект

По дисциплине «Технологии программирования»

На тему «Разработка приложения для
"Формирование статистики по оценкам в
заданиях"»

Выполнил:
Студентка гр. ИНФ-20-4
Грицюк Н.В.

Проверила:
ст. преп. каф.
информатики
Волкова Е.А.

Екатеринбург, 2023г

Оглавление

1. Постановка задачи	3
1.1 Характеристика объекта	3
1.2 Потенциал внедрения	3
1.3 Краткое техническое задание.....	3
1.3.1 Назначение и цели создания системы.....	3
1.3.1.1 Назначение системы	3
1.3.1.2 Цели создания системы	3
1.3.2 Требования к системе	3
1.3.2.1 Требования к системе в целом.....	3
1.3.2.1.1 Требования к структуре и функционированию системы.....	3
2. Проектирование системы.....	4
2.1 Моделирование системы.....	5
2.2 Проектирование БД.....	6
2.3 Проектирование интерфейса	7
3. Разработка системы	8
3.1 Выбор средств реализации	8
3.2 Этапы реализации	8
3.3 Реализация	8
3.3.1 Листинг и описание классов	8
3.3.2 Интерфейс приложения	11

1. Постановка задачи

Целью этого курсового проекта является создание WEB-приложения для формирования статистики по оценкам в заданиях.

Приложение должно отвечать требованиям к многоуровневой структуре (т.е. иметь как минимум уровень БД, бизнес-логики и, собственно, интерфейс), а также обязательно иметь возможности разграничения доступа (администратор, группы пользователей).

1.1 Характеристика объекта

Объектом информатизации является WEB-приложение, которое обрабатывает БД.

1.2 Потенциал внедрения

Данное приложение разрабатывается для студентов и преподавателей, которые хотят сэкономить время на создании статистики по общим баллам в заданиях.

В дальнейшем данное приложение будет очень полезно высшим и средним учебным заведениям. Так же его можно будет применить для других различных школ и заведений. Для этого можно быстро разработать бизнес-логику для данных задач.

1.3 Краткое техническое задание

Согласно ГОСТ 34.602-89

1.3.1 Назначение и цели создания системы

1.3.1.1 Назначение системы

Данное приложение предназначена для формирования статистики и предоставления её пользователю.

1.3.1.2 Цели создания системы

- обеспечение сбора и первичную обработку исходной информации, необходимой для пользователей.

- предоставление пользователям удобной платформы для формирования статистики с оценками.

1.3.2 Требования к системе

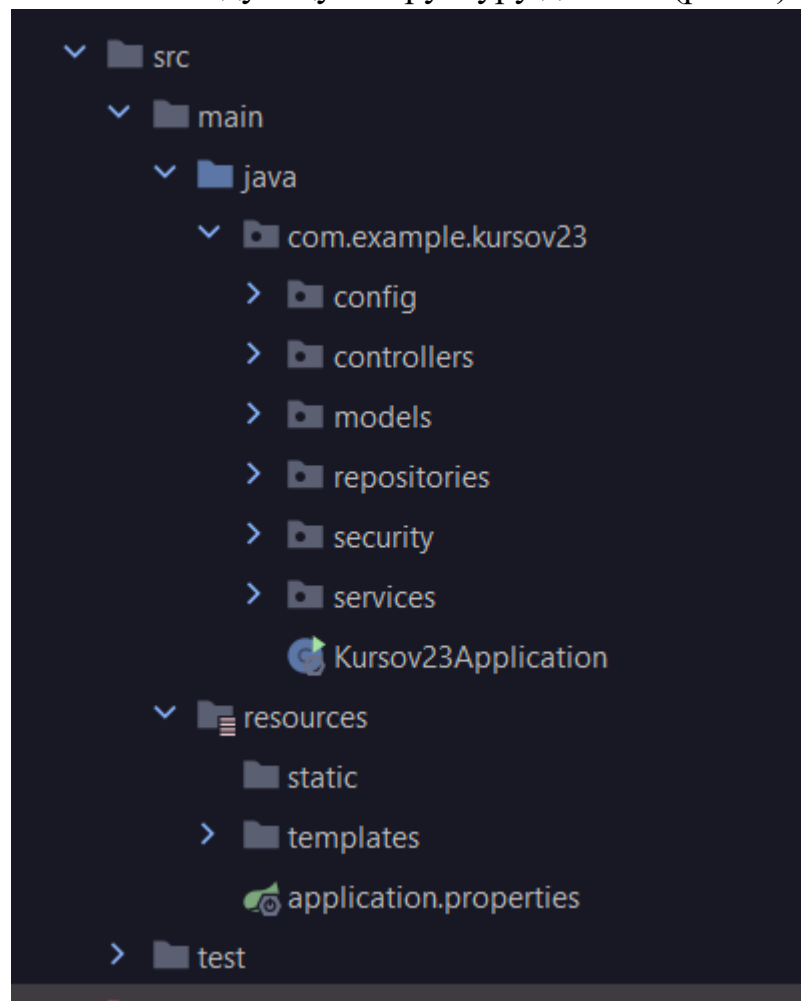
1.3.2.1 Требования к системе в целом

1.3.2.1.1 Требования к структуре и функционированию системы

Источниками данных для Системы должны быть СУБД MY SQL.

2. Проектирование системы

Данный проект имеет следующую структуру данных (рис. 1)



(рис.1)

Папка config содержит в себе служебные классы, такие как разграничение доступа и доступ данным.

В папке controllers написаны классы, предназначенные для непосредственной обработки запросов от клиента и возвращения результатов.

Models содержит в себе описание сущностей БД и их связи.

В папке repositories содержатся интерфейсы обработки данных.

Папка service содержит в себе классы с логикой обработки данных.

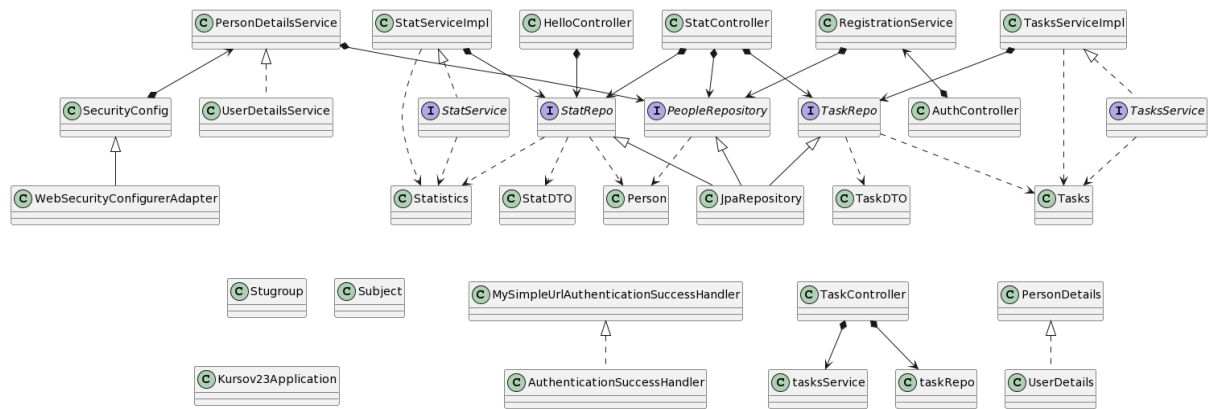
В папке security содержатся классы для работы с spring security.

Папка services содержит в себе классы для работы с БД.

В resoureces находятся шаблоны и статические элементы, которые предназначены для визуализации данных.

2.1 Моделирование системы

UML диаграмма показана ниже (рис. 2)

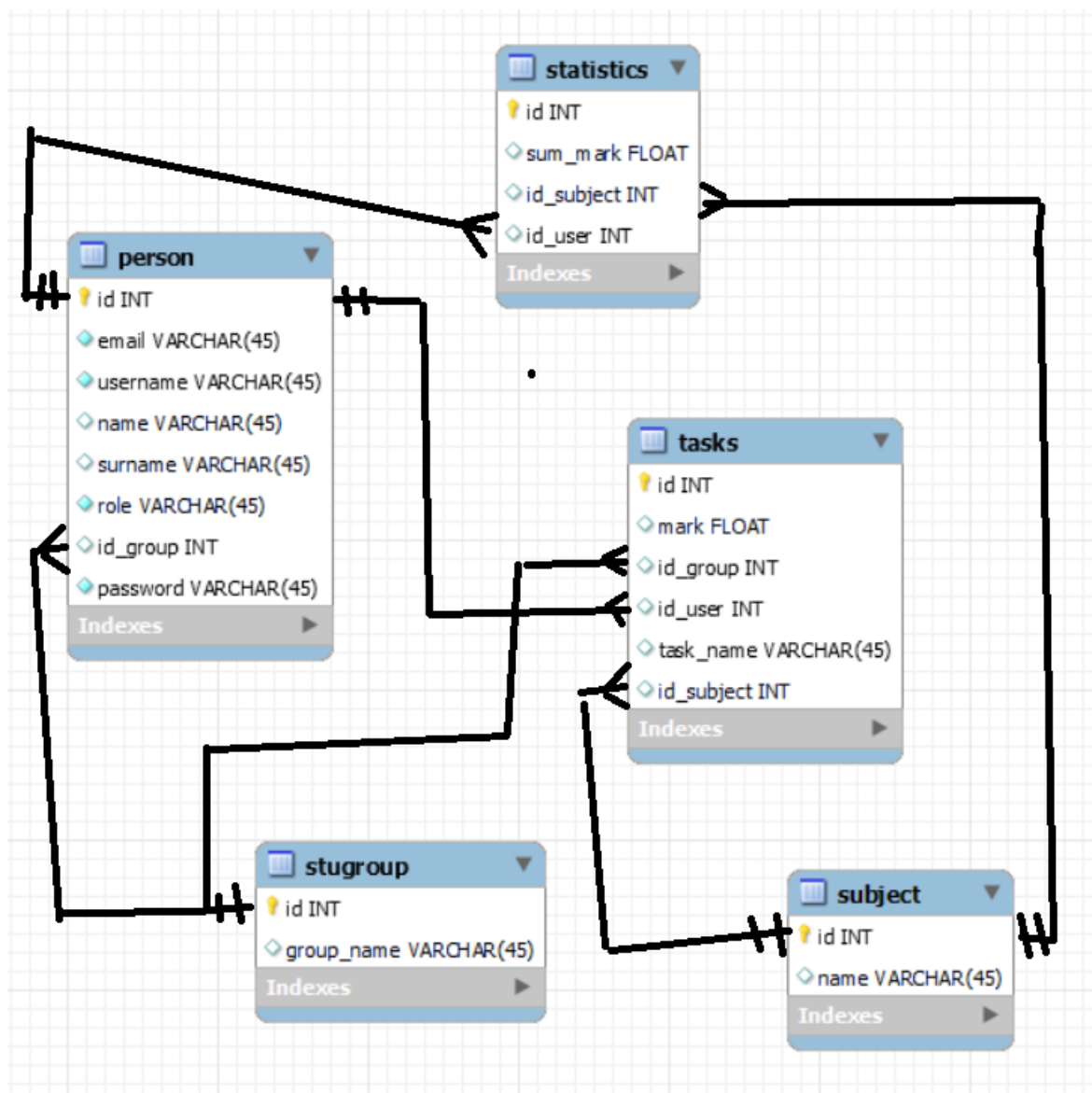


(рис. 2)

2.2 Проектирование БД

В качестве сервера баз был выбран MYSQL версии 8.0.

ER-диаграмма, описывающая структуру БД представлена на (рис. 5)



(рис. 5)

2.3 Проектирование интерфейса

Интерфейс приложения должен содержать навигационную панель. В основной части должен содержаться листинг той или иной информации. В качестве оформления интерфейса будет использован Bootstrap.

3. Разработка системы

3.1 Выбор средств реализации

В качестве языка программирования был выбран java, использующее Spring-boot. Такое решение было выбрано, так как на этом языке наиболее удобнее делать web-приложение, также код выполняется достаточно быстро и занимает мало места.

Базой данных будет служить бесплатный сервер баз данных MySQL.

Для оформления HTML страниц и скриптов был выбран Bootstrap, потому что им удобно пользоваться и не нужны дополнительные настройки CSS.

3.2 Этапы реализации

Реализация была разделена на несколько этапов:

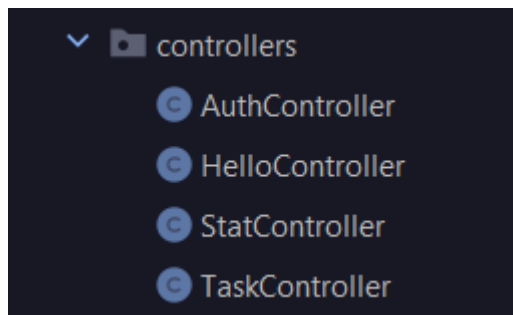
- Проектирование структуры БД
- Добавление зависимостей для работы со Spring
- Разработка листинга создания заданий
- Разработка логики добавления оценок и редактирования/удаления заданий
- Разработка листинга статистики
- Разработка логики для формирования статистики
- Реализация разграничений доступа
- Оформление интерфейса
- Разработка логики создания заданий

Проект сейчас находится на начале последнего этапа. Изначальный план по разработке приложения «Формирование статистики по оценкам» был выполнен. Поэтому последний этап находится ещё в разработке. Также данный проект можно дополнять ещё многими функционалами, например, чат между пользователями, просмотр решения в заданиях и другое.

3.3 Реализация

3.3.1 Листинг и описание классов

Общая структура приложения была приведена на (рис.1).
В данный момент приложение имеет 4 контроллера (рис. 6)



(рис.6)

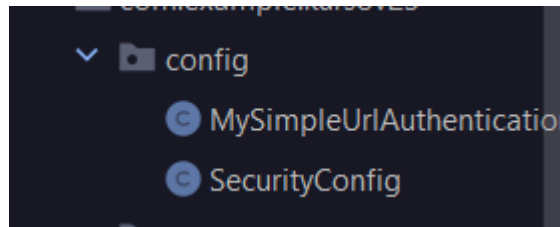
- AuthController отвечает за все действия связанные с заходом на главную страницу.
- HelloController отвечает за все действия связанные с аутентификацией.
- StatController отвечает за все действия связанные с просмотром и формированием статистики.
- TaskController отвечает за простое создание и взаимодействие с заданиями.

В листинге 1 показан метод формирования статистики.

Листинг 1

```
@GetMapping("/showStatistics")
public String showStatistics(Model model) {
    List<Tasks> tasks = taskRepo.finTasksByGrSub(id_gr, id_su);
    List<Person> person = peopleRepository.findAllUserByGroup(id_gr);
    System.out.println(tasks.size());
    System.out.println(person.size());
    if (!person.isEmpty()){
        for (int i = 0; i < person.size(); i++){
            float count = 0.0f;
            for (int j = 0 ; j < tasks.size(); j++){
                if ((tasks.get(j).getId_user() == person.get(i).getId())
                    && (tasks.get(j).getId_group() ==
person.get(i).getGroup())){
                    count = count + tasks.get(j).getMark();
                }
            }
            Statistics st = statRepo.finStatBySuUs(person.get(i).getId(),
id_su);
            if (st != null){
                st.setSum_mark(count);
                statRepo.save(st);
            } else {
                Statistics newStat = new Statistics();
                newStat.setSum_mark(count);
                newStat.setId_subject(id_su);
                newStat.setId_user(person.get(i).getId());
                statRepo.save(newStat);
            }
        }
    }
    model.addAttribute("statList1", statRepo.finStatBySuGr(id_su,
id_gr));
    return "showStatistics";
}
```

В папке config содержится 2 класса (рис. 7)



(рис. 7)

- MySimpleUrlAuthenticationSuccessHandler служит для перенаправления пользователя на ту или иную страницу после входа в систему (это зависит от роли).
- SecurityConfig служит за разграничения доступа пользователей.

В листинге 2 представлен класс WebSecurityConfig

Листинг 2

```
package com.example.kursov23.config;

@EnableWebSecurity
public class SecurityConfig extends WebSecurityConfigurerAdapter {

    private final PersonDetailsService personDetailsService;

    @Autowired
    public SecurityConfig(PersonDetailsService personDetailsService) {
        this.personDetailsService = personDetailsService;
    }

    @Override
    protected void configure(HttpSecurity http) throws Exception {
        http.csrf().disable()
            .authorizeRequests()
            .antMatchers("/admin/**").hasRole("ADMIN")
            .antMatchers("/login", "/error").permitAll()
            .anyRequest().hasAnyRole("USER", "ADMIN")
            .and()
            .formLogin().loginPage("/login")
            .loginProcessingUrl("/process_login")
            .successHandler(myAuthenticationSuccessHandler())
            .failureUrl("/login?error")
            .and()
            .logout()
            .logoutUrl("/logout")
            .logoutSuccessUrl("/login");
    }

    // Настраивает аутентификацию
    protected void configure(AuthenticationManagerBuilder auth) throws
    Exception {
        auth.userDetailsService(personDetailsService);
    }

    @Bean
    public AuthenticationSuccessHandler myAuthenticationSuccessHandler() {
        return new MySimpleUrlAuthenticationSuccessHandler();
    }
}
```

```

@Bean
public PasswordEncoder getPasswordEncoder() {
    return NoOpPasswordEncoder.getInstance();
}
}

```

На данный момент в папке service находится только 6 классов:

- PersonDetailsService, который нужен для работы с пользователем.
- RegistrationService, отвечающий за регистрацию.
- StatService для обращения к бд.
- StatServiceImpl, который нужен для работы с статистикой.
- TasksService для обращения к бд.
- TasksServiceImpl, который нужен для работы с заданиями

В листинге 3 показан метод добавления пользователя.

Листинг 3

```

@Service
public class RegistrationService {

    private final PeopleRepository peopleRepository;

    @Autowired
    public RegistrationService(PeopleRepository peopleRepository) {
        this.peopleRepository = peopleRepository;
    }

    @Transactional
    public void register(Person person) {
        person.setPassword(person.getPassword());
        person.setRole("ROLE_ADMIN");
        peopleRepository.save(person);
    }
}

```

3.3.2 Интерфейс приложения

На (рис. 8) показана страница со статистикой.

<div>Log Out</div>				
Фамилия	Имя	Название группы	Название предмета	Общая оценка
user	user	ИНФ-20-2	ТРПО	14.0

(рис. 8)

Как можно видеть мы зашли за пользователя с именем user и просматривать существующую статистику по предметам.

Как видим на (рис. 9) у пользователя Admin куда больше возможностей. Он имеет право создавать и редактировать задания, а так же просматривать интересующую статистику. Теперь нажмём на кнопку редактировать и отредактируем эту опечатку.



(рис.9)

Для редактирования и добавления заданий используется такая форма ввода (рис. 10, 11). После нажатия кнопки «Сохранить задание» применяются все изменения, которые были записаны в форме.

Название задания

Айди предмета

Айди группы

Айди студента

Сохранить Задание

Назад

(рис. 10)

Название задания

Айди предмета

Айди группы

Айди студента

Оценка

Сохранить Задание

Назад

(рис. 11)

