

МИНОБРНАУКИ РОССИИ

Федеральное государственное бюджетное образовательное учреждение  
высшего образования

**«САРАТОВСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ  
ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ИМЕНИ Н.Г.ЧЕРНЫШЕВСКОГО»**

Кафедра математического и компьютерного моделирования

**Проектирование и реализация ИС «Управление бизнес-проектами»**

АВТОРЕФЕРАТ БАКАЛАВРСКОЙ РАБОТЫ

студента 4 курса 441 группы

направление 09.03.03 — Прикладная информатика

механико-математического факультета

Степаненко Алексея Павловича

Научный руководитель  
доцент, к.э.н.

Ю. В. Мельникова

Зав. кафедрой  
зав. каф., д.ф.-м.н., доцент

Ю.А. Блинков

Саратов 2019

**Введение.** На сегодняшний день существует большое количество технологий, облегчающих разработку и объединение программных компонентов. Одними из таких технологий являются Spring Framework и Hibernate, используемые для разработки приложений, написанных на языке программирования Java.

Управление задачами – универсальная область деятельности, не ограниченная конкретной сферой. Данная область присутствует в работе организаций любого масштаба, независимо от направления деятельности последних. Управление деятельностью организации и делегирование задач занимает существенное время, огромное количество которого можно сэкономить, используя систему управления задачами. Особенно актуально это в организациях крупного масштаба.

Целью данной выпускной квалификационной работы бакалавра является знакомство с востребованными технологиями, используемыми при разработке приложений, такими как Spring и Hibernate, а также реализация актуальной в настоящее время информационной системы менеджмента задач.

Работа включает в себя 3 раздела. В первом приведено описание предметной области и видов информационных систем. Второй раздел содержит описание технологий, используемых в работе. В третьем разделе описан процесс разработки информационной системы.

## **Описание предметной области**

**Информационные системы** Информационные системы – это системы, предназначенные для хранения, поиска и обработки информации, то есть для удовлетворения конкретных информационных потребностей в рамках определенной предметной области.

Существуют различные классификации систем, каждая из которых предназначена для удовлетворения информационных потребностей в рамках конкретной предметной области. Система, разрабатываемая в данной работе, принадлежит типу ERP-систем планирования (управления) ресурсами предприятия. Прежде всего, под таким планированием подразумевается обработка критически важных для работы данных.

Критически важные данные – это данные, без которых работа предприятия невозможна. Сюда могут быть отнесены данные о работе различных отделов организации по разным показателям, включая сведения о состоянии различных задач.

**Использование информационной системы управления задачами**  
Менеджер задач (или «Task Manager») – это инструмент для управления проектами. Существует огромное количество реализаций таких систем, которые обладают своими особенностями, сходствами и различиями с другими системами.

Использование менеджера задач упрощает и организует рабочий процесс, позволяет в режиме реального времени просматривать информацию о состоянии задач и занятости сотрудников. Дополнительным преимуществом является возможность сбора статистики.

Одновременно с этим, такая система упрощает процесс коммуникации между сотрудниками, что позволяет экономить большое количество полезного времени.

### **Технологии, используемые в работе**

**Spring Framework** Фреймворк – программная платформа, определяющая структуру программной системы; программное обеспечение, облегчающее разработку и объединение разных компонентов большого программного проекта.

Spring Framework – фреймворк с открытым исходным кодом для java-платформы, который позволяет разрабатывать современные Enterprise-приложений (промышленные, корпоративные приложения) на языке программирования Java.

Spring фокусируется на реализации внутренних особенностей корпоративных приложений, благодаря чему разработчики могут сосредоточиться на бизнес-логике приложения, не занимаясь разработкой функций, которые предоставляет фреймворк (таких как функция авторизации пользователей и другие).

**Hibernate** Hibernate – это библиотека с открытым исходным кодом для платформы Java, предназначенная для решения задач ORM (Object/Relational Mapping, объектно-реляционное отображение).

ORM – технология программирования, которая связывает базы данных с концепциями объектно-ориентированных языков программирования.

Hibernate имеет ряд преимуществ перед классическим способом подключения проекта к базе данных, удаляет повторяющийся код обращений к базе, поддерживает некоторые концепции объектно-ориентированного программирования (ассоциации, наследование, коллекции).

## **СУБД MySQL**

База данных – это организованная структура, предназначенная для хранения, изменения и обработки взаимосвязанной информации.

Стоит отметить, что существуют различные модели хранения информации, различные типы баз данных. Однако, самой популярной моделью данных является реляционная, представляющая хранимую информацию в виде набора двумерных таблиц.

Для автоматизированной работы с базами данных существуют, так называемые, системы управления базами данных (СУБД). СУБД – совокупность программных и лингвистических средств общего или специального назначения, обеспечивающих управление созданием и использованием баз данных.

Одной из таких систем является MySQL, используемая в данной работе. MySQL – это свободная реляционная СУБД. Это достаточно быстрая и надежная система, предназначенная для использования в приложениях малого и среднего масштабов.

## **Разработка информационной системы**

**Постановка задачи и анализ требований** Необходимо определить конкретные требования к системе. В данной системе, в первую очередь, в системе должна быть реализована логичная структура базы данных, для хранения служебной, организационной и вспомогательной информации. Важнейшими функциями, которые должны быть реализованы в системе, являются:

- Создание задач;
- назначение задач;

- редактирование задач;
- завершение задач.

Однако, зачастую к подобным системам предъявляются дополнительные требования, которые направлены на повышение удобства и эффективность рабочего процесса:

- Хранение информации о сотрудниках;
- создание контактов сотрудников;
- ведение переписки в системе;
- хранение информации о должностях.

Для обеспечения возможности использования системы всеми сотрудниками, независимо от их технических навыков, необходимо реализовать графический веб-интерфейс.

**Процесс разработки** Разработка данной информационной системы начинается с создания моделей сущностей, используемых в системе. В процессе выполнения данной задачи созданы модели соответствующие таблицам сущностей базы данных.

Для того, чтобы обеспечить взаимодействие используемых фреймворков с моделями сущностей, необходимо соблюдать определенные правила:

- Класс-сущность должен быть помечен аннотацией «@Entity» чтобы класс был доступен для взаимодействия с базой данных через фреймворк Hibernate. Также необходимо указать параметр name, который связывает сущность с таблицей базы данных;
- необходимо создать поля данных, имеющие такие же имена, как и поля в соответствующей таблице базы данных. При этом, поле-идентификатор необходимо пометить аннотацией «@Id», чтобы сделать его первичным ключом;
- необходимо создать стандартный пустой конструктор и конструктор с нужными параметрами, чтобы обеспечить инициализацию сущности как объекта в программе, при считывании из базы или при создании соответственно;
- необходимо создать для каждого поля данных методы для получения и установки значений полей (getter и setter).

В результате соблюдения данных принципов на примере модели «Сотрудник» реализован следующий код:

```
1  @Entity(name = "EMPLOYEES")
2  public class Employee {
3
4      @Id
5      @GeneratedValue(strategy = GenerationType.IDENTITY)
6      private int employee_id;
7
8      private String first_name;
9      private String last_name;
10     private String password;
11     private String email;
12     private String phone;
13     private int location_id;
14     private int employee_enabled;
15
16     public Employee() {
17     }
18
19     public Employee(int employee_id,
20                    String first_name,
21                    String last_name,
22                    String password,
23                    String email,
24                    String phone,
25                    int location_id,
26                    int employee_enabled) {
27         this.employee_id = employee_id;
28         this.first_name = first_name;
29         this.last_name = last_name;
30         this.password = password;
31         this.email = email;
32         this.phone = phone;
33         this.location_id = location_id;
34         this.employee_enabled = employee_enabled;
35     }
36
37     public int getEmployee_id() {
38         return employee_id;
39     }
40
41     public void setEmployee_id(int employee_id) {
42         this.employee_id = employee_id;
43     }
44
```

```
45 | // остальные getters and setters
46 |
47 | }
```

Аннотация «@GeneratedValue(strategy = GenerationType.IDENTITY)» означает, что идентификатор сущности генерируется автоматически и его не нужно указывать.

Аналогичным образом разработаны остальные модели сущностей, представленных таблицей, приведенной в соответствии с рисунком ??.

**Разработка веб-интерфейса** Информационная система, предназначенная для использования большим количеством людей должна обладать простым и интуитивно понятным интерфейсом. Правильно реализованный графический интерфейс не только эстетичен, но и экономит большое количество времени, поскольку обеспечивает более быструю реакцию пользователей на элементы управления.

Для разработки графического интерфейса веб-страниц в рамках данного учебного проекта выбрана достаточно распространенная технология JSP.

JSP (Java Server Pages) – технология, позволяющая создавать содержимое, которое имеет как статические, так и динамические компоненты. Страницы JSP могут содержать текст двух типов:

- Статические исходные данные, которые могут быть оформлены в одном из текстовых форматов: HTML, SVG, WML, или XML;
- JSP-элементы, которые конструируют динамическое содержимое [19].

Данный способ описания содержимого веб-страниц обладает большой популярностью, поскольку исходный код страниц не хранит конкретных данных а работает со всеми данными, имеющимися в системе.

Для описания веб-страницы необходимо создать документ формата jsp и реализовать код, который ссылается на необходимую часть информационной системы. Далее приводится пример реализации страницы входа в систему:

```
1 | <!DOCTYPE html>
2 | <head>
3 |     // служебные настройки и подключение стилей оформления
4 | </head>
5 | <body>
6 |
7 | <div class="login-page">
```

```

8     <div class="form">
9         <form method="post" class="login-form" action="/login">
10            <input name="username" type="text" class="form-control"
11                placeholder="Username"
12                autofocus="true" />
13            <input name="password" type="password" class="form-control"
14                placeholder="Password" />
15            <button type="submit">login</button>
16            <p class="message">Not registered?
17                <a href="/registration">Create an account</a></p>
18        </form>
19    </div>
20</div>
21
22</body>
23</html>

```

На странице создается форма класса login-form, что означает, что в ней вводятся логин и пароль пользователя, которые затем отправляются в систему для проверки. В форме указан параметр action=«/login» – это стандартный адрес, по которому отправляются данные для входа в систему. После нажатия на кнопку login, система проводит проверку введенных данных и, если они соответствуют одному из существующих пользователей, происходит аутентификация.

### **Использование системы**

На стартовой странице присутствует форма аутентификации пользователя. Данная форма позволяет уже зарегистрированному сотруднику войти в систему. Если сотрудник еще не прошел регистрацию, ему потребуется перейти по ссылке «Create an account». Страница аутентификации представлена в соответствии с рисунком 1.



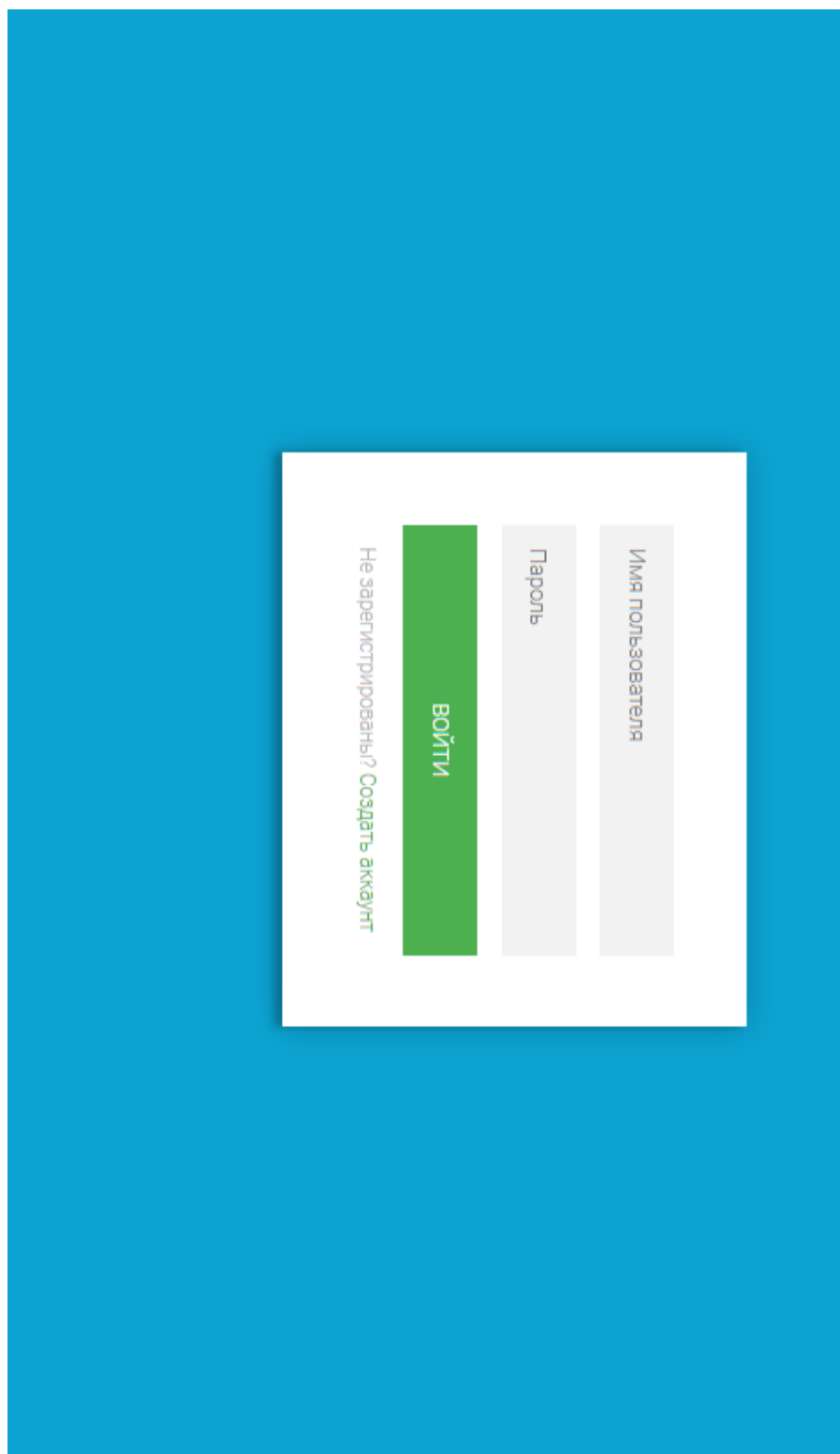


Рисунок 1 – Стартовая страница информационной системы менеджера задач

После регистрации или аутентификации открывается главная страница информационной системы – страница списка задач. На данной странице пред-

ставлены все задачи, которые назначены текущему сотруднику. Данная страница приведена в соответствии с рисунком 2.

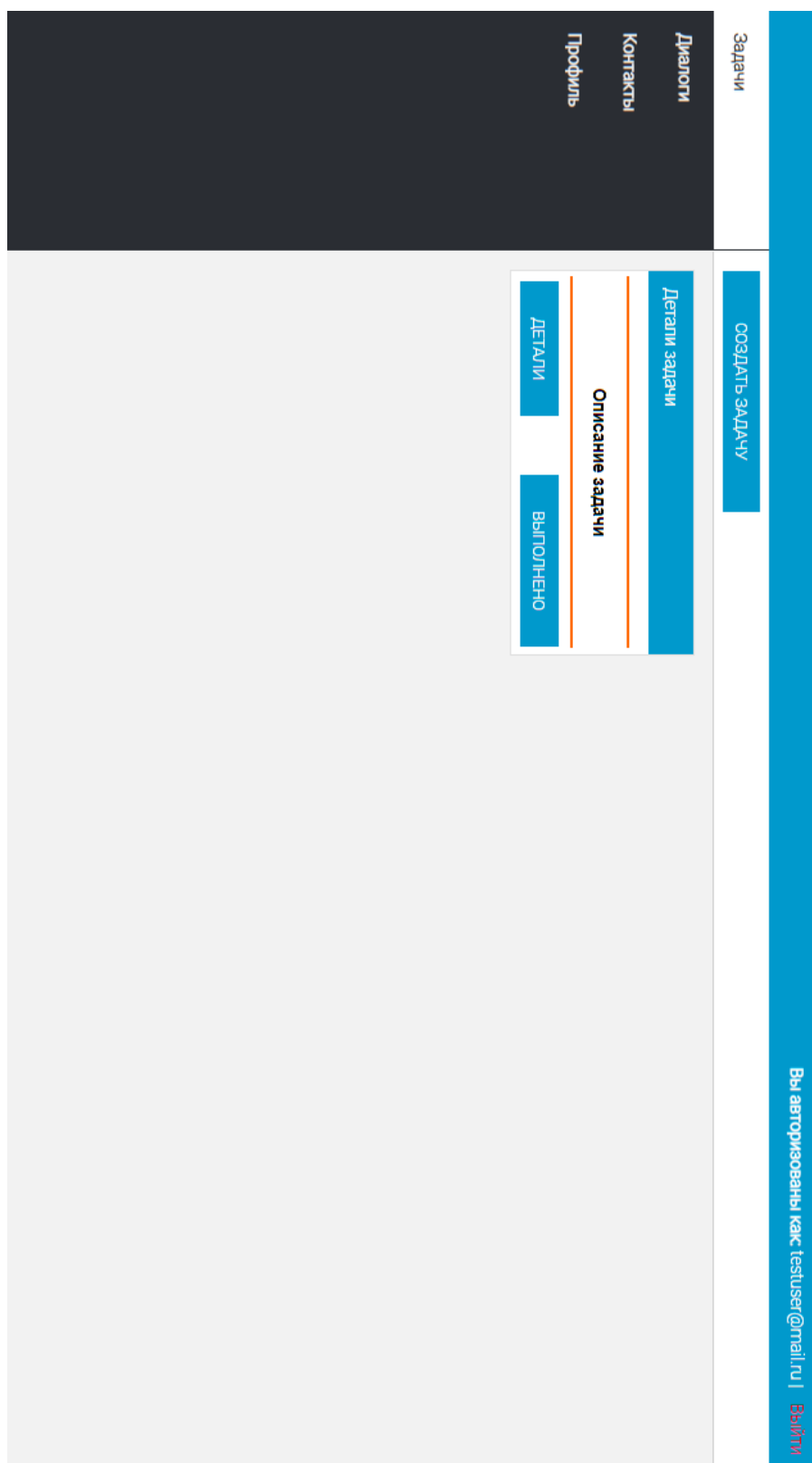


Рисунок 2 — Страница списка задач

**Заключение.** На сегодняшний день успех функционирования различных организаций зависит от многих факторов, одним из которых является правильный подход к управлению деятельностью сотрудников. Общеизвестен тот факт, что если взаимодействие внутри организации выстроено качественно, то производительность будет на высоком уровне. Одними из инструментов, используемых для достижения данной цели являются системы управления задачами.

В результате выполнения данной выпускной квалификационной работы было произведено ознакомление с современными технологиями, используемыми при разработке информационных систем, рассмотрены их преимущества и особенности. Также, в соответствии с требованиями, изложенными в пункте , была разработана собственная система управления задачами.

Был получен опыт проектирования и разработки базы данных в условиях поставленной задачи, а также разработан графический интерфейс системы.