|  |  |
| --- | --- |
| **RTK_LOGO.jpg** | **Санкт-Петербургское Государственное бюджетное профессиональное образовательное учреждение «Радиотехнический колледж»**  199155, Санкт-Петербург, наб. реки Смоленки, д.1  Тел.: (812) 405-85-59 факс (812) 405-85-59 http://www.rtplspb.ru, e-mail: info@spb-rtk.ru |

**Отчет**

об учебной практике

по профессиональному модулю

ПМ.05: «Проектирование и разработка информационных систем»

по программе подготовки специалистов среднего звена, специальности 09.02.07. Информационные системы и программирование

4 семестр

сроки прохождения практики

с «3» мая 2024 по «4» июня 2024

**Ешкилев Марк Евгеньевич**

Руководитель практики: **Харин Е. Н.**

СОДЕРЖАНИЕ

[ВВЕДЕНИЕ 3](#_Toc168408124)

[СОДЕРЖАНИЕ УЧЕБНОЙ ПРАКТИКИ 4](#_Toc168408125)

[1 Создание Use-Case диаграммы информационной системы 4](#_Toc168408126)

[2 Создание IDEF0 диаграммы информационной системы 6](#_Toc168408127)

[2.1 Создание контекстного уровня 6](#_Toc168408128)

[2.2 Декомпозиция idef0 диаграммы 1-го уровня 7](#_Toc168408129)

[2.3 Декомпозиция 2-го уровня в нотации IDEF0 8](#_Toc168408130)

[3 Создание ER диаграммы базы данных 11](#_Toc168408131)

[4 Реализация информационной системы, написание программного кода 13](#_Toc168408132)

[4.1 Регистрация и авторизация пользователя: 13](#_Toc168408133)

[4.2 Функционал для взаимодействия с постами: 13](#_Toc168408134)

[4.3 Страница со всеми постами: 13](#_Toc168408135)

[4.4 Реализация функционала для создания комментариев к постам: 14](#_Toc168408136)

[4.5 Реализация функционала администратора: 15](#_Toc168408137)

[5 Создание диаграммы mind-map информационной системы 16](#_Toc168408138)

[ЗАКЛЮЧЕНИЕ 19](#_Toc168408139)

[СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ: 20](#_Toc168408140)

# ВВЕДЕНИЕ

Специальность 09.02.07 "Информационные системы и программирование" сейчас очень востребована, потому что технологии быстро развиваются, и компаниям постоянно нужны специалисты для разработки, внедрения и поддержки программного обеспечения и информационных систем.

Цель этого отчета — показать практико-теоретическое содержание учебной практики. Вся оценочная документация из дневника практики и характеристики будет представлена руководителю практики от учебного заведения в виде отдельных документов.

Для документирования результатов учебной практики будут использоваться скриншоты и фотографии с пояснениями разных этапов работы.

Так как для выполнения заданий нужно уделять внимание теоретической части — формированию основных понятий и представлений по предметной области, в отчет будет включена информация из дополнительных источников. Результаты работы будут ежедневно записываться в дневник практики и сопровождаться оценкой руководителя, но в отчет эти данные включены не будут.

# СОДЕРЖАНИЕ УЧЕБНОЙ ПРАКТИКИ

## 1 Создание Use-Case диаграммы информационной системы

На Use-case диаграммах (рисунок 1) отображается взаимодействие между вариантами использования, представляющими функции системы, и действующими лицами, представляющими людей или системы, получающие или передающие информацию в данную систему.

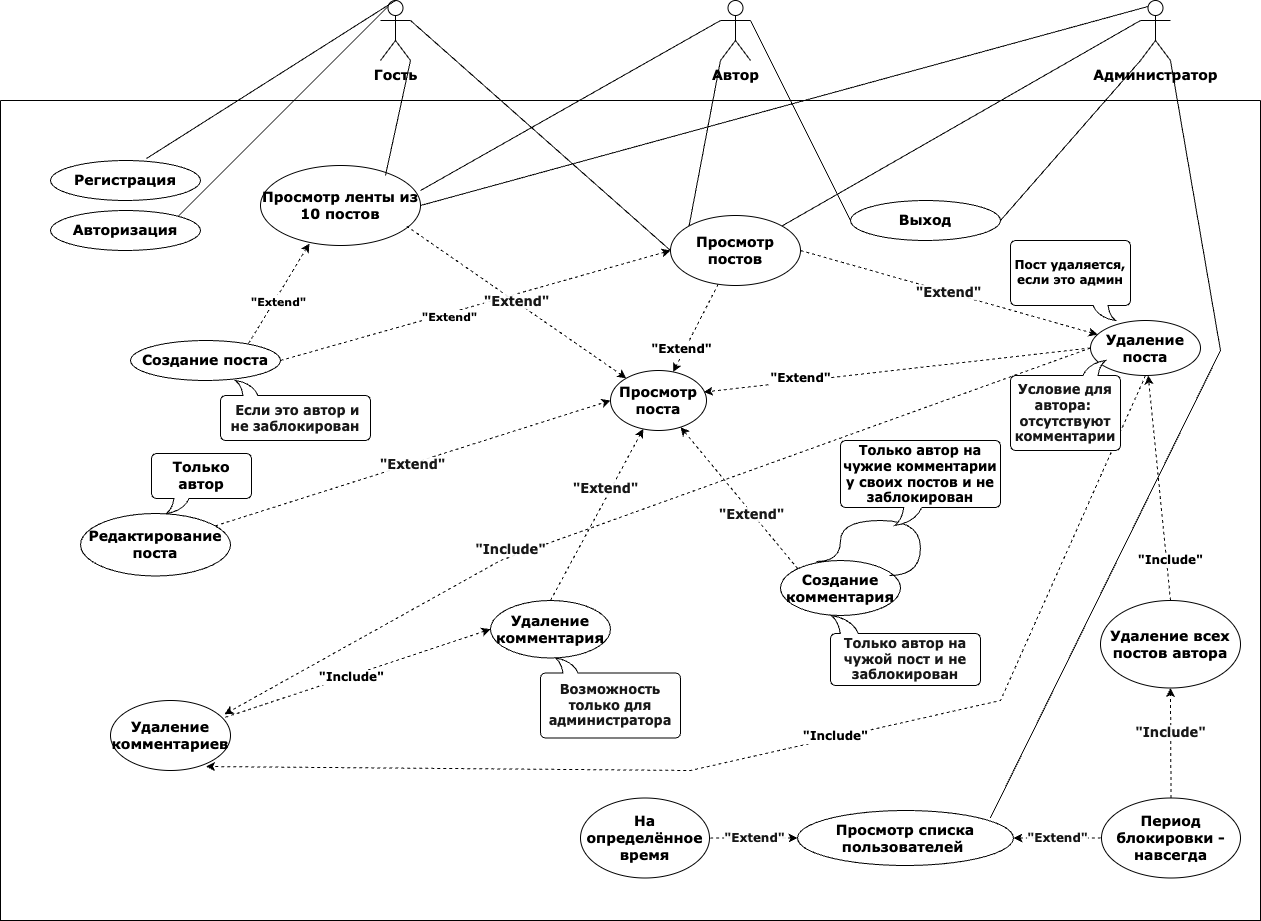


Рисунок 1 – Use case диаграмма

Основываясь на техническом задании, было выделено 3 роли, а также функционал доступный в зависимости от того какая роль у пользователя.

Роли:

1. гость (неавторизированный пользователь);
2. автор (авторизированный пользователь, способный создавать контент).
3. администратор (авторизированный пользователь, занимающийся управлением пользователями);

Функционал:

* регистрация: возможность зарегистрироваться в системе и добавить аватар;
* авторизация: процесс входа в систему;
* просмотр всех постов: возможность просмотра всех доступных постов в системе;
* просмотр ленты постов: возможность просмотра ленты последних 10 постов;
* просмотр поста: просмотр конкретного поста;
* создание/редактирование поста: возможность создания или редактирования поста. (доступно только для авторов);
* написание комментариев: возможность добавлять комментарии к постам. (доступно только авторам и только под чужими постами);
* удаление комментария: возможность удаления комментария. (доступно только для администратора);
* удаление всех комментариев: возможность удаления всех комментариев. (включается в удаление поста);
* удаление поста: возможность удаления поста. (доступно для автора поста, при условии, что количество комментариев равно нулю и администратора в независимости от количества комментариев.);
* удаление всех постов: возможность удаления всех постов (включается в перманентную блокировку);
* выход: выход из системы;
* просмотр списка пользователей: возможность просматривать список всех пользователей системы (только для администратора);
* блокировка пользователя;
* временная блокировка: временное ограничение доступа пользователя к системе;
* постоянная блокировка полное ограничение доступа пользователя к системе;
* удаление всех постов и комментариев после постоянной блокировки все посты и комментарии пользователя будут удалены.

2 Создание IDEF0 диаграммы информационной системы

### **2.1 Создание контекстного уровня**

Диаграммы нотации IDEF0 отображают структуру и функции системы, а самое главное, то какие информационные потоки с каким функционалом взаимодействуют. Первым этапом диаграммы будет ее контекстный уровень (рисунок 2)



Рисунок 2 – Контекстный уровень диаграммы IDEF0

На диаграмме нулевого уровня, или же контекстного показывается вся наша информационная система в виде одного блока, слева показаны информационные потоки, которые принимает наша система для ее работы, а справа потоки, являющиеся результатом информационной системы, в нашем случае к примеру, будет все то, что мы будем видеть на экране. Сверху описаны элементы ограничивающие нашу систему, снизу то с чем и благодаря чему функционирует наша система.

#### Элементы ограничения и контроля:

* **ТЗ** - техническое задание;
* **ФЗ № 152** - нормативно-правовые акты, регулирующие обработку данных.

#### Входные данные:

* **Данные пользователя** - информация о пользователе;
* **логин и пароль** - учетные данные пользователей для доступа к системе;
* **данные поста** - информация, связанная с созданием и редактированием постов
* **данные комментариев** - информация для создания комментариев.;

#### Элементы, благодаря которым и с чем работает система:

* **администратор** - управляет пользователями;
* **гость** - имеет доступ к просмотру постов, но не к созданию контента;
* **автор** - создает и редактирует посты на форуме;
* **интернет** - среда, через которую осуществляется доступ к форуму.

#### Выходная информация:

* **последние 10 постов** - отображаются на главной странице форума;
* **все посты** - полный список постов на форуме
* **полная информация о посте** - детальная информация о каждом посте.;
* **список пользователей** - информация о зарегистрированных пользователях форума;
* **комментарии поста** - список комментариев к конкретному посту.

### **2.2 Декомпозиция idef0 диаграммы 1-го уровня**

Диаграмму IDEF0 можно декомпозировать, до достижения нужной детализации, благодаря чему систему можно разбить на очень подробные функциональные блоки. Ниже описана декомпозиция первого уровня (Рисунок 3)

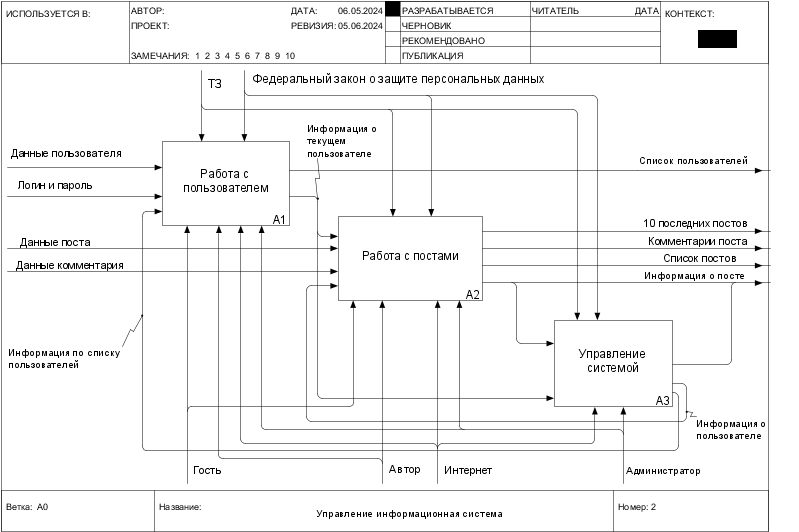


Рисунок 3 – Декомпозиция первого уровня диаграммы IDEF0

В декомпозиции 1 уровня система была разбита на 3 функциональных крупных блока:

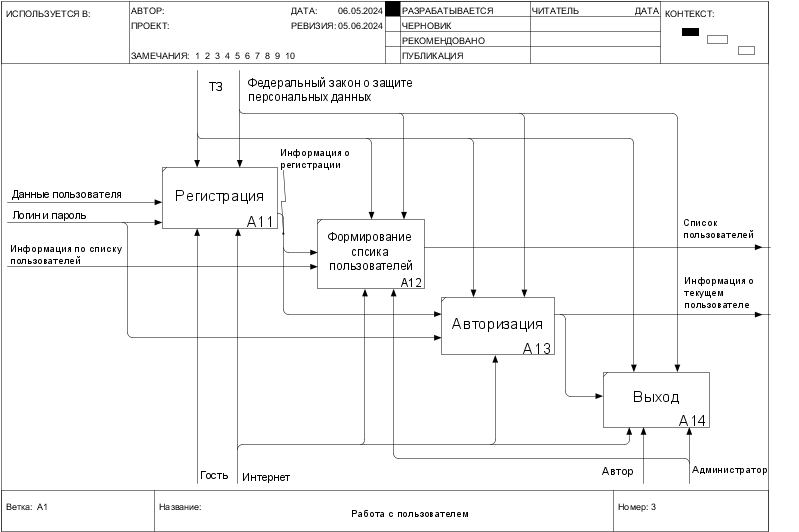
1. «Работа с пользователем»,
2. «Работа с постами»,
3. «Управление системой».

Элементы контроля для всех одинаковы Входные данные распределены по тому, где конкретно они нужны, поэтому данные пользователя идут в блок «Работа с пользователем» а данные для поста в «Работу с постами».

«Работа с пользователем» дает текущую информацию о пользователе в «Работу с постами» на выходе мы получим все связанное с постами, ленты, их информацию и комментарии, «Работа с пользователем» и «Управление системой» формируют список пользователей.

### 2.3 **Декомпозиция 2-го уровня в нотации IDEF0**

На рисунке 4 видно из какого функционала состоит данный блок, а также информационные потоки, которые идут из блока в блок.

Рисунок 4 – Декомпозиция блока «Работа с пользователем»

Здесь видно из какого функционала состоит данный блок, а также информационные потоки, которые идут из блока в блок. Для фукнции «выхода» нам надо знать данные пользователя, чтобы произвести этот самый выход, их мы получим из авторизации. При регистрации же новый пользователь заноситься в базу системы, из которой формируется список пользователей, что здесь и показано.

На декопмпозиции «Работа с постами» подробно показано, какой функционал работает с каикими данными, и как взаимодействует с другим функционалом (рисунок 5).

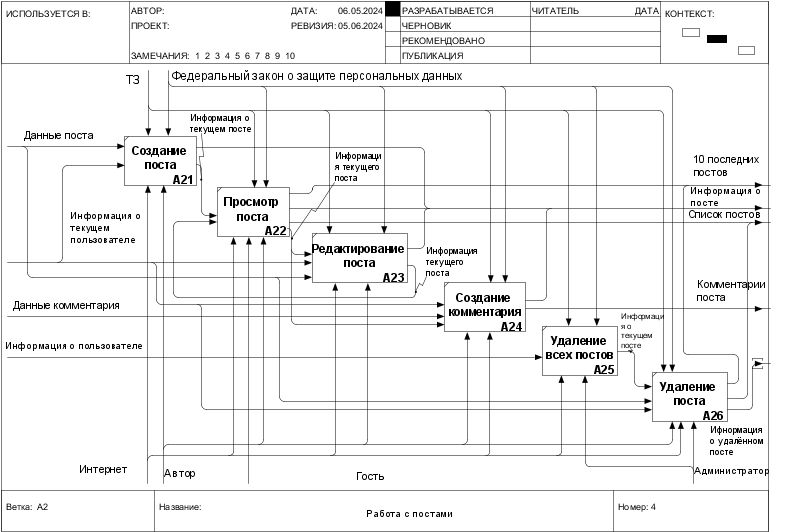
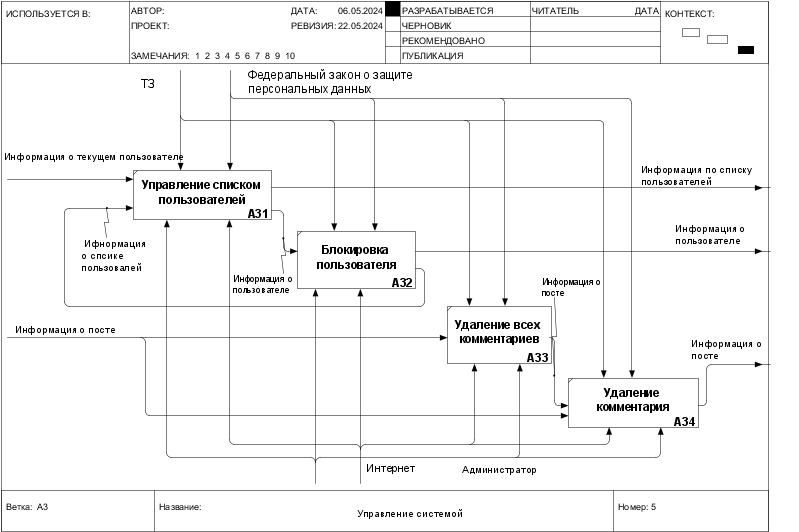


Рисунок 5 – Декомпозиция блока «Работа с постами»

Функция «Создания поста» принимает в себя данные для поста, которые ввел пользователь и информацию об этом пользователь, чтобы система понимала, какой именно пользователь хочет создать пост. После этого информация о посте идет в «просмотр поста», что в дальнейшем выдает нам ленту с постами или ленту со всеми постами, а также в «Создание комментария».

Рисунок 6 – Декомпозиция блока «Управление системой»

На декомпозиции блока «Управление системой» (рисунок 6) видно, что для любого типа блокировки нам нужна информация о пользователе, если временная блокировка просто обновит нам список пользователей, где будет видно, что пользователь заблокирован, то пермаментная блокировка запустит цепочку удаления контента пользователя.

3 Создание ER диаграммы базы данных

ER диаграммы показывают какие сущности будут в нашей системе, связь между ними а так же их атрибуты. На основе созданной диаграммы (рисунок 7) будет построена база данных нашей системы.

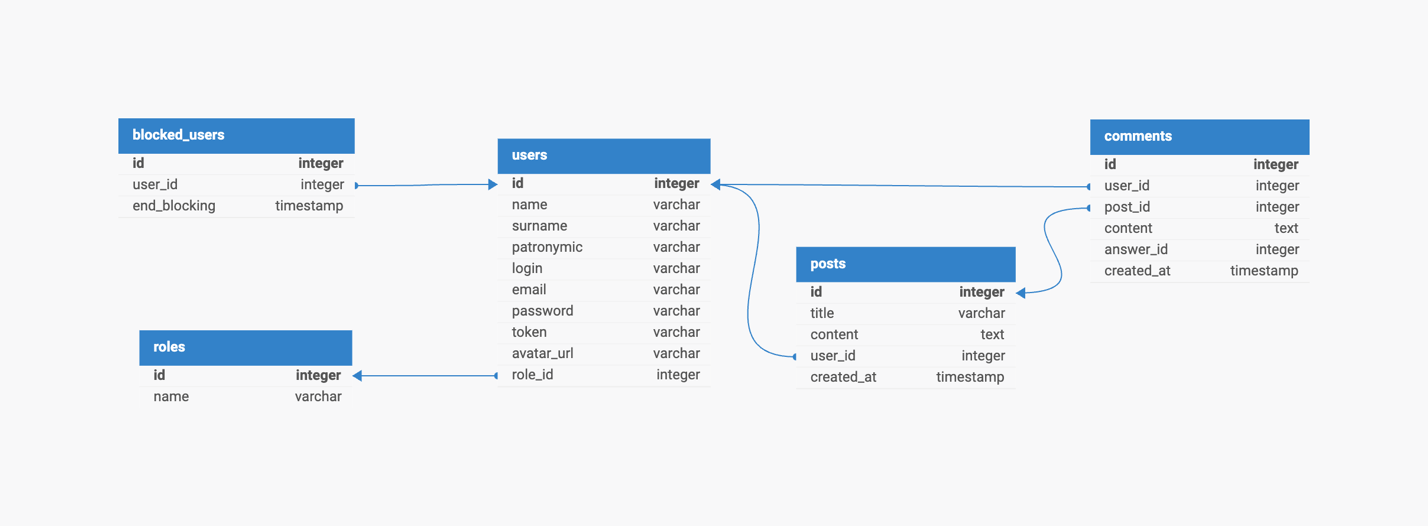


Рисунок 7 – ER диаграмма

* user (пользователь):
  1. Id (идентификатор пользователя, integer): уникальный номер пользователя;
  2. name (имя, varchar): имя пользователя;
  3. surname (фамилия, varchar): фамилия пользователя;
  4. patronymic (отчество, varchar): отчество пользователя;
  5. login (логин, varchar): логин или имя пользователя для входа в систему;
  6. email (электронный адрес, varchar): адрес электронной почты пользователя;
  7. password (пароль, varchar): пароль пользователя;
  8. token (токен, integer): уникальный идентификатор токена для пользователя;
  9. avatar\_url (аватар, varchar): ссылка на изображение аватара пользователя;
  10. role\_id (роль, integer): идентификатор роли пользователя;
* role (роль):
  1. id (идентификатор роли, integer): уникальный номер роли;
  2. name (название роли, varchar): название роли;
* post (пост):
  1. id (идентификатор поста, integer): уникальный номер поста;
  2. user\_id (пользователь, integer): идентификатор пользователя, создавшего пост;
  3. title (заголовок, varchar): заголовок поста;
  4. content (текст, text): содержимое поста;
  5. created\_at (дата создания, timestamp): дата и время создания поста.
* comment (комментарий):
  1. id (идентификатор комментария, integer): уникальный номер комментария
  2. content (содержимое, text): текст комментария;
  3. post\_id (пост, integer): идентификатор поста, к которому относится комментарий;
  4. user\_id (пользователь, integer): идентификатор пользователя, оставившего комментарий;
  5. answer\_id (комментарий, integer): идентификатор родительского комментария (для вложенных комментариев).
  6. created\_at (дата создания, timestamp): дата и время создания комментария.
* blocked\_users (блокировка):
  1. id (идентификатор блокировки, integer): уникальный номер записи блокировки;
  2. end\_blocking (дата, timestamp): дата и время блокировки.;
  3. user\_id (пользователь, integer): идентификатор заблокированного пользователя.

## 4 Реализация информационной системы, написание программного кода

### 4.1 Регистрация и авторизация пользователя:

На данном этапе была реализована регистрация и авторизация пользователя. Был создан класс «User», и с помощью его методов, а также заранее созданных классов для работы с базой данных и методами получения данных от пользователя, сначала система получала его данные после успешной валидации заносила их в базу данных, если данные были не корректны, то пользователю высвечивалась ошибка (рисунок 8).

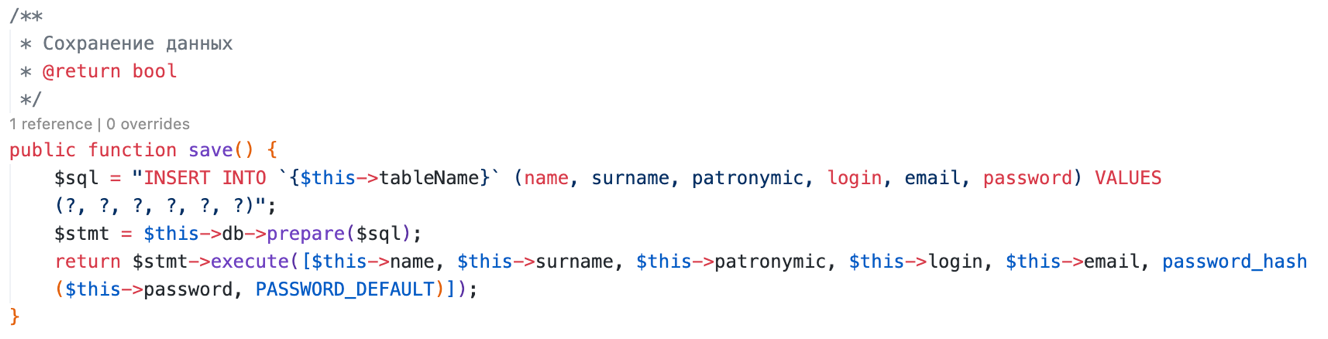


Рисунок 8 – метод save для занесения данных пользователя в базу данных

### 4.2 Функционал для взаимодействия с постами:

На данном этапе была реализована регистрация и авторизация пользователя. Был создан класс «Post», с помощью его методов, сначала система получала его данные после успешной валидации заносила их в базу данных, если данные были не корректны, то пользователю высвечивалась ошибка. При удалении поста система проверяла роль пользователя и количество комментариев, после чего исходя из технического задания, либо проводилось удаление поста, либо нет. Пример кода показан на рисунке 9.



Рисунок 9 – метод создания поста

### 4.3 Страница со всеми постами:

Для реализации данного функционала, был создан метод, формирующий массив объектов, после чего уже в файле страницы со всеми постами данный массив был перебран и каждый пост выведен на страницу. Пример метода рисунок 10.



Рисунок 10 – метод создающий массив всех постов

### 4.4 Реализация функционала для создания комментариев к постам:

Принимая данные от пользователя, система проводит проверку на пустоту поля, после чего комментарий заносился в базу. Так как пользователь может отвечать на комментарии других пользователей, для вывода комментариев был разработаны методы формирования массива комментариев. Метод создания комментария и сохранения его в базу показан на рисунке 11.



Рисунок 11 – метод создания комментария

### 4.5 Реализация функционала администратора:

Были разработаны метод для временной и перманентной блокировки пользователя, при перманентной блокировке у пользователя удаляются все его посты и комментарии. Метод блокировки представлен на рисунке 12.

****

Рисунок 12 – метод блокировки пользователя

## 5 Создание диаграммы mind-map информационной системы

#### Общие элементы меню:

* **главная,**
* **блоги,**
* **о нас,**
* **вход,**
* **регистрация,**
* **выход.**

Страницы информационной системы:

#### главная страница;

* о нас;
* страница с подробной информацией о посте;
* лента с 10 постами;
* страница со всеми постами;
* страница авторизации;
* страница регистрации;
* страница создания редактирования поста;
* страница «пользователи»;
* страница «temp-block»;

Главная страница включает в себя два функциональных блоков:

* **блок с 10 последними постами** отображает последние 10 постов, опубликованных в системе;
* **блок одного поста** включает информацию о конкретном посте и элементы управления постом;

Страница "О нас" содержит блок с информацией о информационной системе и разработчике.

На странице сподробной информацией о посте представлена подробная информация о выбранном посте, а также элементы управления постом:

* кнопка редактирования поста
* кнопка удаления поста.

Страница авторизации включает следующие элементы:

* **заголовок формы;**
* **форма авторизации**: Состоит из полей для ввода логина и пароля, а также кнопки "Вход";
* **Информация о блокировке**: Сообщение о блокировке пользователя, если он заблокирован;
* **Ошибки полей**: Пустое поле, использование специальных символов, длина более 255 символов, меньше 6 символов и т.д;

Страница регистрации включает форму для создания нового пользователя:

* поля для ввода имени, фамилии, отчества, логина, пароля, повторного ввода пароля и фото профиля;
* кнопка "регистрация";
* **ошибки полей**: пустое поле, превышение 255 символов, несовпадение паролей, логин, который уже существует в базе данных.

Страница с лентой всех постов имеет следующие блоки:

* **блок всех постов,**
* **блок одного поста**: Включает информацию о посте и элементы управления:
* кнопка редактирования поста,
* кнопка удаления поста,
* кнопка "подробнее".
* **кнопка создания поста**: Позволяет создать новый пост;

На странице "Пользователи" представлен список всех пользователей системы:

* **заголовок списка "Пользователи"**
* **список пользователей**: Содержит столбцы с номером пользователя, именем, фамилией, логином, почтой, информацией о временной и постоянной блокировке
* **блок пользователя**: Включает кнопку "Заблокировать" для блокировки пользователя и информацию о блокировке.

#### Страница временной блокировки (Temp-Block):

* **заголовок формы**
* **логин пользователя**: Поле для ввода логина пользователя, которого блокируют
* **форма блокировки**: Поле для ввода даты блокировки и кнопка "Блокировать";

Страница создания (редактирования) поста:

* **заголовок формы**
* **форма создания поста**: Включает поля для ввода заголовка, превью, содержания, добавления фотографии и кнопку "Создать"
* **ошибки полей**: пустое поле

Миниатюра диаграммы изображена на рисунке 13, также диаграмма доступна по ссылке <https://atlas.mindmup.com/2024/06/8ca3df50245311efb8541b6b9d2c4f40/map/index.html>

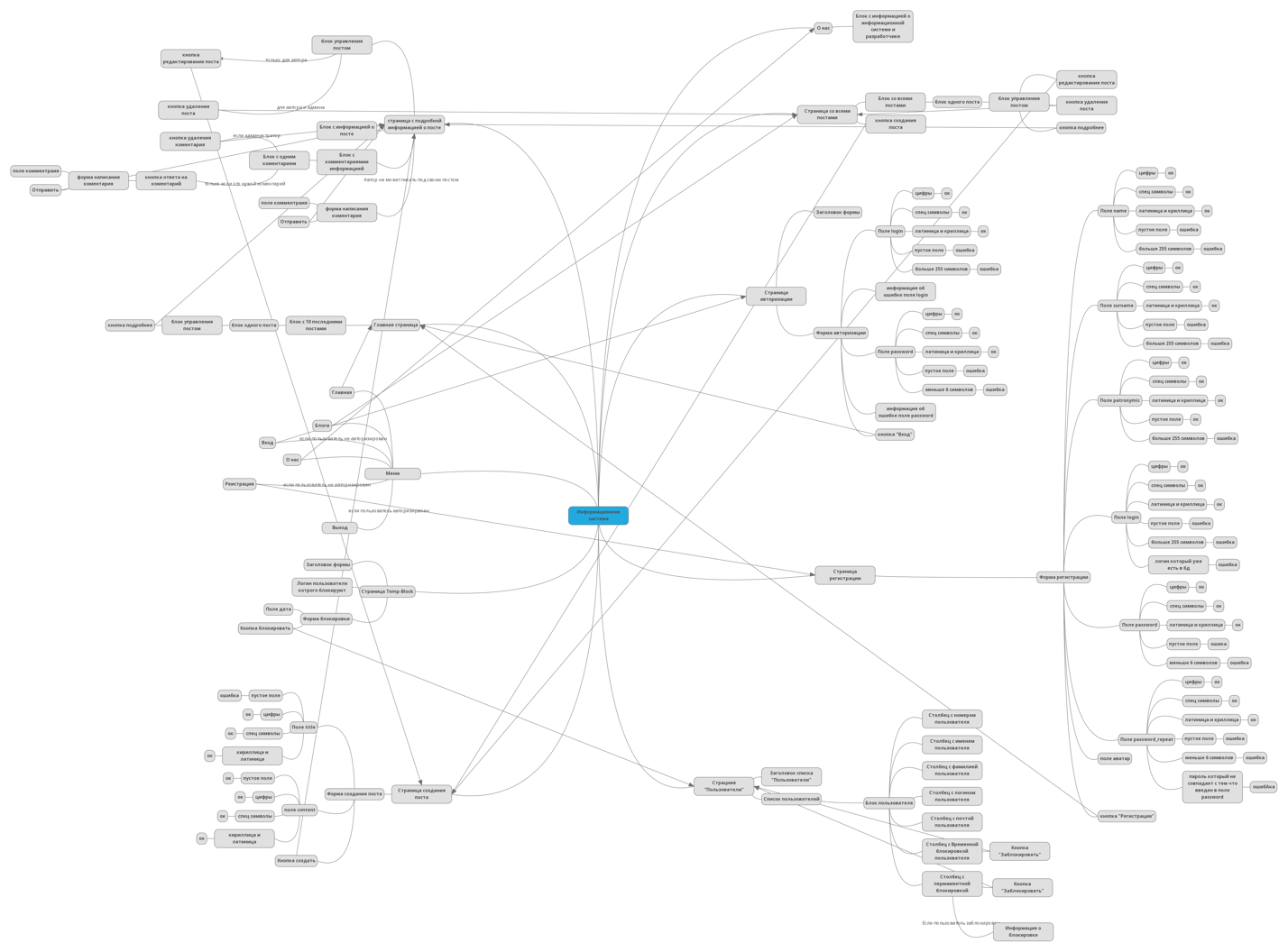


Рисунок 13-миниатюра mind map

# ЗАКЛЮЧЕНИЕ

Таким образом, за время прохождения учебной практики были освоены все необходимые практические компетенции (ПК 5.1 — ПК 5.7), предполагаемые профессиональным модулем ПМ.05: «Проектирование и разработка информационных систем».

В процессе выполнения задач разрабатывались и анализировались различные диаграммы, описывающие работу и структуру информационной системы. Среди них были диаграммы Use-Case, IDEF0, DFD, ER, а также диаграмма классов и Mind Map системы. На заключительном этапе была реализована информационная система с написанием программного кода.

Эти знания и умения окажутся полезными в дальнейшей профессиональной деятельности, что подтверждает значимость данного этапа обучения. Полученные результаты продемонстрировали способность к системному мышлению, аналитическому подходу и навыкам проектирования и разработки информационных систем, что соответствует требованиям современной IT-индустрии.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Кроссплатформенное программное обеспечение для построения визуальных диаграмм, используемых для наглядного отображения различных бизнес процессов [Электронный ресурс]. — Режим доступа: https://ramussoftware.com/, свободный.
2. Кроссплатформенное программное обеспечение для построения ER диаграмм [Электронный ресурс]. — Режим доступа: https://erd.dbdesigner.net/, свободный.
3. Кроссплатформенное программное обеспечение для построения mind map диаграмм [Электронный ресурс]. — Режим доступа https://app.mindmup.com/map/\_free/, свободный.
4. Техническая документация [Электронный ресурс]. — Режим доступа: https://www.php.net/,свободный.