МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ

РОССИЙСКОЙ ФЕДЕРАЦИИ

федеральное государственное автономное образовательное учреждение высшего образования «Санкт-Петербургский политехнический университет Петра Великого»

(ФГАОУ ВО «СПБПУ»)

**ИНСТИТУТ СРЕДНЕГО ПРОФЕССИОНАЛЬНОГО ОБРАЗОВАНИЯ**

|  |  |
| --- | --- |
|  | **Допустить к защите**  Заместитель директора  по учебно-методической работе  \_\_\_\_\_\_\_\_\_ Е.Г. Конакина\_\_  (Подпись) (И.О.Ф)  «\_\_\_»\_\_\_\_\_\_\_\_\_\_ 20\_\_ г. |
|  |

**ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА**

Тема Разработка IT-форума "ProgHelp" для ООО "Смарт

Трейд Маркетинг"

специальность 09.02.03 группа 428/2

Студент (ка) Кумачев А. А

(подпись) (ФИО)

Руководитель Яценко А. И.

(подпись) (ФИО)

Санкт-Петербург

2022

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ

РОССИЙСКОЙ ФЕДЕРАЦИИ

федеральное государственное автономное образовательное учреждение высшего образования «Санкт-Петербургский политехнический университет Петра Великого»

(ФГАОУ ВО «СПБПУ»)

**ИНСТИТУТ СРЕДНЕГО ПРОФЕССИОНАЛЬНОГО ОБРАЗОВАНИЯ**

**УТВЕРЖДАЮ**

Заместитель директора

по учебно-методической работе

\_\_\_\_\_\_\_\_\_ Е.Г. Конакина \_\_\_

(Подпись) (ИОФ)

«\_\_\_»\_\_\_\_\_\_\_\_\_\_ 202\_\_ г.

**ЗАДАНИЕ**

**НА ВЫПУСКНУЮ КВАЛИФИКАЦИОННУЮ РАБОТУ**

**Студенту(ке) \_\_**4**\_\_\_\_\_ курса \_**428/2**\_\_ группы, специальности\_\_\_**09.02.03**\_\_\_\_\_\_\_\_\_\_\_**

**\_\_\_\_\_\_\_\_** «Программирование в компьютерных системах» **\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**Кумачеву Антону Александровичу**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

(Фамилия И.О.)

**Тема: \_\_\_\_\_** Разработка IT-форума "ProgHelp" для

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_** ООО "Смарт \_Трейд Маркетинг"**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Тема предложена \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**ООО «Смарт Трейд Маркетинг» **\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

(Наименование организации)

**Дата выдачи задания \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**С заданием ознакомлен \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Срок выполнения ВКР \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Руководитель ВКР \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

(Подпись)

Яценко Анна Ивановна

(Фамилия И.О.)

**Рассмотрено и одобрено на заседании предметно-цикловой комиссии \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Протокол № \_\_\_\_ от \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Председатель предметно-цикловой комиссии \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

(Подпись)

(Фамилия И.О.)

**СОДЕРЖАНИЕ ПОЯСНИТЕЛЬНОЙ ЗАПИСКИ**

**Введение**

**Общая часть**

1.1. Анализ предметной области

1.2. Постановка задачи

1.3. Функции проектируемого приложения

1.4. Анализ существующих решений

1.5. Архитектура приложения

1.6. Обоснование методики, технологии и инструментальных средств разработки

**Специальная часть**

2.1. Разработка базы данных

2.2. Объектно-ориентированное программирование системы

2.3. Разработка пользовательского интерфейса

2.4. Тестирование мобильного приложения

2.5. Автоматизация развертывания сервера

**Экономическая часть**

3.1. Технический и социальный эффект

3.2. Организационная часть

3.3. Расчет затрат на разработку программного продукта

**Техника безопасности и охрана труда**

4.1. Анализ условий труда программиста

4.2. Расчет искусственного освещения рабочего места программиста

4.3. Электробезопасность

4.4. Пожарная безопасность

**Заключение**

**Список источников**

**Приложение А**

**Приложение Б**

**ЛИСТ ГОТОВНОСТИ ВЫПУСКНОЙ КВАЛИФИКАЦИОННОЙ РАБОТЫ**

**К ЗАЩИТЕ**

Фамилия, имя, отчество студента (ки) \_Кумачев Антон Александрович\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Специальность \_\_09.02.03 «Программирование в компьютерных системах» \_\_\_\_\_\_\_\_\_\_\_\_

Тема ВКР \_\_\_\_Разработка IT-форума "ProgHelp"

для ООО "Смарт Трейд Маркетинг"\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Замечаний нет:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| № п/п | Консультант | Фамилия, имя, отчество | Подпись | Дата |
| 1 | Руководитель выпускной  квалификационной работы |  |  |  |
| 2 | Экономическая часть |  |  |  |
| 3 | Техника безопасности и  охрана труда |  |  |  |
| 4 | Графическая часть |  |  |  |
| 5. | Нормоконтроль |  |  |  |

Допустить к защите

Председатель ПЦК\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ «\_\_\_» \_\_\_\_\_\_\_\_\_ 20 \_\_ г.

Назначить защиту выпускной квалификационной работы на «\_\_\_» \_\_\_\_\_\_\_\_\_ 20 \_\_ г.

Заведующий отделением ОИТ\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ «\_\_\_» \_\_\_\_\_\_\_\_\_ 20 \_\_ г.

(Подпись) (ФИО)

**ОТЗЫВ\***

руководителя выпускной квалификационной работы

Тема \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

ФИО \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

студента (ки)\_\_\_\_\_\_\_\_\_\_\_\_\_\_ группы

специальности \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(код и название)

Работа заслуживает\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ оценки.

Место работы и должность руководителя \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Фамилия, имя, отчество \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

|  |  |
| --- | --- |
| «\_\_\_\_» \_\_\_\_\_\_\_\_\_\_ 20\_\_ г. | Подпись:\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**\***Давая заключение о выпускной квалификационной работе, наряду с характеристикой качества графических работ (при наличии), связности изложения и грамотности составления пояснительной записки, степени самостоятельности работы обучающегося над ВКР и проявленной им инициативы, следует охарактеризовать теоретическую и практическую подготовку обучающегося, выявившуюся способность решать конкретные производственные и конструкторские задачи на базе последних достижений науки и техники. Общая оценка ВКР дается по пятибалльной системе.

**РЕЦЕНЗИЯ\*\***

на выпускную квалификационную работу

Тема \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

ФИО \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

студента (ки)\_\_\_\_\_\_\_\_\_\_\_\_\_\_ группы

специальности \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(код и название)

Работа заслуживает\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ оценки.

Место работы и должность рецензента \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Фамилия, имя, отчество \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

|  |  |
| --- | --- |
| «\_\_\_\_» \_\_\_\_\_\_\_\_\_\_ 20\_\_ г. | Подпись:\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |

\*\* Рецензия должна содержать: 1. Заключение об актуальности темы и степени соответствия выполненной выпускной квалификационной работы заданию на ВКР; 2. Характеристику выполнения каждого раздела ВКР, его научно-технический уровень; 3. Оценку качества выполнения графической части ВКР (при наличии); 4. Достоинства и недостатки ВКР. Общая оценка ВКР дается по пятибалльной системе.

**СОДЕРЖАНИЕ**

[ВВЕДЕНИЕ 2](#_Toc104736461)

[1. Аналитическая часть 3](#_Toc104736462)

[1.1. Анализ предметной области 3](#_Toc104736463)

[1.2. Постановка задачи 4](#_Toc104736464)

[1.3. Функции проектируемого приложения 4](#_Toc104736465)

[1.4. Анализ рынка существующих решений 6](#_Toc104736466)

[1.5. Архитектура приложения 6](#_Toc104736467)

[1.6. Обоснование методики, технологии и инструментальных средств проектирования и разработки 7](#_Toc104736468)

[2. Специальная часть 10](#_Toc104736469)

[2.1. Разработка базы данных 10](#_Toc104736470)

[2.2. Объектно-ориентированное проектирование системы 11](#_Toc104736471)

[2.3. Разработка пользовательского интерфейса 14](#_Toc104736472)

[2.3.1. Разработка логотипа 14](#_Toc104736473)

[2.3.2. Разработка графических элементов интерфейса 14](#_Toc104736474)

[2.4. Тестирование мобильного приложения 16](#_Toc104736475)

[2.5. Автоматизация развертывания сервера 18](#_Toc104736476)

[3. Экономическая часть 20](#_Toc104736477)

[3.1. Технический и социальный эффект 20](#_Toc104736478)

[3.2. Организационная часть 20](#_Toc104736479)

[3.3. Расчет затрат на разработку программного продукта 23](#_Toc104736480)

[3.3.1. Затраты на оплату труда разработчиков 23](#_Toc104736481)

[3.3.2. Затраты по страховым взносам 24](#_Toc104736482)

[3.3.3. Затраты на техническое и программное обеспечение 25](#_Toc104736483)

[3.3.4. Затраты на электроэнергию 25](#_Toc104736484)

[3.3.5. Затраты на специальное программное обеспечение 26](#_Toc104736485)

[3.3.6. Затраты на расходные материалы 26](#_Toc104736486)

[3.3.7. Затраты по накладным расходам 27](#_Toc104736487)

[3.3.8. Расчет цены программного продукта 28](#_Toc104736488)

[4. Техника безопасности и охрана труда 31](#_Toc104736489)

[4.1 Анализ условий труда программиста, работающего в ООО «Смарт Трейд Маркетинг» 31](#_Toc104736490)

[4.2 Расчет искусственного освещения рабочего места программиста в ООО «Смарт Трейд Маркетинг» 34](#_Toc104736491)

[4.3 Электробезопасность ООО «Смарт Трейд Маркетинг» 36](#_Toc104736492)

[4.4 Пожарная безопасность в ООО «Смарт Трейд Маркетинг» 38](#_Toc104736493)

[ЗАКЛЮЧЕНИЕ 40](#_Toc104736494)

[СПИСОК ИСПОЛЬЗОВАННЫХ ЛИТЕРАТУРНЫХ ИСТОЧНИКОВ 41](#_Toc104736495)

[ПРИЛОЖЕНИЕ А 42](#_Toc104736496)

[ПРИЛОЖЕНИЕ Б 50](#_Toc104736497)

**ВВЕДЕНИЕ**

С растущей популярностью профессий, связанных с IT, появилась потребность в коммуникации специалистов и взаимовыручки. До популяризации смартфонов для этого было достаточно обычного сайта, но с ростом количества мобильных устройств появилась потребность в мобильных клиентах для традиционных веб-приложений.

Благодаря IT-форумам программист может быстро найти решение существующей проблемы, а в случае, если проблема встречена впервые, задать вопрос сам. Это увеличивает эффективность работы IT-специалистов всего земного шара и позволяет им быстрее решать любую поставленную задачу и проще осваивать новые технологии разработки.

С мобильным клиентом пользователь может получить доступ к приложению из любого места. Это помогает удерживать пользователей в инфраструктуре приложения, что ускоряет его развитие.

Мобильный клиент должен быть кроссплатформенным для того, чтобы упростить поддержку приложения на всех мобильных операционных системах и сократить количество ошибок, свойственных конкретной модели устройства или прошивки.

1. **Аналитическая часть**
   1. **Анализ предметной области**

IT-форум «ProgHelp» – комплекс ПО, разработанный для общения программистов. Основной темой общения является решение проблем, связанных с рабочей деятельностью. Приложение поставляется в виде мобильного клиента, с помощью которого пользователь может авторизоваться, задать интересующий вопрос, посмотреть ответы на него, а также добавить свой ответ.

У форума должна иметься система аутентификации и авторизации. Новый пользователь при входе создает аккаунт с уникальным именем. Если пользователь уже имеет учетную запись, он должен иметь возможность войти в нее.

Комплекс ПО состоит из серверной и клиентской части – клиентская часть – мобильное приложение, устанавливается на телефон, а вся логика работы и данные хранятся на разрабатываемом серверном приложении.

Для упрощения запуска серверная часть должна быть легко настраиваемой под нужды каждой конкретной организации.

Дизайн приложения должен быть разработан в первую очередь для устройств на ОС Android и адаптирован под различные формы экранов для комфортной работы всех пользователей.

* 1. **Постановка задачи**

В результате анализа предметной области были сформированы следующие требования к комплексу ПО:

1. Необходимо разработать систему регистрации, аутентификации и авторизации пользователей.
2. Необходимо разработать сервис для просмотра списка недавних вопросов на форуме, с загрузкой большего числа вопросов при скроллинге.
3. Необходимо разработать программное обеспечение для просмотра содержимого каждого вопроса, а также списка ответов на него.
4. Необходимо сформировать систему рейтинга вопросов и ответов для посетителей форума.
   1. **Функции проектируемого приложения**

В результате постановки задачи были выделены следующие функциональные требования к информационной системе:

1. Серверная часть:
   * + 1. Регистрация пользователя;
       2. Аутентификация пользователя по логину и паролю;
       3. Продление срока жизни токена доступа без повторного ввода логина и пароля;
       4. Возможность получить список вопросов на форуме;
       5. Возможность подробного просмотра вопроса;
       6. Возможность просмотра ответов к вопросу;
       7. Возможность оценки вопроса и/или ответа;
       8. Возможность написать свой вопрос;
       9. Возможность написать свой ответ или комментарий к вопросу;
2. Клиентская часть:
   * + 1. Окно аутентификации;
       2. Список вопросов, отсортированный по дате. При скроллинге должно загружаться больше вопросов;
       3. Просмотр детальной информации о вопросе;
       4. Просмотр ответов к вопросу, с загрузкой большего числа при скроллинге;
       5. Добавление своего вопроса;
       6. Добавление ответа к вопросу;
       7. Оценка вопросов и ответов, позволяющая другим пользователям быстрее узнать интересующий ответ;
       8. Markdown-рендеринг текста вопросов и комментариев для расширенных возможностей верстки.
       9. Хранение токена доступа в памяти приложения для снижения количества вводов логина и пароля

REST-сервис должен обрабатывать запрос и отправлять ответ клиенту не более чем за 100 миллисекунд. Также, должна осуществляться проверка на сервере и на клиенте на валидность введенных данных. Клиент должен иметь возможность постепенной загрузки новых данных для снижения нагрузки на сеть и на сервер. Сервер должен быть настроен на поддержку работы в docker для быстрого развертывания экземпляра приложения на предприятии.

* 1. **Анализ рынка существующих решений**

В данный момент на рынке уже существует большое количество разнообразных форумов по IT-тематике, например, stackoverflow. Однако, ввиду внутренних ограничений некоторых IT-компаний, их сотрудники не могут пользоваться этими форумами для решения рабочих задач. Предлагаемое решение может быть развернуто системным администратором предприятия на локальном сервере и предоставлять доступ только сотрудникам для предотвращения утечек конфиденциальной информации.

Для написания серверной части можно использовать любой язык программирования, имеющий возможность написания REST API, например: Java spring, asp.net core, python Django и прочие.

Для написания мобильного клиента можно использовать кроссплатформенные фреймворки, такие как React Native или Flutter.

* 1. **Архитектура приложения**

Приложение состоит из клиентской из следующих компонентов:

1. Серверная часть
   * + 1. Сервис авторизации – осуществляет корректность ввода логина и пароля пользователя и выдает токен сгенерированный по стандарту JWT. В токене помимо данных об окончании срока действия необходимо хранить id пользователя. Также сервис должен иметь возможность генерации нового токена при отправке клиентом старого. Таким образом при частом посещении форума клиенту не нужно вводить логин и пароль.
       2. Сервис вопросов – предоставляет пользователю список вопросов, с возможностью загрузки большего количества. Также имеет функционал по просмотру и добавлению ответов на вопросы.
       3. Сервис оценок – дает пользователю возможность оценить вопрос или ответы на него
       4. База данных – в качестве системы управления базой данных используется postgresql.
       5. Развертывание приложения – серверная часть может быть запущена внутри контейнеров docker с помощью написанных конфигураций.
2. Клиентская часть:
   * + 1. Клиент должен быть кроссплатформенным, иметь возможность сборки приложения как под android, так и под ios из одного исходного кода.
       2. Окно регистрации и входа – при первом входе в приложение пользователю предоставляется возможность зарегистрироваться или войти в систему.
       3. После успешного входа пользователь попадает на страницу со списком вопросов. У каждого вопроса указан автор и количество оценок.
       4. При нажатии на вопрос открывается его подробное описание и список ответов.
       5. Пользователь имеет возможность оценки вопросов и ответов, а также написания своих.
   1. **Обоснование методики, технологии и инструментальных средств проектирования и разработки**

Рассмотрим фреймворки для создания REST сервиса:

1. Spring Framework:

Веб-фреймворк, основанный на Java. Позволяет писать масштабируемые решения, как для API, так и для MVC приложений. Имеет множество библиотек для работы с базами данных, широко поддерживается многими операционными системами и обладает практически стопроцентной обратной совместимостью.

1. Django:

Фреймворк, основанный на Python, благодаря нему можно быстро написать сервис приложения, обеспечивающий достаточную производительность для поставленной задачи.

1. ASP.NET Core

Фреймворк, основанный на C#. Поддерживает очень большое количество веб-сервисов, таких как REST, MVC, gRPC, RPC over JSON и пр.

Имеет ультимативное средство для работы с базами данных Entity Framework. С недавнего времени C# является полностью открытым языком, поэтому имеет нативную поддержку Linux, являющейся самой распространённой ОС для серверов.

1. Node.JS

Веб-фреймворк, в качестве основного языка которого используется JavaScript. Это позволяет упростить разработку всего комплекса приложений, ведь JavaScript является самым распространенным языков в Web, что делает его коммерчески самым дешевым в большинстве случаев.

Ввиду наличия у меня большого опыта в разработке на asp.net core, мной был выбран именно этот фреймворк. Знания позволяют начать изначально строить правильную архитектуру, что способствует более чистому коду у финального продукта, а значит и более высокой степени оптимизации и ускоренной адаптации новых сотрудников к проекту.

Теперь, когда инструмент для создания серверной части был выбран, выберем фреймворк для мобильной разработки:

1. Flutter

Достаточно молодой фреймворк, дающий возможность писать кроссплатформенные мобильные приложения на Dart. Приложение отлаживается на устройстве или эмуляторе в реальном времени, а для клиентов генерируется нативный код, Java или Swift в зависимости от операционной системы, который потом собирается в пакет приложения, которое может быть установлено на устройство.

1. React Native

Работает по такому же принципу, однако в основе лежит фреймворк React.Js, что позволяет из кода сгенерировать также традиционное веб-приложение. К тому же react поддерживает TypeScript, что позволяет использовать типизацию при разработке приложения, что нивелирует главный минус JS – отсутствие типов.

Для инструмента мобильной разработки был выбран React Native. React включает в себя множество средств для разработки клиентских приложений и имеет доступ ко всей инфраструктуре JavaScript, позволяющей написать клиент для любого сервиса.

1. **Специальная часть**
   1. **Разработка базы данных**

Поскольку для взаимодействия серверного приложения и базы данных был выбран ASP.NET Core, то все таблицы были созданы с помощью сгенерированного кода. Однако, для оптимизации работы приложения, каждая миграция была просмотрена и изменена вручную.

В результате, была создана база данных, состоящая из шести таблиц:

1. Authentication
   * + 1. Id (bigint) – идентификатор записи
       2. UserRef (bigint, index) – Id пользователя
       3. Password (char varying) – хэш пароля пользователя
2. Users
   * + 1. Id (bigint) – идентификатор пользователя
       2. Nickname (char varying) – имя, отображаемое на форуме
       3. Email (char varying) – электронная почта пользователя
3. Posts
   * + 1. Id (bigint) – идентификатор вопроса
       2. Description (char varying) – короткое описание для отображения в списке
       3. CreatedAt (bigint) – время создания вопроса
       4. AuthorRef (bigint, index) – Id пользователя, создавшего вопрос
       5. Text (char varying) – полное содержание вопроса
4. Comments
   * + 1. Id (bigint) – идентификатор ответа
       2. PostRef (bigint, index) – Id вопроса, которому назначался ответ
       3. AuthorRef (bigint, index) – Id пользователя, написавшего ответ
       4. CreatedAt (bigint) – время написания ответа
       5. Text (char varying) – текст ответа
5. PostLike (список пользователей, которые оценили вопрос)
   * + 1. Id (bigint) – идентификатор оценки
       2. PostRef (bigint, index) – Id вопроса
       3. UserRef (bigint, index) – Id пользователя
6. CommentLike (список пользователей, которые оценили ответ)
   * + 1. Id (bigint) – идентификатор оценки
       2. PostRef (bigint, index) – Id ответа или комментария
       3. UserRef (bigint, index) – Id пользователя

Для хранения данных используется postgresql 13.6, приложение при запуске проверяет соответствие структуры базы данных ожидаемой, и, в случае отсутствия таблиц, создает их при помощи механизма миграций баз данных.

* 1. **Объектно-ориентированное проектирование системы**

В ходе разработки были созданы классы для каждой сущности форума, а также сервисы, в которых находится логика обработки запросов пользователя. Для каждого сервиса был написан интерфейс с его методами и класс с их реализацией. Сущности форума представляют собой такую же структуру, как и таблицы базы данных, поскольку именно с помощью них генерировались эти таблицы.

Исходный код сущностей приведен в Приложении А.

Ниже приведен программный интерфейс каждого из сервисов:

1. IAccessTokenGenerator:
   * GetToken – генерирует токен доступа для пользователя
2. IAccessTokenValidator:
   * ValidateToken – проверяет токен и возвращает передаваемую в нем информацию
3. IAuthenticationService
   * Login – выдает токен доступа при корректном вводе учетных данных
   * Regenerate – генерирует новый токен доступа при валидности старого
4. IUserService
   * RegisterUser – регистрирует нового пользователя в системе
   * GetUserById – возвращает пользователя с указанным id
5. IPostService
   * CreatePost – создает новый вопрос
   * ListPosts – возвращает следующую порцию вопросов в списке
   * PostDetails – возвращает подробную информацию о вопросе
6. ICommentService
   * GetPostComments – Возвращает следующую порцию ответов на вопрос
   * CreateComment – создает новый ответ на вопрос
7. ILikeService
   * LikePost – ставит оценку на вопрос
   * LikePostRetract – отменяет установку оценки на вопрос
   * LikeComment – ставит оценку на комментарий
   * LikeCommentRetract – отменяет установку оценки на комментарий

Приведенные выше сервисы не обрабатывают запросы напрямую. Методы сервисов вызываются контроллером, который выполняет роль шлюза, проверяющего корректность передаваемых данных и данные авторизации (если требуется), а также преобразованием объектно-ориентированных сущностей в удобный для клиента формат. Для работы с базой данных используются следующие сущности:

1. ForumUser – пользователь форума

* Nickname – имя пользователя
* Email – адрес почты пользователя
* UserAuth – сущность для авторизации пользователей
* LikedPosts – список оцененных пользователем вопросов
* LikedComments– список оцененных пользователем ответов и комментариев

1. ForumPost – вопрос на форуме

* Description – краткое описание
* Text – Полный текст вопроса
* CreatedAt – Дата и время создания
* Author – автор вопроса
* Comments– список ответов/комментариев к вопросу
* LikedUsers– список оценивших вопрос пользователей

1. UserAuth – вопрос на форуме

* User – Пользователь, для которого задана авторизация
* Password – захешированный пароль пользователя

1. ForumComment– комментарий на форуме

* Post – вопрос, к которому был оставлен
* Author – Автор комментария
* CreatedAt – Дата и время создания
* Text – Текст комментария
* LikedUsers– список оценивших комментарий пользователей

1. ResultBase<TResult, TError> – результат запроса
2. SuccessResult<TResult, TError> – ответ со статусом успех и результатом выполнения запроса

* Result – результат выполнения запроса

1. ErrorResult<TResult, TError> – ответ со статусом ошибка и данными о ней

* Error – сведения об ошибке
  1. **Разработка пользовательского интерфейса**

Поскольку приложение-клиент будет запускаться на мобильных устройствах, проектированию пользовательского интерфейса было уделено отдельное внимание. Список схем и снимков финального интерфейса приведен в Приложении Б.

* + 1. **Разработка логотипа**

Для создания логотипа использовалась программа Gimp. В ней я сделал надпись, используемую на логотипе, и наложил поверх нее градиент. В результате получилось следующее изображение.



Рисунок 1 – логотип приложения

Данное изображение было в дальнейшем уменьшено в разрешении, чтобы не занимать избыточное пространство на накопителе.

* + 1. **Разработка графических элементов интерфейса**

В приложении, помимо логотипа, есть пять графических элементов интерфейса, используемых пользователем для взаимодействия.

Первый – это пиктограммы, указывающие, оценил ли пользователь вопрос либо ответ к нему. Пиктограмма отсутствия оценки – черный контур сердца без заполнения, наличия – черный контур сердца с красной заливкой.

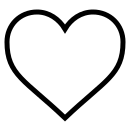


Рисунок 2 – пиктограммы оценки записей

Далее, для обозначения кнопки для добавления новой записи (вопроса или ответа) был нарисован знак ‘+’ в темно-сером круге. Для подтверждения создания был разработан значок ‘✓’ в темно-сером круге. Ниже представлены снимки экрана с данными элементами интерфейса.

Изображение выглядит как текст

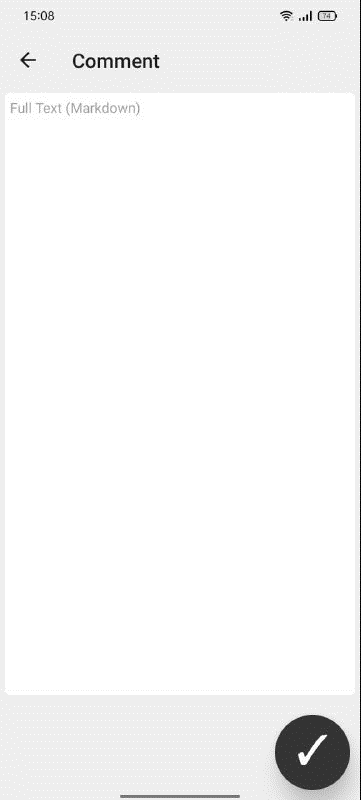
Автоматически созданное описание 

Рисунок 3 – окна с элементами в сером круге

Остальные элементы интерфейса приложения были не нарисованы в растровом редакторе, а написаны в xml-разметке приложения, что позволяет легче их использовать в дальнейшем.

* 1. **Тестирование мобильного приложения**

Ниже, в таблице 1, представлены сценарии и результаты комплексного тестирования мобильного приложения и API-сервера.

Таблица 1 – тестирование приложения

| **№ п/п** | **Сценарий теста** | **Ожидаемый результат** | **Результат** |
| --- | --- | --- | --- |
| 1 | Проверка возможности регистрации:   * Войти в приложение * Перейти на вкладку «Регистрация» * Ввести имя нового пользователя, электронную почту, пароль и повторение пароля * Нажать кнопку «зарегистрироваться» | Сервер зарегистрирует нового пользователя, а приложение откроет список недавних вопросов | Пройдено |
| 2 | Проверка возможности входа:   * Войти в приложение * Перейти на вкладку «Вход» * Ввести имя пользователя и пароль * Нажать кнопку «Войти» | Откроется список недавних вопросов | Пройдено |
| 3 | Проверка ввода неправильного логина или пароля:   * Войти в приложение * Перейти на вкладку «Вход» * Ввести имя пользователя и некорректный пароль * Нажать кнопку «Войти» | Приложение выведет сообщение о неправильном вводе учетных данных | Пройдено |
| 4 | Проверка автоматического входа в приложение:   * Войти в приложение | Откроется список недавних вопросов | Пройдено |

|  |  |  |  |
| --- | --- | --- | --- |
| **№ п/п** | **Сценарий теста** | **Ожидаемый результат** | **Результат** |
| 5 | Проверка создания нового вопроса:   * Войти в приложение * Нажать на символ ‘+’ * Ввести тему и текст вопроса * Нажать на символ ‘✓’ | Приложение выведет список вопросов, добавив в него созданный пользователем | Пройдено |
| 6 | Проверка оценки вопроса:   * Войти в приложение * Нажать на символ сердца возле вопроса | Символ сердца будет заполен красным, а счетчик оценок увеличен | Пройдено |
| 7 | Проверка создания ответа на вопрос:   * Войти в приложение * Нажать на любой вопрос * Нажать на символ ‘+’ * Написать свой ответ на вопрос * Нажать на символ ‘✓’ | Приложение перейдет к вопросу, а в списке комментариев будет добавлен новый, созданный пользователем | Пройдено |
| 8 | Проверка оценки комментария:   * Войти в приложение * Нажать на любой вопрос * Нажать на символ сердца возле комментария | Символ сердца будет заполен красным, а счетчик оценок увеличен | Пройдено |

* 1. **Автоматизация развертывания сервера**

Для автоматизации процесса запуска и настройки сервера был выбран docker и docker compose. Docker осуществляет сборку проекта в виртуальной среде, а docker compose – настройку самого сервера и всех приложений, необходимых ему (например, СУБД). Ниже приведен листинг файлов Dockerfile и docker-compose.yml, используемых в приложении.

Dockerfile:

FROM mcr.microsoft.com/dotnet/sdk:6.0 AS build

WORKDIR /app

COPY . ./

RUN dotnet restore ./ForumBackend.sln

RUN dotnet publish -c release -o build ./ForumBackend.sln

FROM mcr.microsoft.com/dotnet/aspnet:6.0 AS runtime

WORKDIR /app

COPY --from=build /app/build ./

EXPOSE 80

ENTRYPOINT ["dotnet", "ForumBackendApi.dll"]

Docker-compose.yml:

version: "3.8"

services:

  database:

    image: postgres:13.6-alpine

    ports:

      - "7002:5432"

    volumes:

      - ".:/var/lib/postgresql/data"

    networks:

      - "backend"

    environment:

      POSTGRES\_DB: "ForumData"

      POSTGRES\_USER: "forum"

      POSTGRES\_PASSWORD: "secretpasswd"

      PGDATA: "/var/lib/postgresql/data/ForumData"

  application:

    build: .

    ports:

      - "7001:80"

    networks:

      - "backend"

    restart: always

    depends\_on:

      - database

    environment:

      POSTGRES\_USERNAME: "${POSTGRES\_USERNAME}"

      POSTGRES\_PASSWORD: "${POSTGRES\_PASSWORD}"

      POSTGRES\_DATABASE\_NAME: "${POSTGRES\_DATABASE\_NAME}"

      POSTGRES\_HOST: "${POSTGRES\_HOST}"

      POSTGRES\_PORT: "${POSTGRES\_PORT}"

      AUTH\_PUBLIC\_KEY: "${AUTH\_PUBLIC\_KEY}"

      AUTH\_SECRET\_KEY: "${AUTH\_SECRET\_KEY}"

      LIFETIME: "${LIFETIME}"

networks:

  backend:

Для запуска необходимо установить на целевую систему docker и docker compose и в папке с API-приложением выполнить:

sudo docker compose up

после чего приложение будет запущено на порте 7001, а база данных на порте 7002.

1. **Экономическая часть**
   1. **Технический и социальный эффект**

Данное приложение имеет аналоги аналогов в открытом доступе, выполнено по индивидуальному заказу предприятия. Приложение имеет простой и понятный интерфейс для возможности работы с ним пользователя, не имеющего большого опыта обращения с мобильными устройствами. Данное приложение имеет весь заявленный функционал и может выполнять все требуемые задачи.

Разработанный программный продукт будет использоваться сотрудниками, в частности IT отдела, для решения внутренних задач и организации рабочего процесса.

В данное время предприятие для решения проблем разработки ПО не использует сторонних решений ввиду конфиденциальности внутренней информации. Разработка и внедрение данной системы может упростить коммуникацию между IT-специалистами.

* 1. **Организационная часть**

Трудоемкость разработки можно определить в таблице 1, т.е. строка «всего» показывает общую трудоемкость разработки.

Таблица 2 – Трудоемкость разработки программного продукта

| **Этапы работ** | **Содержание работ** | **Исполнители** | | | **Время выполнения, дни** |
| --- | --- | --- | --- | --- | --- |
| Кол-во, чел. | Должность | Оклад, руб. |  |
| Подготовительный | Ознакомление с заданием и распределение обязанностей | 1 | Аналитик | 17000 | 1 |
| Подготовительный | Установка требуемого ПО | 1 | Техник | 40000 | 1 |
| Анализ | Анализ методов решений | 1 | Аналитик | 35000 | 2 |
| Проектирование | Создание шаблона приложения | 1 | Программист | 60000 | 3 |
| Проектирование | Проектирование работы с БД | 1 | Программист | 60000 | 2 |
| Проектирование | Разработка API-сервиса приложения | 1 | Программист | 60000 | 3 |
| Проектирование | Разработка мобильного приложения | 1 | Программист | 60000 | 2 |
| Тестирование | Тестирование программного продукта | 1 | Программист | 60000 | 3 |
| Вывод в эксплуатацию | Оформление документации | 1 | Техник | 40000 | 1 |
| Всего (общая трудоемкость выполнения разработки) | | | | | 13 |

Таблица 3 – Разработчики программного продукта

|  |  |  |
| --- | --- | --- |
| **Исполнители** | **Оклад, руб.** | **Дневная тарифная ставка, руб. (Ор*i*)** |
| Аналитик | 35000 | 1590 |
| Техник | 40000 | 1818 |
| Программист | 60000 | 2727 |

Дневная тарифная ставка определяется по формуле:

Ор*i* = , руб. в день (1)

* 1. **Расчет затрат на разработку программного продукта**

Расчет полных затрат на разработку проектного решения (программного продукта) осуществляется по формуле:

Зрп = Зот + Зст + Апп (или Зтс) + Зэ/э +Зсп + Зрм + Знр, (2)

где Зот – затраты на оплату труда разработчика (разработчиков), руб.;

Зст – страховые взносы по оплате труда во внебюджетные фонды, руб.;

Апп (или Зтс) – затраты, связанные с содержанием вычислительной техники, руб.;

Зэ/э – затраты на электроэнергию, руб.;

Зсп – затраты на специальное программное обеспечение, руб.;

Зрм – затраты на расходные материалы, необходимые при разработкепрограммного продукта, руб.;

Знр – затраты по накладным расходам, приходящиеся на разработку программного продукта, руб.

* + 1. **Затраты на оплату труда разработчиков**

Размер фонда оплаты труда разработчиков рассчитывается по формуле:

Зот =, (3)

где Зот – затраты на оплату труда разработчиков, руб.;

Тр*i* – время разработки проектного решения *i-*го разработчика, дни;

Орi – заработная плата в деньi-го разработчика, руб. (Орi = );

Кд – коэффициент дополнительной заработной платы разработчика проекта (Кд = 0,12);

Количество разработчиков – n, чел.

* + 1. **Затраты по страховым взносам**

Сумма страховых взносов определяется по формуле:

Зст = Кст Зот, руб (4)

где Кст – коэффициент страховых взносов, установленный государством, для расчета отчислений во внебюджетные фонды.

Таблица 4 – Заработная плата работников, занятых разработкой проекта

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **№п/п** | **Должность** | **Оклад, руб./мес** | **Оплата в день, руб.** | **Трудоемкость, дни** | **Тарифная заработная плата, руб.** |
| 1 | Аналитик | 35000 | 1590 | 3 | 4770 |
| 2 | Техник | 40000 | 1818 | 2 | 3636 |
| 3 | Программист | 60000 | 2727 | 13 | 35451 |
| Итого тарифная заработная плата | | | | | 43857 |
| Доплаты (премия 20% от тарифной заработной платы) | | | | | 8771,4 |
| Итого основная заработная плата (тарифная заработная плата +доплаты) | | | | | 52628,4 |
| Дополнительная заработная плата (12% от основной заработной платы) | | | | | 6315,4 |
| Всего заработная плата (сумма основной и дополнительной заработной платы) (Зот) | | | | | 58493,8 |
| Страховые взносы (30% от заработной платы или процент берется по данным предприятия) (Зст) | | | | | 17683,1 |

* + 1. **Затраты на техническое и программное обеспечение**

Общая стоимость требуемого обеспечения определяется по формуле

Зоб =, (5)

где К*i*–количество*i-* го вида оборудования, шт.;

Цi–цена i-го вида оборудования, руб.;

Ктзр –коэффициент учитывающий все затраты связанные с приобретением оборудования (Ктзр = 0,15);

Количество оборудования– m, штук.

Таблица 5 – Затраты на техническое и программное обеспечение

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| №п/п | Наименование | Ед. изм. | Расход | Цена, руб. | Стоимость, руб. |
| 1 | ПК ZET Neo M010 (intel core i5 10400f, 16GB DDR4) | Шт. | 1 | 69999 | 69999 |
| Итого (общая стоимость) | | | | | 69999 |
| Затраты по доставке (10% от общей стоимости) | | | | | 6999 |
| Итого затрат с учетом доставки | | | | | 76998 |

* + 1. **Затраты на электроэнергию**

Затраты, связанные с потреблением электроэнергии определяются на основе тарифа на электрическую энергию и расхода кВт на разработку программного продукта. По Санкт-Петербургу тариф 4,65 руб./кВтч.

Таблица 6 – Расчет планируемых затрат на электроэнергию

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Наименование | Мощность, кВт | Количество часов | Количество кВт | Стоимость кВт•ч, руб. | Сумма, руб. |
| ПК ZET Neo M010 (intel core i5 10400f, 16GB DDR4) | 0,3 | 160 | 48 | 4,65 | 223,2 |
| Итого затрат на электроэнергию (Зэ/э) | | | | | 223,2 |

* + 1. **Затраты на специальное программное обеспечение**

Затраты на специальные программные продукты, необходимые для разработки проектного решения определяются по формуле:

Зсп = ,руб, (6)

где Цρ – цена ρ-го специального программного средства, руб.;

n – количество видов программных продуктов, шт.

Таблица 7 – Расчет планируемых затрат на программные продукты

|  |  |  |  |
| --- | --- | --- | --- |
| Наименование | Цена за 1 ед., руб | Количество, ед. | Сумма, руб |
| Лицензия IDE для разработки Rider | 2424 | 1 | 2424 |
| Итого затрат на специальное ПО(Зсп) | | | 2424 |

* + 1. **Затраты на расходные материалы**

Затраты на расходные материалы необходимые для разработки проектного решения определяются по формуле:

Зрм = , (7)

где Qi – количество i-го вида материала, шт.;

Pi – цена i-го вида материала, руб.;

n – количество видов материалов.

Затраты на расходные материалы удобнее определять в таблице 15.

Таблица 8 – Затраты на расходные материалы

|  |  |  |  |
| --- | --- | --- | --- |
| Наименование | Цена (без НДС), руб. | Количество, шт. | Стоимость, руб. |
| Пачка бумаги для печати формата А4 | 600 | 1 | 600 |
| USB накопитель 8Гб | 1300 | 1 | 1300 |
| Ручка шариковая | 60 | 4 | 240 |
| Карандаш простой | 30 | 4 | 120 |
| Итого | | | 2260 |
| ТЗР (15% от Итого) | | | 339 |
| Всего затрат с учетом ТЗР (Зрм) | | | 2599 |

* + 1. **Затраты по накладным расходам**

Затраты по накладным расходам определяются по формуле:

Знр = Кнр Зот,руб. (8)

где Кнр – коэффициент накладных расходов по данным предприятия

(если нет таких данных, то принять Кнр = 0,35);

Зот - затраты по оплате труда, руб., (см. таблицу 3).

Все полученные расчеты сводятся в таблицу 8.

Таблица 9 – Полные затраты на разработку программного продукта

|  |  |  |
| --- | --- | --- |
| Перечень затрат | Обозначение | Значение, руб. |
| Оплата труда | Зот | 58493,8 |
| Страховые взносы | Зст | 17683,1 |
| Затраты на технические средства | Зтс | 76998 |
| Затраты на электроэнергию | Зэ/э | 223,2 |
| Специальные программы | Зсп | 2424 |
| Расходные материалы | Зрм | 2599 |
| Накладные расходы | Знр | 20472,8 |
| Итого затрат на разработку | Зрп | 263201,48 |

* + 1. **Расчет цены программного продукта**

Цена программного продукта для работы отдела продаж определяется по формуле:

Цпо = Зрп+Пп +НДС, руб (9)

где Зрп– итого затрат на программный продуктработы отдела продаж, руб.;

Пп – планируемая прибыль (принимается = 15 % от суммы затрат на программный продукт работы отдела продаж, т.е. от Зрп) руб.;

НДС (налог на добавленную стоимость) – налог на добавленную стоимость (ставка НДС = 20%), руб.

Планируемая прибыль также может быть рассчитана исходя из нормативной рентабельности, определяемая организацией и рассчитывается по формуле:

Пп = ЗрпRп, руб (10)

где Rп – коэффициент рентабельности, определенный предприятием или принять Rп = 0,15.

Налог на добавленную стоимость определяется по ставке, установленной государством от суммы затрат на программный продукт и величины плановой прибыли по формуле:

НДС = (Зрп + Пп)Кндс, руб.(10)

где Кндс**–**коэффициент НДС (Кндс = 0,20 при ставке НДС 20 %).

В результате полученного положительного финансового результата (прибыли) предприятие должно начислить налог на прибыль (налога 20 % от полученной прибыли) по формуле:

Нприб. = Пп, руб (11)

Прибыль, оставшаяся на предприятии после уплаты налога на прибыль, определяется по формуле

Пч = ПпНприб., руб (12)

В целом поступления в бюджет от реализации программного продукта для работы отдела продаж составят:

Пбюдж.=Нприб. +НДС, руб (13)

Все полученные расчеты сводятся в итоговую таблицу 9.

Таблица 10 – Основные экономические показатели по созданию продукта

|  |  |
| --- | --- |
| Показатели | Значение, руб. |
| Затраты на разработку (Зрп) | 263201,48 |
| Налог на добавленную стоимость (НДС) |  |
| Норматив рентабельности (Rп) | 0,15 |
| Плановая прибыль (Пп) |  |
| Налог на прибыль (Нприб) |  |
| Прибыль, оставшаяся в распоряжении предприятия (Пч) |  |
| Поступления в бюджет (Пбюдж) |  |

Вывод: стоимость разработки составляет 263201,48 руб., после продажи, чистая прибыль предприятия составит 31584 руб. Цена, согласно формуле 9, составит 363217,48 руб.

# **Техника безопасности и охрана труда**

**4.1 Анализ условий труда программиста, работающего в ООО «Смарт Трейд Маркетинг»**

Рабочее место программиста — это его рабочий кабинет, либо часть помещения, в котором он проводит большую часть работы. Совершенно очевидно, что чем лучше приспособлено рабочее место для выполнения функций программиста, тем производительнее и эффективнее будет его трудовая деятельность.

Программист имеет собственное рабочее место, оборудованное ПЭВМ. Рабочий стул (кресло) должен быть подъемно-поворотным и регулируемым по высоте и углам наклона сиденья и спинки, а также – расстоянию спинки до переднего края сиденья.

Рабочая поверхность стола сделана из дерева. На столе размещены ЭВМ, монитор и принтер. Площадь поверхности стола предполагает размещение монитора, устройств ввода/вывода и рабочей зоны с местом расположения оперативной документации.

Кабинет, в котором расположено рабочее место, представляет собой помещение площадью 30 м2, длиной 5 м, шириной 6 м и высотой 3 м.

Для создания благоприятных условий для зрительного восприятия кабинет оформлен в светло-бежевом цвете.

Для создания и поддержания независимо от наружных условий оптимальных значений температуры, влажности, чистоты воздуха, в холодное время используется водяное отопление, в теплое время года применяется кондиционирование воздуха.

Режим работы программиста - односменный с пятидневной рабочей неделей и нормированным рабочим днем. Продолжительность ежегодного отпуска – 28 календарных дней.

При идентификации вредных производственных факторов было установлено, что опасными могут являться:

* шум и вибрация, источником которого является оргтехника;
* электромагнитные поля и излучения от дисплея ПК;
* статическое электричество, накапливаемое на клавиатуре, дисплее, корпусе системного блока;
* электрический ток в электрических сетях;
* запыленность рабочей зоны, так влажная уборка рабочей поверхности проводиться не ежедневно, а только по мере загрязнения;
* статические перегрузки, так как работа программиста связана с неизменной статической позой, иногда неудобной позе (работа за ПК), что может привести к искривлению позвоночника, остеохондрозу, застою в органах малого таза;
* перенапряжение анализаторов, в большей степени зрительного, что может привести к различным нарушениям зрения (снижение остроты, спазм и уменьшение запаса аккомодации), помимо этого повышение зрительной нагрузки может приводить к общему утомлению, возникновению головной боли, ухудшению самочувствия.
* умственное перенапряжение, так как в процессе деятельности программист анализирует и перерабатывает огромное количество административных сетей, разрабатывает программные продукты, поддерживает технику в рабочем состоянии.

В целях выявления вредных и (или) опасных производственных факторов и осуществления мероприятий по приведению условий труда в соответствие государственными нормативными требованиями охраны трудаежегодно проводится аттестация рабочих мест по условиям труда – оценка условий труда на рабочих местах. Аттестации подлежат все имеющиеся в организации рабочие места.

При аттестации рабочих мест проводят оценку условий труда, гигиеническую оценку и оценку травмоопасности рабочих мест. При этом учитывают наличие средств коллективной защиты, обеспеченность работников средствами индивидуальной защиты и определяют эффективность этих средств.

Согласно санитарным правилам СП 2.2.3670–20 "Санитарно-эпидемиологические требования к условиям труда", дисплей должен располагаться на расстоянии 60-70 см, но не менее 50 см от глаз. Между боковыми поверхностями мониторов должно быть не менее 1,2 м. При использовании жидкокристаллических дисплеев на каждый компьютер должно приходиться не менее 4,5 м2 площади. На дисплей ПК не должен попадать прямой солнечный свет во избежание бликов и повышенной нагрузки на зрение.

В кабинете, где работает программист, работает еще 4 человека, а комплектующая оргтехника расположена достаточно далеко непосредственно от самого специалиста. Это обусловлено тем, что все копировальные аппараты, принтеры и факс в периоды своей работы выделяют не только вредные токсичные газы, но и шумы и вибрацию. Данные факторы оказывают гораздо меньшее негативное влияние, нежели излучение от компьютера, но также являются вредными и могут нанести вред здоровью сотрудника.

Для снижения напряжения с глаз, а также для снятия мышечного напряжения во время рабочего дня проводится проветривание в течение 10–15 минут каждые 1,5 часа.

**4.2 Расчет искусственного освещения рабочего места программиста в ООО «Смарт Трейд Маркетинг»**

Через зрительный анализатор человек получает около 80% из общего объема информации. Качество поступающей информации во многом зависит от освещения: неудовлетворительное количественно или качественно, оно не только утомляет зрение, но и вызывает утомление организма в целом. Кроме того, нерациональное освещение может явиться причиной травматизма: плохо освещенные опасные зоны, слепящие источники света и блики от них, резкие тени ухудшают видимость настолько, что вызывают полную потерю ориентировки работающих, снижают производительность труда и увеличивают брак продукции. Поэтому необходимо достаточное искусственное освещение для рабочего места.

Условия работы для расчета показателя освещенности рабочего места программиста являются:

* Помещение размером 30 квадратных метров;
* Высота помещения 3 метра, длина 5 метров, ширина 6 метров;
* Лампы люминесцентные (дневного света) в количестве 4 штук, новые;
* Крупногабаритная мебель отсутствует.

Расчет общего равномерного искусственного освещения горизонтальной рабочей поверхности выполняется методом коэффициента использования светового потока. Световой поток (лм) одной лампы рассчитывается по следующей формуле:

 (14)

где Ен - нормируемая минимальная допустимая освещенность по СП 1.3330.2016, для помещения 2000 лк;

S – площадь освещаемого помещения, м2;

z – коэффициент неравномерности освещения, z = 1,1;

K – коэффициент запаса, учитывающий снижение освещенности в процессе эксплуатации вследствие загрязнения и старения ламп и светильников, а также снижения отражающих свойств поверхностей помещения для кабинета будет равен 1,3;

Nc – число светильников в помещении;

 – коэффициент затенения, обычно 1;

 – коэффициент использования светового потока.

Коэффициент использования светового потока, давший название методу расчета, определяется по индексу помещения i в зависимости от типа светильника и коэффициентов отражения света от потолка, стен и пола:

 (15)

где А, В - длина и ширина помещения в плане, м;

Нс - высота подвеса светильников над рабочей поверхностью, м.

Согласно СП 1.3330.2016 величина полученного светового потока достаточно для данного помещения.

**4.3 Электробезопасность ООО «Смарт Трейд Маркетинг»**

При поступлении на работу сотрудник должен пройти инструктаж по технике безопасности и электробезопасности. Работника знакомят с основными правилами по технике безопасности, предлагают внимательно прочитать действующие на предприятии инструкции, поясняя при этом отдельные правила и требования.

Инструктаж по технике безопасности при выполнении конкретной работы проводит руководитель соответствующего производственного участка, показывая инструктируемому правильные безопасные приемы работы.

Повторный инструктаж проводится для рабочих независимо от их квалификации, стажа и опыта работы, не реже 1 раза в 6 месяцев по программе инструктажа на рабочем месте.

При нарушении работающими правил и инструкций по технике безопасности, технологической и производственной дисциплины, а также в случаях изменения технологического процесса или вида работы проводят дополнительные инструктажи.

Все инструктажи оформляются записями в специальном журнале с указанием номеров или шифров инструкций. Журнал о проведении инструктажа хранится у руководителя подразделения.

Работник на рабочем месте должен соблюдать общие меры электробезопасности. При этом запрещается:

* прикасаться к задней панели системного блока при включенном питании;
* допускать попадание влаги на поверхность системного блока, монитора, рабочую поверхность клавиатуры, дисководов, принтеров и др. устройств;
* производить самостоятельно вскрытие и ремонт оборудования;
* пользоваться неисправными электроприборами и электропроводкой;
* ремонтировать электроприборы самостоятельно;
* подвешивать электропровода на гвоздях, металлических и деревянных предметах, перекручивать провод, закладывать провод и шнуры на водопроводные трубы и батареи отопления, вешать что-либо на провода, вытягивать за шнур вилку из розетки;
* прикасаться одновременно к персональному компьютеру и к устройствам, имеющим соединение с землей (радиаторы отопления, водопроводные краны, трубы и т.п.);
* пользоваться самодельными электронагревательными приборами и электроприборами с открытой спиралью;
* наступать на переносимые электрические провода, лежащие на полу.

При работе с электроприборами и оргтехникой (персональные компьютеры, принтеры, сканеры, копировальные аппараты, факсы, бытовые электроприборы, приборы освещения):

* автоматические выключатели и электрические предохранители должны быть всегда исправны.
* Изоляция электропроводки, электроприборов, выключателей, штепсельных розеток, ламповых патронов и светильников, а также шнуров, с помощью которых включаются в электросеть электроприборы, должны быть в исправном состоянии.
* Электроприборы необходимо хранить в сухом месте, избегать резких колебаний температуры, вибрации, сотрясений.
* Для подогрева воды пользоваться сертифицированными электроприборами с закрытой спиралью и устройством автоматического отключения, с применением несгораемых подставок.

**4.4 Пожарная безопасность в ООО «Смарт Трейд Маркетинг»**

Инструктаж по пожарной безопасности проводится по программе, разработанной инженером по охране труда ООО «Смарт Трейд Маркетинг», с учетом требований стандартов, правил, норм и инструкций о мерах пожарной безопасности. Продолжительность инструктажа устанавливается в соответствии с утвержденной программой. Инструктаж по пожарной безопасности, как правило, проводится совместно с инструктажем по технике безопасности и в те же сроки.

Инструктаж по пожарной безопасности проходят все работники организации, независимо от их образования, стажа работы по данной профессии или должности, временные работники, командированные, обучающиеся и студенты, прибывшие на практику.

О проведении инструктажа по пожарной безопасности работник, проводивший инструктаж, делает запись в журнале учета проведения инструктажей по пожарной безопасности, с обязательной подписью инструктируемого и инструктирующего. При регистрации внепланового инструктажа по пожарной безопасности указывают причину его проведения.

Все работники организаций должны допускаться к работе только после прохождения противопожарного инструктажа, а при изменении специфики работы проходить дополнительное обучение предупреждению и тушению возможных пожаров в порядке, установленном руководителем.

На проведение первичного противопожарного инструктажа необходимо отводить не менее 1 ч. Инструктируемые должны ознакомиться:

* с действующими на объекте правилами пожарной безопасности и инструкциями;
* с производственными участками, наиболее опасными в пожарном отношении, где запрещается курить, применять открытый огонь;
* с возможными причинами возникновения пожара и мерами его предупреждения;
* с практическими действиями в случае возникновения пожара - вызов пожарной помощи, использование первичных средств пожаротушения, место расположения ближайшего телефона и ознакомление с правилами поведения в случае возникновения пожара, эвакуации людей и материальных ценностей.

При первичном инструктаже инструктирующий обязан рассказать о производственных установках с повышенной пожарной опасностью, мерах предотвращения пожаров и загораний, указать место курения, ознакомить вновь поступившего с имеющимися на объекте средствами пожаротушения, показать ближайший телефон и объяснить правила поведения в случае возникновения пожара.

Проведение противопожарного инструктажа в обязательном порядке должно сопровождаться практическим показом способов использования имеющихся на объекте средств пожаротушения (противогазы, респираторы, огнетушители и т.д.).

Соблюдение рассмотренных в данном разделе мероприятий по охране труда и технике безопасности в офисном здании позволяет снизить утомляемость и травматизм, повысить производительность труда, обеспечить комфортные условия трудовой деятельности специалиста, работающего в должности программиста.

**ЗАКЛЮЧЕНИЕ**

Целью работы было создание форума, на котором программисты смогут узнать решение интересующих их проблем. При этом форум должен иметь возможность локального развертывания на предприятии для обеспечения конфиденциальности информации. Также, интерфейс должен быть адаптивным и рабочим на необычных формах экранов мобильных устройств. Для достижения указанной цели был поставлен ряд задач, таких как:

* Разработка серверного приложения, обеспечивающего необходимый для форума функционал.
* Обеспечение совместимости с сервисами контейнеризации (напр. docker), для ускорения настройки и развертывания.
* Разработка клиента для мобильных устройств, который может быть запущен на разных ОС и обладающего привлекательным пользовательским интерфейсом.

При решении поставленных задач были изучены различные подходы к написанию программного обеспечения, изучены технологии написания приложений и их проектирования, а также обеспечена совместимость ПО с практически любым аппаратным обеспечением.

Таким образом, задачи решены в полном объеме, а цель – создать удобный IT-форум, состоящий из серверного приложения и мобильного клиента, достигнута.

**СПИСОК ИСПОЛЬЗОВАННЫХ ЛИТЕРАТУРНЫХ ИСТОЧНИКОВ**

1. Сырямкин, В.И. Информационные устройства и системы: учеб. пособие / А. А. Грешилов: Изд-во Томского Университета, 2016. – 521 с.
2. Исаев, Г. Н. Информационные системы в экономике: учеб. пособие / Г. Н. Исаев: Изд-во Омега-Л, 2015. – 462 с.
3. Грукул, В. И., Денищенко, Г.Н. Проектирование информационных систем: учеб. пособие / В. И. Грекул, Г.Н. Денищенко: Изд-во Бином. Лаборатория знаний, 2008. – 304 с.
4. Гвоздева, Т. В., Баллод, Б. А. Проектирование информационных систем: Реферат/ Изд-во: Феникс, 2009. – 508 с.
5. Информационные системы. Структура и классификация информационных систем: Электрон. дан. – М.: ЯКласс, 2018. – Режим доступа: http://www.rsl.ru, свободный. – Загл. с экрана.
6. Национальный исследовательский Мордовский государственный университет им. Н. П. Огарёва, Документирование и сертификация [Электронныйресурс] / Н. П. Огарёв -Проверено: 02.06.2016. <http://www.mrsu.ru>
7. Робинсон, С., Глинн. Д. C# для профессионалов. Комплект в 2-х томах: учеб. Пособие / С. Робинсон, Д. Глинн: Изд-во Лори, 2015. – 478 с.
8. Лок, Э. ASP.NET Core в действии : Учебное пособие / Э. Лок, изд-во ДМК пресс, 2021. – 906 с.

**ПРИЛОЖЕНИЕ А**

Исходный код программного обеспечения дипломного проекта

Ниже приведены сущности базы данных и интерфейсы сервисов, используемые в дипломном проекте.

using ForumBackend.Core.Model;

namespace ForumBackend.Core.Services;

public interface IAccessTokenGenerator

{

public string GetToken(ForumUser user);

}

using System.Security.Claims;

namespace ForumBackend.Core.Services;

public interface IAccessTokenValidator

{

ClaimsIdentity? ValidateToken(string token);

}

using ForumBackend.Core.DataTransfer;

namespace ForumBackend.Core.Services;

public interface IAuthenticationService

{

Task<ResultBase<string, string>> Login(string username, string password);

Task<ResultBase<string, string>> Regenerate(string token);

}

using ForumBackend.Core.DataTransfer;

using ForumBackend.Core.Model;

namespace ForumBackend.Core.Services;

public interface IUserService

{

Task<ResultBase<ForumUser, string>> RegisterUser(string nickname, string email, string password);

Task<ForumUser?> GetUserById(int id);

}

using ForumBackend.Core.DataTransfer;

using ForumBackend.Core.Model;

namespace ForumBackend.Core.Services;

public interface IPostService

{

Task<ResultBase<ForumPost, string>> CreatePost(string description, string text, int authorId);

Task<IEnumerable<ForumPost>> ListPosts(int start, int length, long? before = null);

Task<ForumPost?> PostDetails(int id);

}

using ForumBackend.Core.DataTransfer;

using ForumBackend.Core.Model;

namespace ForumBackend.Core.Services;

public interface ICommentService

{

Task<IEnumerable<ForumComment>> GetPostComments(int id, int start, int length);

Task<ResultBase<ForumComment, string>> CreateComment(int id, int authorId, string text);

}

using ForumBackend.Core.DataTransfer;

namespace ForumBackend.Core.Services;

public interface ILikeService

{

Task<ResultBase<string, string>> LikePost(int postId, int userId);

Task<ResultBase<string, string>> LikePostRetract(int postId, int userId);

Task<ResultBase<string, string>> LikeComment(int commentId, int userId);

Task<ResultBase<string, string>> LikeCommentRetract(int commentId, int userId);

}

using System.ComponentModel.DataAnnotations;

namespace ForumBackend.Core.Model;

public class ForumUser

{

[Key]

public int Id { get; set; }

[StringLength(127, MinimumLength = 3)]

[Required]

public string Nickname { get; set; } = string.Empty;

[StringLength(127, MinimumLength = 3)]

[Required]

public string Email { get; set; } = string.Empty;

#pragma warning disable

[Required]

public UserAuth UserAuth { get; set; }

public ICollection<PostLike> LikedPosts { get; set; }

public ICollection<CommentLike> LikedComments { get; set; }

#pragma warning restore

}

using System.ComponentModel.DataAnnotations;

using System.ComponentModel.DataAnnotations.Schema;

namespace ForumBackend.Core.Model;

public class UserAuth

{

[Key]

public int Id { get; set; }

[ForeignKey(nameof(User))]

[Required]

public int UserRef { get; set; }

#pragma warning disable

[Required]

public ForumUser User { get; set; }

#pragma warning restore

[Required]

public string Password { get; set; } = string.Empty;

}

using System.ComponentModel.DataAnnotations;

using System.ComponentModel.DataAnnotations.Schema;

namespace ForumBackend.Core.Model;

public class ForumPost

{

[Key]

public int Id { get; set; }

[Required]

[StringLength(150, MinimumLength = 1)]

public string Description { get; set; } = string.Empty;

[Required]

public string Text { get; set; } = string.Empty;

[Required]

public long CreatedAt { get; set; }

[Required]

[ForeignKey(nameof(Author))]

public int AuthorRef { get; set; }

#pragma warning disable

[Required]

public ForumUser Author { get; set; }

#pragma warning restore

#pragma warning disable

// ReSharper disable once CollectionNeverUpdated.Global

public ICollection<ForumComment> Comments { get; set; }

public ICollection<PostLike> LikedUsers { get; set; }

#pragma warning restore

}

using System.ComponentModel.DataAnnotations;

using System.ComponentModel.DataAnnotations.Schema;

namespace ForumBackend.Core.Model;

public class ForumComment

{

[Key]

public int Id { get; set; }

[Required]

[ForeignKey(nameof(Post))]

public int PostRef { get; set; }

#pragma warning disable

[Required]

public ForumPost Post { get; set; }

#pragma warning restore

[Required]

[ForeignKey(nameof(Author))]

public int AuthorRef { get; set; }

#pragma warning disable

[Required]

public ForumUser Author { get; set; }

#pragma warning restore

[Required]

public long CreatedAt { get; set; }

[Required]

public string Text { get; set; } = string.Empty;

#pragma warning disable

public ICollection<CommentLike> LikedUsers { get; set; }

#pragma warning restore

}

using System.ComponentModel.DataAnnotations;

using System.ComponentModel.DataAnnotations.Schema;

namespace ForumBackend.Core.Model;

public class PostLike

{

[Key]

public int Id { get; set; }

#pragma warning disable

[Required]

[ForeignKey(nameof(Post))]

public int PostRef { get; set; }

[Required]

public ForumPost Post { get; set; }

[Required]

[ForeignKey(nameof(User))]

public int UserRef { get; set; }

[Required]

public ForumUser User { get; set; }

#pragma warning restore

}

using System.ComponentModel.DataAnnotations;

using System.ComponentModel.DataAnnotations.Schema;

namespace ForumBackend.Core.Model;

public class CommentLike

{

[Key]

public int Id { get; set; }

#pragma warning disable

[Required]

[ForeignKey(nameof(Comment))]

public int CommentRef { get; set; }

[Required]

public ForumComment Comment { get; set; }

[Required]

[ForeignKey(nameof(User))]

public int UserRef { get; set; }

[Required]

public ForumUser User { get; set; }

#pragma warning restore

}

namespace ForumBackend.Core.DataTransfer;

public abstract class ResultBase<TResult, TError> where TResult : class where TError : class

{

    internal ResultBase()

    {

    }

    public abstract TResult? GetResult();

    public abstract TError? GetError();

    public abstract bool IsSuccess();

}

namespace ForumBackend.Core.DataTransfer;

public sealed class SuccessResult<TResult, TError> : ResultBase<TResult, TError> where TResult: class where TError: class

{

    public SuccessResult(TResult result)

    {

        Result = result;

    }

    public TResult Result { get; }

    public override TResult GetResult()

    {

        return Result;

    }

    public override TError? GetError()

    {

        return null;

    }

    public override bool IsSuccess()

    {

        return true;

    }

}

namespace ForumBackend.Core.DataTransfer;

public sealed class ErrorResult<TResult, TError> : ResultBase<TResult, TError> where TResult: class where TError: class

{

    public ErrorResult(TError error)

    {

        Error = error;

    }

    public TError Error {get;}

    public override TResult? GetResult()

    {

        return null;

    }

    public override TError GetError()

    {

        return Error;

    }

    public override bool IsSuccess()

    {

        return false;

    }

}

Полный исходный код веб-сервиса и клиента доступны по ссылке: <https://github.com/Antttezy>

**ПРИЛОЖЕНИЕ Б**

Проектирование интерфейса мобильного приложения

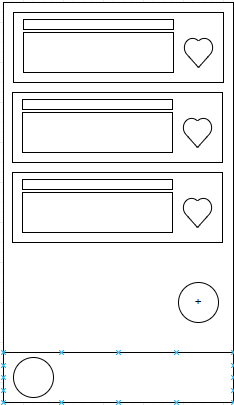


Рисунок Б-1 – разметка интерфейса списка вопросов

Изображение выглядит как текст

Автоматически созданное описание

Рисунок Б-2 – разметка интерфейса окна входа

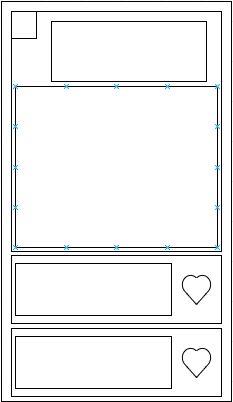


Рисунок Б-3 – разметка интерфейса ответов на вопрос

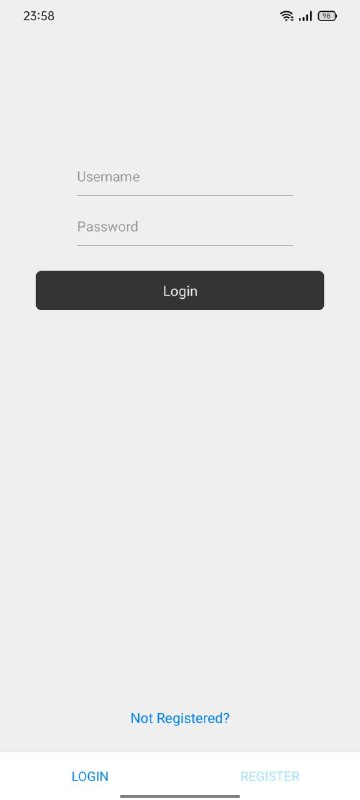


Рисунок Б-4 – окно входа в систему

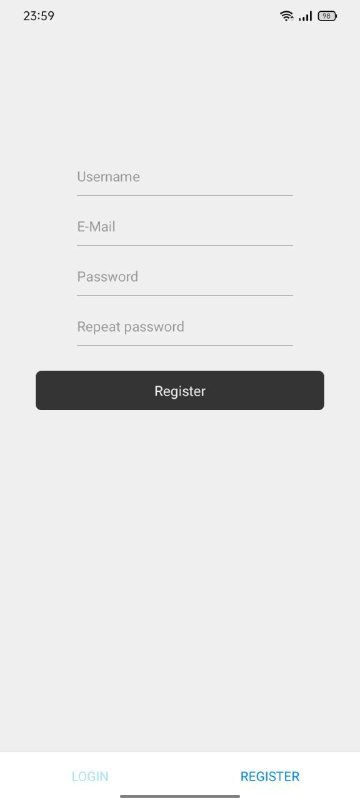


Рисунок Б-5 – окно регистрации

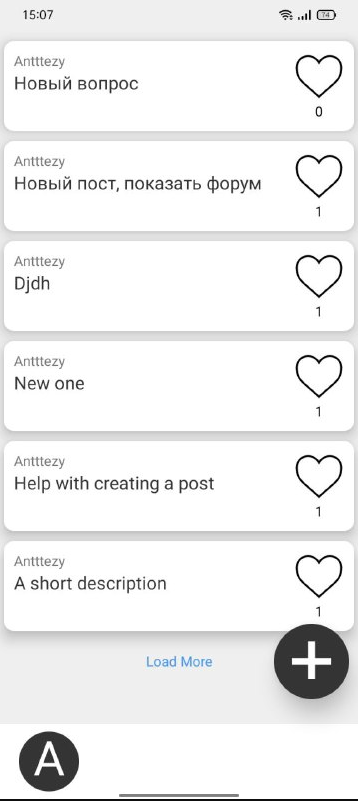


Рисунок Б-6 – список последних вопросов

Изображение выглядит как текст

Автоматически созданное описание

Рисунок Б-7 – оценка вопроса

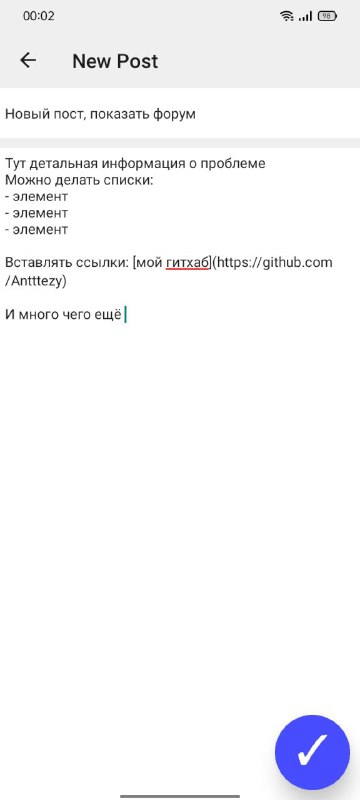


Рисунок Б-8 – интерфейс создания вопроса

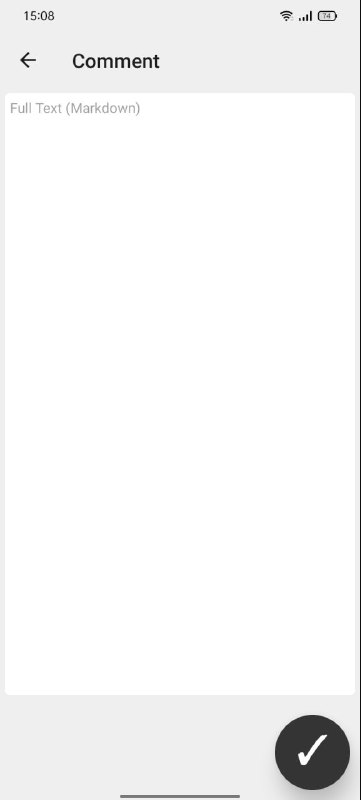


Рисунок Б-10 – Интерфейс создания комментария

Изображение выглядит как текст

Автоматически созданное описание

Рисунок Б-11 – созданный комментарий