Частное учреждение образования

«Колледж бизнеса и права»

ВЕБ-ПРИЛОЖЕНИЕ ДЛЯ СОВМЕСТНОГО

ПРОСМОТРА ФИЛЬМОВ «ROVIE»

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

по учебному предмету «Конструирование программ  
 и языки программирования»

КП Т.095009.401

Руководитель проекта (В.Ю.Михалевич)

Учащаяся (Д.С.Капур)

2023

**Содержание**

Изм.

Изм.

Изм.

Изм.

Изм.

Изм.

Изм.

Изм.

Изм.

Изм.

Изм.

Изм.

Изм.

Изм.

Изм.

Изм.

Лист

Лист

Лист

Лист

Лист

Лист

Лист

Лист

Лист

Лист

Лист

Лист

Лист

Лист

Лист

Лист

№ докум.

№ докум.

№ докум.

№ докум.

№ докум.

№ докум.

№ докум.

№ докум.

№ докум.

№ докум.

№ докум.

№ докум.

№ докум.

№ докум.

№ докум.

№ докум.

Подпись

Подпись

Подпись

Подпись

Подпись

Подпись

Подпись

Подпись

Подпись

Подпись

Подпись

Подпись

Подпись

Подпись

Подпись

Подпись

Дата

Дата

Дата

Дата

Дата

Дата

Дата

Дата

Дата

Дата

Дата

Дата

Дата

Дата

Дата

Дата

Лист

Лист

Лист

Лист

Лист

Лист

Лист

Лист

Лист

Лист

Лист

Лист

Лист

Лист

Лист

Лист

3

3

3

3

3

3

3

3

3

3

3

3

3

3

3

3

КП Т.095009.401 ПЗ

КП Т.716007.401 ПЗ

КП Т.716009.401 ПЗ

КП Т.716007.401 ПЗ

КП Т.095009.401 ПЗ

КП Т.716007.401 ПЗ

КП Т.716009.401 ПЗ

КП Т.716007.401 ПЗ

КП Т.095009.401 ПЗ

КП Т.716007.401 ПЗ

КП Т.716009.401 ПЗ

КП Т.716007.401 ПЗ

КП Т.095009.401 ПЗ

КП Т.716007.401 ПЗ

КП Т.716009.401 ПЗ

КП Т.716007.401 ПЗ

Разраб.

Разраб.

Разраб.

Разраб.

Разраб.

Разраб.

Разраб.

Разраб.

Разраб.

Разраб.

Разраб.

Разраб.

Разраб.

Разраб.

Разраб.

Разраб.

*Капур Д.С.*

*Капур Д.С.*

*Капур Д.С.*

*Капур Д.С.*

Провер.

Провер.

Провер.

Провер.

Провер.

Провер.

Провер.

Провер.

Провер.

Провер.

Провер.

Провер.

Провер.

Провер.

Провер.

Провер.

Михалевич В.Ю.

Банцевич. С.В

Банцевич. С.В

Банцевич. С.В

Михалевич В.Ю.

Банцевич. С.В

Банцевич. С.В

Банцевич. С.В

Михалевич В.Ю.

Банцевич. С.В

Банцевич. С.В

Банцевич. С.В

Михалевич В.Ю.

Банцевич. С.В

Банцевич. С.В

Банцевич. С.В

Т. контр.

Т. контр.

Т. контр.

Т. контр.

Т. контр.

Т. контр.

Т. контр.

Т. контр.

Т. контр.

Т. контр.

Т. контр.

Т. контр.

Т. контр.

Т. контр.

Т. контр.

Т. контр.

Н. контр.

Н. контр.

Н. контр.

Н. контр.

Н. контр.

Н. контр.

Н. контр.

Н. контр.

Н. контр.

Н. контр.

Н. контр.

Н. контр.

Н. контр.

Н. контр.

Н. контр.

Н. контр.

Утверд.

Утверд.

Утверд.

Утверд.

Утверд.

Утверд.

Утверд.

Утверд.

Утверд.

Утверд.

Утверд.

Утверд.

Утверд.

Утверд.

Утверд.

Утверд.

ВЕБ-ПРИЛОЖЕНИЕ ДЛЯ СОВМЕСТНОГО ПРОСМОТРА ФИЛЬМОВ «ROVIE»

ПРОГРАММНАЯ РЕАЛИЗАЦИЯ ХОРРОР ИГРЫ «ЧУЖАК»

ПРОГРАММНАЯ РЕАЛИЗАЦИЯ ХОРРОР ИГРЫ «ЧУЖАК»

ПРОГРАММНАЯ РЕАЛИЗАЦИЯ ХОРРОР ИГРЫ «ЧУЖАК»

Лит.

У

Лит.

У

Лит.

У

Лит.

У

Лит.

У

Лит.

У

Лит.

У

Лит.

У

Лит.

У

Лит.

У

Лит.

У

Лит.

У

Лит.

У

Лит.

У

Лит.

У

Лит.

У

Листов

Листов

Листов

Листов

Листов

Листов

Листов

Листов

Листов

Листов

Листов

Листов

Листов

Листов

Листов

Листов

60

50

55

50

60

50

55

50

60

50

55

50

60

50

55

50

КБП

КБП

КБП

КБП

КБП

КБП

КБП

КБП

КБП

КБП

КБП

КБП

КБП

КБП

КБП

КБП

[Введение 3](#_Toc135645275)

[1 Описание задачи 4](#_Toc135645276)

[1.1 Анализ предметной области 4](#_Toc135645277)

[1.2 Постановка задачи 4](#_Toc135645278)

[2 Проектирование веб-приложения 5](#_Toc135645279)

[2.1 Проектирование модели 5](#_Toc135645280)

[2.2 Требования к веб-приложению 5](#_Toc135645281)

[2.3 Структура веб-приложения 5](#_Toc135645282)

[2.4 Проектирование макета веб-приложения 5](#_Toc135645283)

[2.5 Программно-технические средства для разработки приложения 6](#_Toc135645284)

[2.6 Программно-технические средства для разработки приложения 6](#_Toc135645285)

[2.7 Программно-технические средства для разработки приложения 6](#_Toc135645286)

[3 Реализация веб-приложения 8](#_Toc135645287)

[3.1 Описание разделов веб-приложения 8](#_Toc135645288)

[3.2 Разработка административной части приложения 9](#_Toc135645289)

[3.3 Разработка клиентской части приложения 9](#_Toc135645290)

[3.4 Описание используемых функций и процедур 9](#_Toc135645291)

[3.5 Функциональное тестирование 10](#_Toc135645292)

[4 Применение 13](#_Toc135645293)

[4.1 Назначение веб-приложения 13](#_Toc135645294)

[4.2 Программно-аппаратное обеспечение сервера и клиента 14](#_Toc135645295)

[Заключение 15](#_Toc135645296)

[Список информационных источников 16](#_Toc135645297)

[Приложение А Текст программных модулей 17](#_Toc135645298)

[Приложение Б Результаты работы программы 18](#_Toc135645299)

**Введение**

Нынешнее развитие технологий позволяет решать абсолютно не ограниченный круг задач. Интернет – это интерактивный канал связи, что позволяет передавать информацию всевозможных типов: аудио, видео, текстовую и т.д.

Приложения для совместного просмотра фильмов набирают все больший актив в сетевом пространстве. Они представляют собой стриминговые сервисы, позволяющие просматривать какой-либо видео контент онлайн с друзьями в режиме реального времени. Существует множество разнообразных вариантов. Это может быть отдельное приложение или функция в популярных социальных сетях. Кроме того, некоторые приложения предлагают не только просмотр видео, но и игры, голосовые чаты и дополнительные функции. В таком разнообразии каждый может найти подходящий вариант для себя и своей компании.

Стриминговые сервисы становятся все более популярными, а посещаемость кинотеатров резко упала. Эти две тенденции связаны, и видно, как формируется будущее кино. Одним из наиболее очевидных преимуществ потоковых сервисов перед кинотеатрами является качество просмотра. При походе в кинотеатр, приходится сталкиваться с несколькими проблемами, которые делают впечатления от просмотра далеко не идеальными. Во-первых, нет возможности контролировать звук в кинотеатре, поэтому для некоторых людей это может быть слишком громким и раздражающим для просмотра. Однако, при просмотре фильма онлайн, можно контролировать громкость и качество звука. Кроме того, можно использовать наушники или хороший динамик, чтобы послушать фильм и сделать его более приятным. А при просмотре фильма на любом потоковом сервисе, есть возможность управлять им, не выходя из дома, на своих условиях.

Задачей курсового проектирования является реализация веб-приложения для совместного просмотра фильмов. Пользователь может зарегистрироваться и войти в веб приложение с помощью электронной почты и пароля. В кабинете пользователя можно изменять личные данные, создавать и публиковать виртуальные комнаты. При создании виртуальной комнаты пользователь имеет возможность в поиске пользователей для их присоединения к приватному пространству. Для обеспечения приватности комнат будут отсылаться коды доступа на почты присоединенных пользователей, в последующем с помощью которых они смогут войти в необходимую комнату. В виртуальной комнате будет происходить стриминг демонстрации экрана автора комнаты и возможность отправки сообщений и голосовых переговоров пользователей с помощью чата.

Решение поставленных задач отражено в пояснительной записке, которая состоит из четырех разделов и содержит необходимую и достаточную информацию по использованию данного программного средства.

В первом разделе «Описание задачи» анализируется предметная область, указываются функции для автоматизации, а также описываются существенные аналоги.

Во втором разделе «Проектирование веб-приложения», описаны проектируемые модели, установлены различные требования к веб-приложению. В подпункте «Структура веб-приложения» представление логической структуры и структурной схемы. Также втором разделе располагается графическое представление макета веб-приложения, описание программно-технических средств, необходимых для разработки проекта, информация о защите и сохранности данных, организация и ведение информационной базы.

В третьем разделе «Реализация веб-приложения» можно найти описание разделов, административной и клиентской части веб-приложения, описание используемых функций и процедур и функциональное тестирование.

В четвертом разделе «Применение» можно узнать информацию о назначении веб-приложения и требования к необходимому для веб-приложения программно-аппаратному обеспечению сервера и клиента для корректной работы приложения.

В последнем разделе «Заключение» описано как выполнена поставленная задача, степень соответствия проектных решений заданию и причины несоответствия.

В приложении А представлен код программных модулей, в веб-приложении, в приложении Б – результаты работы программы.

Графическая часть представлена диаграммами вариантов использования, классов, деятельности.

**1 Описание задачи**

* 1. **Анализ предметной области**

Примерно за десять лет цифровизация радикально изменила потребление фильмов вне кинотеатра. Смартфоны и планшеты вместе с улучшенной инфраструктурой широкополосного доступа в интернет и другими технологиями разработки позволяют смотреть фильмы на ходу, расширяя понимание «домашнего видео», чтобы охватить просмотр фильмов везде, кроме кинотеатров. Следовательно, для большинства из потребителей арена домашнего потребления видео расширилась как во времени, так и в пространстве, и стриминговые онлайн сервисы быстро превратились в ведущие рыночные каналы, в то время как прежние доминирующие каналы, такие как физические форматы «DVD» и «Blu-ray», оказались на обочине. А требования физического дистанцирования во время пандемии «Covid-19» стимулировали рост потокового видео.

«Rovie» – программа для синхронизации удалённого управления воспроизведением мультимедиа с демонстрацией экрана. Суть работы программы заключается в следующем: все проигрыватели на локальных компьютерах подключаются к одному серверу.

Комната – это отдельный канал связи, доступный лишь для пользователей с определенным уровнем доступа.

Стриминг – это метод непрерывной передачи цифровых медиафайлов – обычно аудио и видео – из одной точки в другую.

Чат – средство обмена текстовыми и голосовыми сообщениями по компьютерной сети в режиме реального времени, а также программное обеспечение, позволяющее организовывать такое общение. Процесс обмена сообщениями в таком приложении очень похож на обмен обычными SMS-сообщениями на телефоне. Вы выбираете контакт, которому хотите отправить сообщение, пишите текст и нажимаете на кнопку «Отправить».

Чтобы пользоваться приложением, нужно пройти регистрацию – процесс, где пользователь вводит почту и пароль, после чего ему предоставляются права на пользование веб-приложением. После регистрации посетитель становится пользователем и может зайти в веб-приложение после того, как пройдет авторизацию.

Авторизация – предоставление определённому лицу или [группе лиц](https://ru.wikipedia.org/wiki/%D0%93%D1%80%D1%83%D0%BF%D0%BF%D0%B0_%D0%BB%D0%B8%D1%86) прав на выполнение определённых действий; а также процесс проверки (подтверждения) данных прав при попытке выполнения этих действий.

Пойдя авторизацию, пользователю демонстрируется главная страница с активными комнатами. К ним можно подключиться, зная уникальный ключ, который необходимо ввести в форму, отображаемую после нажатия на одну из комнат.

В настройках пользователь имеет возможность изменения личных данных, удаление аккаунта, добавление номера телефона к личным данным.

На странице создания комнаты находится форма, где можно установить имя комнате и присоединить пользователей. После нажатия на «Создать» комната добавляется на главную страницу, а присоединённым пользователям отправляются ключи доступа на почту.

В комнате находится мультимедиа со стримингом демонстрации экрана автора комнаты и чат для организации обмена текстовыми и голосовыми сообщениями между присоединенными пользователями.

Чтобы побороть избыточность данных в базе принято решение удалять созданную комнату после выхода из нее автора этой комнаты.

В целях удобства использования и защиты аккаунта пользователи имеют возможность изменять пароль и почтовый адрес.

Для обеспечения приватности комнат создана система доступа к ним с помощью уникальных ключей доступа, которые пользователи получают на почту после создания комнаты и с условием, что они присоединены к ней.

«Rovie» подойдет русскоговорящим людям, обладающим низкой компьютерной грамотностью, низким системным опытом, низким опытом работы с подобными программами, уровнем чтения 1-11 классы. Также программной реализацией могут пользоваться люди любого пола, возраста от 6 лет. При наличии физических недостатков, таких как: моторная недостаточность, дефекты зрения, слуха – использование «Rovie» будет вызывать неудобства. А также программа может не подходить людям с острыми психическими особенностями, такими как: различные виды эпилепсии, шизофрении и диссоциативных расстройств личности.

* 1. **Постановка задачи**

На основе проанализированных бизнес-процессов можно выделить необходимые для автоматизации функции:

* регистрация и вход в веб-приложение;
* настройки с возможностью изменения личных данных пользователя и удаления аккаунта;
* публикация активных комнат на главной странице c реализацией их поиска;
* форма для создания приватной комнаты с возможностью присоединения пользователей после их выбора в результате поиска по почтовому адресу;
* отсылка кодов доступа после создания комнаты;
* реализация в комнате мультимедиа со стримингом демонстрации экрана автора комнаты и чата для организации обмена текстовыми и голосовыми сообщениями между присоединенными пользователями;
* обеспечить хранение необходимой информации в базе данных.

Самым ярким аналогом можно выделить «Twitch» – самая популярная платформа для стриминга. Особенно здесь в тренде трансляции игр – как стримы от геймеров, так и крупнейшие мировые киберсоревнования. «Twitch» считается крупнейшей площадкой для киберспортсменов. В отличии от нее «» будет иметь не открытые трансляции, а трансляции в приватных комнатах, что не дает смысла в популяризации своего аккаунта, потому что на стримы может приходить только ограниченное количество людей. Также «Rovie» не имеет возможности подписки на какие-либо аккаунты, не имеет возможности оставлять лайки, не имеет алгоритмы для определения самых популярных стримов и не имеет возможности оставлять донаты.

Также нельзя не упомянуть «Watchers» – сервис для совместного онлайн-просмотра фильмов с друзьями или даже знаменитостями. Само приложение направлено на то, чтобы избавить многих людей от «Цифрового одиночества». В отличии от него «Rovie» не может создавать публичные комнаты, нельзя добавить какого-либо бота и нет алгоритмов для выведения в рекомендации самых популярных комнат. Отличие еще и в том, что, создав комнату для просмотра фильмов с друзьями, каждый из участников имеет равные права – любой может отключить звук или поставить видео на паузу, как это происходило бы при реальной встрече.

**2 Проектирование веб-приложения**

**2.1 Проектирование модели**

Главной целью проектирования моделей является отображение функциональной структуры объекта, то есть производимые ими действия и связи между этими действиями. Наиболее распространенным средством моделирования функциональных требований являются диаграмма вариантов использования (UML).

Назначение данной диаграммы состоит в следующем: проектируемая информационная система представляется в форме так называемых вариантов использования, с которыми взаимодействуют внешние сущности или актеры. При этом актером или действующим лицом называется любой объект, субъект или система, взаимодействующая с моделируемой бизнес-системой извне. Вариант использования служит для описания сервисов, которые система предоставляет актеру.

В самом общем случае, диаграмма вариантов использования представляет собой граф специального вида, который является графической нотацией для представления конкретных вариантов использования*,*актеров и отношений между этими элементами.

Теперь определим логику диаграммы вариантов использования. При открытии веб-приложения «Rovie» посетитель должен для начала прости регистрацию, а после он сможет войти в созданный аккаунт. В последующем при открытии веб-приложения нет надобности в повторной регистрации. После входа в аккаунт пользователь может выйти из аккаунта, изменять личные данные, создавать комнаты, присоединяя к ним других пользователей и публикуя на главной странице, а таже можно войти как в собственно созданную комнату или в какую-либо другую, имея ключ доступа. Войдя, автор комнаты демонстрирует свой экран в режиме реального времени, а при входе присоединенных пользователей они могут просматривать эту трансляцию. В виртуальной комнате у всех есть возможность в обмене сообщениями и выходе их комнаты.

Суть диаграммы вариантов использования заключается в том, что проектируемая система представляется в виде множества сущностей или актеров, взаимодействующих с системой с помощью, так называемых вариантов использования.

К «extend» связям относятся:

* обмен сообщениями;
* присоединение пользователей.

К «include» связям относится:

* вход и выход;
* вход в комнаты;
* создание и публикация комнаты;
* выход из комнаты;
* демонстрация экрана;
* просмотр трансляции.

Диаграмма вариантов использования располагается в ГЧ на листе 1.

Диаграммы классов (UML), отображаемая в ГЧ на листе 2, показывают набор классов, интерфейсов, а также их связи. Диаграммы этого вида чаще всего используются для моделирования объектно-ориентированных систем. Они предназначены для статического представления системы.

Большинство элементов UML имеют уникальную и прямую графическую нотацию, которая дает визуальное представление наиболее важных аспектов элемента с помощью сущностей, атрибутов и связей.

Атрибут (свойство) – это именованное свойство класса, описывающее диапазон значений, которые может принимать экземпляр атрибута. Класс может иметь любое число атрибутов или не иметь ни одного.

Операция (метод) – это реализация метода класса.

Абстрактные методы класса обозначаются курсивным шрифтом.

Статические методы класса обозначаются подчеркиванием.

Первая из связей – зависимость – семантически представляет собой связь между двумя элементами модели, в которой изменение одного элемента может привести к изменению семантики другого элемента.

Ассоциация – это структурная связь между элементами модели, которая описывает набор связей, существующих между объектами.

Агрегация – особая разновидность ассоциации, представляющая струк-турную связь целого с его частями.

Композиция — более строгий вариант агрегации. Известна также как агрегация по значению.

Четвертая связь – обобщение – выражает специализацию или наследо- вание, в котором специализированный элемент строится по спецификациям обобщенного элемента. Потомок разделяет структуру и поведение родителя.

Пятая связь – реализация – это семантическая связь между классами, когда один из них определяет соглашение, которого второй обязан придерживаться.

В диаграмме классов можно заметить, что у там присутствует один самый главный базовый класс «BaseApiController». От него наследуются все контроллеры, которые связывают наш вид веб-приложения, отвечающий за интерфейс и модель, реализующую базу данных. Также можно заметить множество классов, реализующих тип пользователя, сообщений чата, комнаты, ее создания, отправителя кодов на почту, различные классы сопутствующие аутентификации и авторизации пользователя и так далее.

Диаграмма деятельности – [UML](https://ru.wikipedia.org/wiki/UML)-диаграмма, на которой показаны действия. Под деятельностью понимается спецификация исполняемого поведения в виде координированного последовательного и параллельного выполнения подчинённых элементов – вложенных видов деятельности и отдельных действий, соединённых между собой потоками, которые идут от выходов одного узла ко входам другого.

Диаграммы деятельности состоят из ограниченного количества фигур, соединённых стрелками. Основные фигуры (узлы):

Прямоугольники с закруглениями действия (операция). Узел управления – это абстрактный узел действия, которое координирует потоки действий.

Ромбы – решения. Узел решения предназначен для определения правила ветвления и различных вариантов дальнейшего развития сценария.

Широкие полосы – начало (разветвление) и окончание (схождение) ветвления действий.

Чёрный круг – начало процесса (начальный узел).

Чёрный круг с обводкой – окончание процесса (финальный узел).

На диаграмме можно увидеть, что начала процесса идет со входа в веб-приложения. Далее можно пройти как «Вход в аккаунт», так и «Регистрацию», после которой мы возвращаемся ко входу. После входа в свой аккаунт пользователь может как выйти из аккаунта, так и войти или же создать комнату, присоединив пользователей и опубликовав ее. После входа в комнату пользователь для начала может выйти из комнаты, также может демонстрировать экран если он создатель комнаты и общаться с другими присоединёнными пользователями с помощью чата и голосовых сообщений.

Диаграмма располагается в ГЧ на листе 3.

**2.2 Требования к веб-приложению**

Стилистические оформление веб-приложения должно быть лёгким, а также должно использовать единую цветовую гамму.

Графический дизайн должен быть креативным, но в то же время лаконичным и не броским. Дизайн должен вызывать доверие пользователя, но не быть «внушительным». Использование в дизайне веб-приложения небольших флешь-элементов: в качестве баннеров, рекламных блоков - не приветствуется.

Основными шрифтовыми гарнитурами являются гарнитуры Arial. Размер шрифтов должен иметь логичную иерархию размеров, позволяющую комфортно воспринимать информацию.

Средствами просмотра веб-приложения является Google Chrome, Microsoft Internet Explorer, Яндекс, FireFox, Opera, Safari.

Контент должен быть краток и ясен. Приветствуется искажение, абстракция информации, которая понятна только людям отдельной специализации.

Система управления контентом должна содержать минимальное количество виджетов, соответствовать требованиям стилистики и графического дизайна веб-приложения.

Компоновка страниц должна иметь логическую иерархию и минимальное количество вкладок.

* 1. **Структура веб-приложения**

Интеллект-карта, представленная на рисунке 2.1, – это наглядное представление естественного течения мыслей от одной центральной идеи. Применяется для создания, развития и анализа идей. Очень удобный инструмент для графического представления информации.

В центре всех ментальных карт – главная идея. От нее отходят ключевые мысли, которые можно делить на подпункты до тех пор, пока вы не структурируете всю информацию.

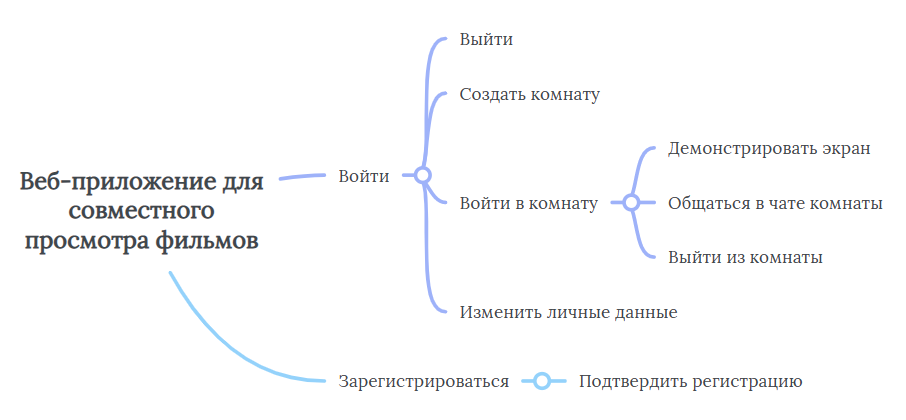


Рисунок 2.1 – Интеллект-карта

По ходу ведения проекта была выбрана ментальная карта, так как при ее построении активизируются различные способности нашего мышления. При составлении ветвей и ключевых слов мы используем иерархии, для картинок – визуализации и ассоциативное мышление, в целом используется пространственно-образное мышление. Все это активизирует память и позволяет запомнить как структуру данных, так и их важные аспекты, поэтому использование ментальных карт улучшает восприятие информации.

Структура веб приложения отображает все его страницы, расположенные определенным образом, для ознакомления пользователями.

Структура Веб-приложения:

«Главная страница» (панель навигации, добавить событие, список событий);

«Вход» (панель навигации, войти);

«Регистрация» (панель навигации, зарегистрироваться).

* структурная схема приложения в виде диаграммы fire flow с указанием взаимосвязи структурных блоков).

**2.4 Проектирование макета веб-приложения**

* обоснование типа дизайна веб-приложения (фиксированный (резиновый), фреймы, табличный, блочный);
* графическое представление макета страницы веб-приложения (главной и второстепенной);
* описание макета и графическое представление, выполненного в Figma (или иные сервисы для построения прототипа).

**2.5 Программно-технические средства для разработки приложения**

Инструментами для разработки данного программного средства будут являться:

* операционная система Windows 10;
* среда программирования Visual Studio;
* платформа ASP.NET;
* язык программирования C#;
* языки программирования HTML, CSS, JavaScript;
* база данных MS SQL;
* программа draw.io.

Windows 10 – операционная система для персональных компьютеров и рабочих станций, разработанная корпорацией Microsoft в рамках семейства Windows NT.

Microsoft Visual Studio написана на языках C++ и C# и предлагает интегрированный отладчик, который работает как на уровне исходного кода, так и на машинном уровне. В частности, IDE от Microsoft позволяет программистам совместно создавать и редактировать код, а также включает вспомогательные инструменты для написания кода, такие как подсветка синтаксиса, модели кодирования AI и инструменты анализа для отладки.

C# – это объектно-ориентированный язык программирования от Microsoft, который позволяет разработчикам создавать приложения, работающие на платформе .NET. Этот язык программирования берет свое начало в семействе языков программирования C и обладает многими теми же характеристиками, что и C и C++, а также Java и JavaScript. C# предоставляет базовый, читабельный язык для построения логики приложения, скрывая при этом большую часть сложности заложенных в язык возможностей. Он считается сильно типизированным языком, что означает, что каждая переменная и константа имеет тип, как и выражения, которые оцениваются в значения.

ASP.NET – бесплатная интернет-платформа для создания веб-сайтов и веб-приложений с помощью HTML, CSS и JavaScript. Она позволяет просто создавать веб-приложения, используя полнофункциональный язык программирования, такой как C# или VB.NET. Это очень популярная и часто используемый платформа. Она безопасна, быстра, проста в использовании и обеспечивает полный контроль над разработкой.

HTML применяется для разметки веб-страниц. Она нужна браузерам, которые преобразуют гипертекст и выводят на экран страницу в удобном для человека формате.

Назначение CSS – отделять то, что задает внешний вид страницы, от ее содержания. Если документ создан только с использованием HTML, то в нем определяется не только каждый элемент, но и способ его отображения (цвет, шрифт, положение блока и т. д.).

JavaScript –  [мультипарадигменный](https://ru.wikipedia.org/wiki/%D0%9C%D1%83%D0%BB%D1%8C%D1%82%D0%B8%D0%BF%D0%B0%D1%80%D0%B0%D0%B4%D0%B8%D0%B3%D0%BC%D0%B5%D0%BD%D0%BD%D0%BE%D0%B5_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5) [язык программирования](https://ru.wikipedia.org/wiki/%D0%AF%D0%B7%D1%8B%D0%BA_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D1%8F). Поддерживает объектно-ориентированный, императивный и функциональный стили. Его обычно используется как встраиваемый язык для программного доступа к объектам [приложений](https://ru.wikipedia.org/wiki/%D0%9A%D0%BE%D0%BC%D0%BF%D1%8C%D1%8E%D1%82%D0%B5%D1%80%D0%BD%D0%B0%D1%8F_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B0). Наиболее широкое применение находит в [браузерах](https://ru.wikipedia.org/wiki/%D0%91%D1%80%D0%B0%D1%83%D0%B7%D0%B5%D1%80) как язык сценариев для придания [интерактивности](https://ru.wikipedia.org/wiki/%D0%98%D0%BD%D1%82%D0%B5%D1%80%D0%B0%D0%BA%D1%82%D0%B8%D0%B2%D0%BD%D0%BE%D1%81%D1%82%D1%8C) [веб-страницам](https://ru.wikipedia.org/wiki/%D0%92%D0%B5%D0%B1-%D1%81%D1%82%D1%80%D0%B0%D0%BD%D0%B8%D1%86%D0%B0).

Microsoft SQL Server – это реляционная система управления базами данных, разработанная компанией Microsoft. Как сервер баз данных, он представляет собой программный продукт с основной функцией хранения и получения данных по запросам других программных приложений, которые могут работать либо на том же компьютере, либо на другом компьютере в сети (включая Интернет).

Программа draw.io – бесплатный онлайн-инструмент для создания диаграмм, который позволяет создавать блок-схемы, диаграммы, карты ума, организационные диаграммы и многое другое.

**2.6 Защита и сохранность данных**

Веб-приложение «Rovie» позволяет создавать комнаты, входить в комнаты, пользоваться всеми, сопутствующими комнате и личным данным, функциональностями только после входа в аккаунт. Система позволяет войти в аккаунт только зарегистрированным пользователям. Также комнаты имеют свою систему доступа, подразумевающую ввод уникальных ключей, что делает их доступными только при знании этого ключа.

**2.7 Организация и ведение информационной базы (модели)**

Организация данных подразумевает создание модели данных, главными элементами которой являются сущности и их связи.

Реляционная модель представляет собой совокупность данных, состоящих из набора двумерных таблиц. В теории множеств таблице соответствует термин отношение, физическим представлением которого является отсюда и название модели – реляционная.

Реляционная модель является удобной и наиболее привычной формой представления данных. При табличной организации данных отсутствует иерархия элементов.

Структура базы данных разрабатываемого программного средства включает одиннадцать таблиц. Их описание приводится в таблицах 2.1-2.11.

Таблица «Пользователи» хранит информацию о зарегистрированных пользователях и представлена в таблице 2.1.

Таблица 2.1 – Структура таблицы «Пользователи»

|  |  |  |  |
| --- | --- | --- | --- |
| Имя поля | Тип поля | Размер поля, байт | Описание поля |
| Идентификатор | text | 256 | Идентификатор пользователя |
| КодПодтвержденияПочты | integer | 50 | Код подтверждения электронной почты |
| ИмяПользователя | char | 256 | Логин пользователя |
| НормальноеИмяПользователя | char | 256 | Имя пользователя в фиксированном формате |
| Почта | char | 256 | Почта пользователя |
| НормальнаяПочта | char | 256 | Почта пользователя в фиксированном формате |
| ПочтаПодтверждена | boolean | 4 | Значения подтверждения почты |
| ХэшПароля | text | 256 | Пароль, прошедший через хэш-функцию |
| МеткаБезопансоти | text | 256 | Подпись авторства пользователя |
| МеткаПарралельности | text | 256 | Метка предотвращения конфликта параллельных обновлений |
| НомерТелефона | text | 256 | Номер телефона пользователя |
| НомерТелефонаПодтвержен | boolean | 4 | Значение подтверждения номера телефона |
| ДвухФакторВключен | boolean | 4 | Значение включение двухфакторной аутентификации |
| КонецБлокировки | timestamp | 4 | Время окончания блокировки аккаунта пользователя |
| БлокировкаВключена | boolean | 4 | Значение включения блокировки |
| КоличествоПроваловДоступа | integer | 8 | Количество неверных попыток в доступе к аккаунту |
| Имя | text | 256 | Имя пользователя |
| Фамилия | text | 256 | Фамилия пользователя |

Таблица «ПользовательскиеТребования», представленная в таблице 2.2, хранит информацию о требованиях к пользователям при использовании веб-приложения и связана с таблицей «Пользователи» с помощью идентификатора пользователя.

Таблица 2.2 – Структура таблицы «ПользовательскиеТребования»

|  |  |  |  |
| --- | --- | --- | --- |
| Имя поля | Тип поля | Размер поля, байт | Описание поля |
| Идентификатор | integer | 4 | Идентификатор требования |
| ИдентификаторПользователя | text | 256 | Внешний ключ, связывающий таблицу «Пользователи» |
| ТипТребования | char | 256 | Тип требования к пользователю |
| ЗначениеТребования | char | 256 | Описание требования к пользователю |

Таблица «ПользовательскиеРоли», представленная в таблице 2.3, связующая таблица пользователей и их ролей с помощью идентификаторов пользователя и роли.

Таблица 2.3 – Структура таблицы «ПользовательскиеРоли»

|  |  |  |  |
| --- | --- | --- | --- |
| Имя поля | Тип поля | Размер поля, байт | Описание поля |
| ИдентификаторПользователя | text | 256 | Ключ, связывающий таблицу «Пользователи» |
| ИдентификаторРоли | text | 256 | Ключ, связывающий таблицу «Роли» |

Таблица «Роли», представленная в таблице 2.4, хранит информацию о ролях пользователей веб-приложения.

Таблица 2.4 – Структура таблицы «Роли»

|  |  |  |  |
| --- | --- | --- | --- |
| Имя поля | Тип поля | Размер поля, байт | Описание поля |
| Идентификатор | text | 256 | Идентификатор роли |
| Имя | char | 256 | Имя роли |
| НормальноеИмя | char | 256 | Имя роли фиксированного формата |
| МеткаПарралельности | text | 256 | Метка предотвращения конфликта параллельных обновлений |

Таблица «РолевыеТребования», представленная в таблице 2.5, хранит информацию о требованиях к ролям при использовании веб-приложения и связана с таблицей «Роли» с помощью идентификатора роли.

Таблица 2.5 – Структура таблицы «РолевыеТребования»

|  |  |  |  |
| --- | --- | --- | --- |
| Имя поля | Тип поля | Размер поля, байт | Описание поля |
| Идентификатор | integer | 4 | Идентификатор требования |
| ИдентификаторРоли | text | 256 | Внешний ключ, связывающий таблицу «Роли» |
| ТипТребования | char | 256 | Тип требования к роли |
| ЗначениеТребования | char | 256 | Описание требования к роли |

Таблица «ПользовательскиеЛогины», представленная в таблице 2.6, хранит информацию о провайдерах пользователей веб-приложения и связанная к таблице «Пользователи» по идентификатору пользователя.

Таблица 2.6 – Структура таблицы «ПользовательскиеЛогины»

|  |  |  |  |
| --- | --- | --- | --- |
| Имя поля | Тип поля | Размер поля, байт | Описание поля |
| ПровайдерЛогина | text | 256 | Поставщик учетных данных |
| ПровайдерКлюча | text | 256 | Ключ поставщика |
| ИмяПровайдера | text | 256 | Отображаемое имя провайдера |
| Идентификатор пользователя | text | 256 | Внешний ключ к таблице «Пользователи» |

Таблица «ПользовательскиеТокены», представленная в таблице 2.7, хранит информацию о провайдерах пользователей веб-приложения и связанная к таблице «Пользователи» по идентификатору пользователя.

Таблица 2.7 – Структура таблицы «ПользовательскиеТокены»

|  |  |  |  |
| --- | --- | --- | --- |
| Имя поля | Тип поля | Размер поля, байт | Описание поля |
| ПровайдерЛогина | text | 256 | Поставщик учетных данных |
| Имя | text | 256 | Имя токена пользователя |
| Значение | text | 256 | Тело токена |
| Идентификатор пользователя | text | 256 | Ключ к таблице «Пользователи» |

Таблица «ФайлыДанных», представленная в таблице 2.8, хранит информацию о файлах с данными необходимыми для реализации функционала комнаты.

Таблица 2.8 – Структура таблицы «ФайлыДанных»

|  |  |  |  |
| --- | --- | --- | --- |
| Имя поля | Тип поля | Размер поля, байт | Описание поля |
| ИдентификаторФайлов | uuid | 16 | Идентификатор файлов |
| ИмяФайла | text | 256 | Название файла |
| Описание | text | 256 | Описнаие файла |
| Расширение | text | 256 | Расширение файла |

Таблица «Комнаты», представленная в таблице 2.9, хранит информацию о созданных пользователями комнатах и связанная к таблице «Пользователи» по идентификатору пользователя и к таблице «ФайлыДанных» по UUID файла.

Таблица 2.9 – Структура таблицы «Комнаты»

|  |  |  |  |
| --- | --- | --- | --- |
| Имя поля | Тип поля | Размер поля, байт | Описание поля |
| Идентификатор | integer | 4 | Идентификатор комнаты |
| Заголовок | text | 256 | Заголовок комнаты |
| Владелец | text | 256 | Внешний ключ, ссылающийся на таблицу «Пользователи» |
| НачалоВидео | boolean | 4 | Значение начала видео |
| Файл | uuid | 16 | Внешний ключ, ссылающийся на таблицу «ФайлыДанных» |

Таблица «СообщенияЧата», представленная в таблице 2.10, хранит информацию о созданных пользователями комнатах и связанная к таблице «Пользователи» по идентификатору пользователя и к таблице «Комнат» по идентификатору комнаты.

Таблица 2.10 – Структура таблицы «Комнаты»

|  |  |  |  |
| --- | --- | --- | --- |
| Имя поля | Тип поля | Размер поля, байт | Описание поля |
| Идентификатор | uuid | 16 | Идентификатор сообщения |
| Текст | text | 256 | Содержание сообщения |
| Дата | timestamp | 4 | Дата отправки сообщения |
| ИдентификаторПользователя | text | 256 | Внешний ключ, ссылающийся на таблицу «Пользователи» |
| ИдентификаторКомнаты | integer | 4 | Внешний ключ, ссылающийся на таблицу «Комнаты» |

Таблица «Участники», представленная в таблице 2.11, хранит информацию о присоединенных пользователях к комнате и связанная к таблице «Пользователи» по идентификатору пользователя и к таблице «Комнат» по идентификатору комнаты.

Таблица 2.11 – Структура таблицы «Комнаты»

|  |  |  |  |
| --- | --- | --- | --- |
| Имя поля | Тип поля | Размер поля, байт | Описание поля |
| ИдентификаторПользователя | text | 256 | Ключ, ссылающийся на таблицу «Пользователи» |
| ИдентификаторКомнаты | integer | 4 | Ключ, ссылающийся на таблицу «Комнаты» |
| КодПодтверждения | text | 256 | Код для входа в комнату |

**3 Реализация веб-приложения**

**3.1 Описание разделов веб-приложения**

* описать главную страницу (административной и клиентской части);
* описать навигационное меню (перечислить пункты меню и описать действия, выполняемые при выборе каждого пункта меню).

**3.2 Разработка административной части приложения**

* описание логической (карта) и физической структуры (каталог) серверной части веб-приложения;
* описание навигации (меню, гиперссылки).

**3.3 Разработка клиентской части приложения**

* описание логической (карта) и физической структуры (каталог) клиентской части веб-приложения;
* описание навигации (меню, гиперссылки).

**3.4 Описание используемых функций и процедур**

* описание функций и процедур пользователя в рамках среды разрабатываемого веб-приложения и закрепленных за определенными элементами управления (или обрабатываемыми событиями, страницами, ссылками);
* взаимосвязи функций;
* листинги всех модулей с комментариями и указанием элементов управления (событиями), которые инициируют их выполнение, должны быть представлены в приложении.

**3.5 Функциональное тестирование**

Необходимо провести тестирование разработанного программного средства на основании тест-кейсов. Тест-кейсы должны охватывать достаточный функционал разработанного программного средства и демонстрировать его работу на корректных и некорректных данных.

Проиллюстрировать результаты работы программного средства: скриншотами экранных форм результатов выполнения тестируемых функций и выходными формами документов.

Результаты работы программного средства могут быть представлены в приложении Б.

**4 Применение**

**4.1 Назначение веб-приложения**

Привести и обосновать:

* назначение веб-приложения;
* цель разработки;
* установку веб-приложения с описанием всего дерева файлов;
* запуск веб-приложения;
* область применения, класс решаемых задач;
* ограничения, накладываемые на область применения.

**4.2 Программно-аппаратное обеспечение сервера и клиента**

* Обосновать требования к необходимому для веб-приложения программно-аппаратному обеспечению сервера и клиента для корректной работы приложения.

**Заключение**

* какая задача была поставлена;
* как выполнена поставленная задача (на чем было разработано Веб-приложение, выполнены ли все задачи и т.д.);
* описание всех разработанных функций;
* преимущества и недостатки разработанного веб-приложения.

**Список информационных источников**

1. Багласова, Т.Г. Методические указания по оформлению курсовых и дипломных проектов / Т.Г. Багласова, К.О. Якимович. – Минск : КБП, 2023.
2. Михалевич В.Ю. Методические указания к курсовому проектированию по учебному предмету «Конструирование программ и языки программирования» / В.Ю.Михалевич. – Минск : КБП, 2023.
3. Программа и методика испытаний. Требования к содержанию, оформлению и контролю качества : ГОСТ 19.301-2000. – Введ. 01.09.2001. – Минск : Межгос. совет по стандартизации, метрологии и сертификации, 2000. – 14 с
4. .NET Framework [Электронный ресурс]. – Microsoft, 2020. – Режим доступа : https://docs.microsoft.com/en-us/dotnet/framework/. – Дата доступа : 01.09.2020.
5. Dr.Explain [Электронный ресурс]. – Dr.Explain, 2021. – Режим доступа : https://www.drexplain.ru/. – Дата доступа : 04.05.2021.
6. MS Office [Электронный ресурс]. – Microsoft, 2021. – Режим доступа : https://officeapplications.net/microsoft-word/. – Дата доступа : 03.05.2021.
7. Smart Install Maker [Электронный ресурс]. – InstallBuilders, 2020. – Режим доступа : https://www.ixbt.com/news/soft/index.shtml?10/52/79/. – Дата доступа : 05.05.2021.
8. Diagrams [Электронный ресурс]. – UML, 2021. – Режим доступа : https://www.diagrams.net/. – Дата доступа : 10.05.2021.
9. Visual Studio [Электронный ресурс]. – Microsoft, 2020. – Режим доступа : https://visualstudio.microsoft.com/vs/. – Дата доступа : 01.09.2020.
10. Что такое мессенджер и для чего он нужен + список самых популярных мессенджеров и рекомендации по выбору [Электронный ресурс]. – https://myrouble.ru/, 2020. – Режим доступа : https://myrouble.ru/messenger/. – Дата доступа : 12.12.2020.

Указываем все-все источники

**Приложение А**

(обязательное)

**Текст программных модулей**

10-15 страниц кода

using Messenger.Extentions;

using Messenger.Models;

using Messenger.ViewModels;

using System;

using System.Linq;

using System.Threading;

namespace Messenger

{

static class ServerManipulation

{

#region Properties

public static string Username { get; set; } = null;

public static string Key { get; set; } = null;

public static string Message { get; set; } = null;

public static object Data { get; set; } = null;

public static MessageModel[] NewMessages { get; set; } = null;

#endregion

#region Methods

public static bool Login(string username, string password)

{

Clear();

QueryResult data;

while (true)

{

if (!Server.ConnectToServer()) break;

if (!Server.SendMessage("[LOG]")) break;

if (!Server.ReadMessage()) break; // + {SYS}

if (!Server.SendMessage(username)) break;

if (!Server.ReadMessage()) break; // + salt

string hashPassword = Server.ComputePasswordHash(password, Server.LastQueryResult.Data);

if (!Server.SendMessage(hashPassword)) break; // - hash

if (!Server.ReadMessage()) break;

data = Server.LastQueryResult;

if (data.Type == QueryResultType.Data)

{

// {101} + ключ

Username = username;

Key = data.Data;

return true;

}

break;

}

Server.Close();

data = Server.LastQueryResult;

if (data.Type == QueryResultType.Message)

Message = data.Data;

return false;

}

**Приложение Б**

(обязательное)

**Результаты работы программы**

Номера рисунков и скриншоты, на которые ссылаетесь в пункте 3.5