**Министерство промышленности и торговли**

**Тверской области**

**ГБПОУ «Тверской колледж им. А.Н. Коняева»**

##### **КУРСОВОЙ ПРОЕКТ**

На тему *«Разработка веб-приложения “Messanger“»*

Студента(ки) группы 4-ИС1

Ф.И.О. Бобкова Даниила Игоревича

Специальность Информационные системы и программирование

Руководитель Ишкова Лариса Георгиевна

(фамилия, имя, отчество)

**Тверь**

**2022**

**Содержание**

[Введение 3](#_Toc152557360)

[1 Теоретический раздел (технологический раздел) 4](#_Toc152557361)

[1.1 Создание десктоп-мессенджера 4](#_Toc152557362)

[1.2 Понятие Front-end разработки 5](#_Toc152557363)

[1.3 Понятие Back-end разработки 6](#_Toc152557364)

[1.3.1 Схема работы 6](#_Toc152557365)

[1.3.2 Используемые языки программирования в бэкенде 7](#_Toc152557366)

# Введение

Тема курсового проекта - Разработка десктоп-приложения “Messanger”

В современном мире очень важна скорость передачи данных, поэтому практически мгновенная отправка сообщений облегчает жизни миллионам людей каждый час. Но современные мессенджеры и социальные сети перегружены обильным количество рекламы и спама, а также занимают большой объём памяти на устройствах пользователя.

Данное приложение создано для облегчения общения между людьми и заведением новых знакомств. Все что нужно сделать так это скачать и зарегистрироваться в нем.

Для использования этого приложения достаточно основных знаний об устройстве.

В рамках курсового проекта можно определить следующие направления развития:

Можно выделить следующие задачи развития курсового проекта:

* Создание пользовательского интерфейса;
* Проектирование и разработка основных функций приложения;
* Интеграция всех модулей в рабочую версию приложения.

# Теоретический раздел (технологический раздел)

## Создание десктоп-мессенджера

Создание десктоп мессенджера на основе C# и WPF, подобно созданию веб-сайта, включает несколько этапов и компонентов:

* Анализ и проектирование: Определение требований к мессенджеру, функциональности, интерфейса пользователя и основных характеристик приложения.
* Бекенд: это серверная часть, обрабатывающая логику бизнес-приложения и взаимодействующая с базой данных. На C# это может быть реализовано с использованием технологий, таких как ASP.NET, .NET Core, или другие фреймворки для создания веб-серверов и API.
* Фронтенд (WPF): WPF (Windows Presentation Foundation) - это технология для создания графического интерфейса пользователя в приложениях под Windows. Ваш десктопный мессенджер будет использовать WPF для создания пользовательского интерфейса. В этом случае, можно будет создавать окна, элементы управления, обработчики событий и визуальные элементы приложения, используя XAML (язык разметки) и C#.
* Коммуникация через сеть: для обмена сообщениями между пользователями через сеть понадобится реализация сетевой логики. Можете использовать TCP/IP или более высокоуровневые протоколы, такие как HTTP/WebSockets для обеспечения связи между клиентами и сервером.
* Безопасность: Реализация механизмов безопасности и аутентификации важна для защиты данных пользователей. Это может включать в себя шифрование сообщений, аутентификацию пользователей и управление доступом.
* Тестирование и отладка: после создания мессенджера необходимо тестировать его на различных устройствах и условиях сети, чтобы убедиться в его надежности и эффективности.

Все это неотъемлемые части создания любого десктоп приложения. Без выполнения этих пунктов невозможно представить корректно работающую программу.

## Понятие Front-end разработки

Frontend — это часть веб-разработки, которая отвечает за создание пользовательского интерфейса веб-сайтов или веб-приложений. Она занимается тем, что пользователь видит и с чем взаимодействует в браузере или приложении.

Frontend-разработчики заботятся о том, чтобы веб-сайт был удобен в использовании, отзывчивым (адаптивным под разные устройства и разрешения экранов) и обладал хорошей пользовательской интеракцией.

Основная цель frontend-разработки - обеспечить пользователей более интуитивным, эстетически приятным и удобным пользовательским опытом. Это включает в себя не только визуальную составляющую, но и обеспечение быстрой загрузки страниц, адаптивности под разные устройства, а также доступность для пользователей с ограниченными возможностями.

Frontend-разработчики тесно взаимодействуют с другими специалистами, такими как дизайнеры, backend-разработчики и специалисты по UX/UI, чтобы создать полноценный и функциональный веб-продукт. Их работа оказывает значительное влияние на восприятие и успешность сайта или приложения, так как пользовательский интерфейс является ключевым элементом взаимодействия пользователя с продуктом в онлайн-среде.

Фронтенд-разработчики используют различные средства для создания пользовательского интерфейса веб-приложений или сайтов. Вот некоторые из них:

* Языки разметки: HTML используется для структурирования контента на веб-страницах. CSS применяется для стилей и внешнего оформления элементов, делая страницу более привлекательной и удобной для использования.
* Языки программирования: JavaScript является основным языком для программирования веб-интерфейса. Он добавляет интерактивность, обработку событий и анимации на веб-страницах.
* Фреймворки и библиотеки: Фронтенд-разработчики часто используют фреймворки и библиотеки, такие как React.js, Angular, Vue.js, чтобы упростить процесс создания сложных пользовательских интерфейсов и повысить эффективность разработки.

## Понятие Back-end разработки

Backend (или серверная часть) отвечает за работу веб-сайта или веб-приложения на стороне сервера. Это то, что происходит за кулисами и не отображается непосредственно перед пользователем, в отличие от фронтенда, который представляет интерфейс для пользователей.

Backend включает в себя серверы, базы данных, приложения и всю логику, необходимую для работы веб-сайта или приложения. Здесь обрабатываются запросы, хранится информация, выполняются операции с данными и формируется ответ, который отправляется обратно пользователю через фронтенд.

### ****Схема работы****

Работа Backend и основные принципы функционирования веб-сайта можно сжать до трех ключевых этапов:

* Получение и обработка данных от пользователя, например, ввод поискового запроса.
* Обработка этих данных на сервере, включая поиск в базе данных и другие запросы.
* Представление полученного ответа от сервера в удобочитаемом виде для пользователя, т.е. вывод результатов.

Хотя взаимодействие с бэкендом кажется простым, его реализация может быть сложной. Это связано с тем, что различные веб-серверы используют разные языки программирования и могут работать с разными базами данных, интерфейсами и архитектурами. Необходимо тщательно объединить все эти компоненты и тестировать каждый шаг, иначе результат может быть нечитаемым.

### Используемые языки программирования в бэкенде

В бекенде (серверной части) веб-приложений и сайтов используются различные языки программирования, каждый из которых имеет свои особенности и подходит для определенных задач:

* Java: это универсальный язык программирования общего назначения. Он используется для создания масштабируемых, надежных и высокопроизводительных веб-приложений. Java подходит для корпоративных систем, распределенных приложений и работы с большими объемами данных. Он также используется в разработке Android-приложений.
* PHP: Это один из наиболее распространенных языков программирования для веб-разработки. PHP подходит для создания динамических веб-сайтов, обработки форм, управления базами данных и разработки веб-приложений. Он часто используется в сочетании с веб-фреймворками, такими как Laravel или Symfony.
* Python: Python хорошо подходит для разработки веб-приложений, имеет широкий спектр применения от простых скриптов до сложных веб-сервисов. Он удобен в использовании, имеет чистый синтаксис и обширные библиотеки, что делает его отличным выбором для быстрой разработки прототипов и сложных систем.
* Ruby: Язык Ruby известен своей простотой и удобством разработки. Он используется для построения гибких и легко поддерживаемых веб-приложений. Фреймворк Ruby on Rails является популярным выбором для быстрой разработки веб-приложений.
* JavaScript (Node.js): Node.js позволяет использовать JavaScript для серверной разработки. Он подходит для создания масштабируемых и высокопроизводительных веб-приложений. Node.js также хорошо совмещается с JavaScript на фронтенде, что позволяет разработчикам использовать один язык для всей системы.

Выбор языка программирования в бекенде зависит от требований проекта, сценариев использования, экспертизы разработчиков и специфики задачи. Кроме того, разработчики также могут использовать различные фреймворки, библиотеки и инструменты, чтобы облегчить процесс разработки и расширить функциональность приложения.

Также не стоит забывать о различных системах управления базами данных. Хотя MySQL является чаще всего используемым вариантом, на рынке существуют альтернативы, такие как PostgreSQL, SQLite, MongoDB и другие. Каждая из этих систем имеет свои особенности и специфику работы. Это предоставляет множество инструментов для выбора, но при этом их набор достаточно структурирован, что помогает сосредотачиваться на конкретных решениях, не рассеивая усилия. Обязанности и задачи backend-разработчика

Исходя из вышесказанного, backend-разработчику нужно выполнять следующие задачи:

* Разработка серверной логики: Создание и поддержка алгоритмов и процессов для корректной обработки запросов на сервере.
* Управление базами данных: Создание, настройка и оптимизация баз данных, включая выполнение запросов SQL и обеспечение безопасности данных.
* API и взаимодействие: Создание и поддержка API для обеспечения взаимодействия между серверной и клиентской частями приложения или различными компонентами системы.
* Безопасность приложения: Разработка мер безопасности, таких как аутентификация, авторизация и защита от взломов и утечек данных.
* Оптимизация производительности: Улучшение скорости и эффективности работы приложения, включая оптимизацию запросов к базам данных и использование кэширования данных.

Таким образом, backend-разработчику приходится поддерживать весь набор веб-систем, исключая разве что функционирование самого веб-сервера и каналов связи (это уже задача системного администратора).