# 4 РЕАЛИЗАЦИЯ

## 4.1 Выбор инструментальных средств, системного и дополнительного программного обеспечения

Приложение, разрабатывающееся в рамках дипломного проектирования, состоит из двух частей: бэкэнд и фронтэнд.

Для разработки бэкэнд части информационной системы был выбран язык C#. C# - это современный объектно-ориентированный и типобезопасный язык программирования. C# позволяет разработчикам создавать разные типы безопасных и надёжных приложений, выполняющихся в .NET. C# относится к широко известному семейству языков C, и покажется хорошо знакомым любому, кто работал с C, C++, Java или JavaScript.

C# - объектно-ориентированный, ориентированный на компоненты язык программирования. C# предоставляет языковые конструкции для непосредственной поддержки такой концепции работы. Благодаря этому C# подходит для создания и применения программных компонентов. С момента создания язык C# обогатился функциями для поддержки новых рабочих нагрузок и современными рекомендациями по разработке ПО. В основном C# - объектно-ориентированный язык. Вы определяете типы и их поведение.

Вот лишь несколько функций языка C#, которые позволяют создавать надёжные и устойчивые приложения. Сборка мусора автоматически освобождает память, занятую недостижимыми неиспользуемыми объектами. Типы, допускающие значение null, обеспечивают защиту от переменных, которые не ссылаются на выделенные объекты. Обработка исключений предоставляет структурированный и расширяемый подход к обнаружению ошибок и восстановлению после них. Лямбда-выражения поддерживают

Изм.

Лист

№ докум.

Подпись

Дата

Лист

УО «ВГТУ» ДП.006 1-40 05 01-01 ПЗ

Разраб.

Лапко М. Л.

Провер.

Дунина Е.Б.

Реценз.

Н. Контр.

Самусев А.М.

Утверд.

Казаков В.Е.

РЕАЛИЗАЦИЯ

Лит.

Листов

УО «ВГТУ» каф. ИСАП гр. Ит-6

приёмы функционального программирования. Синтаксис LINQ создаёт общий шаблон для работы с данными из любого источника. Поддержка языков для асинхронных операций предоставляет синтаксис для создания распределенных систем. В C# имеется Единая система типов. Все типы C#, включая типы-примитивы, такие, как int и double, наследуют от одного корневого типа object. Все типы используют общий набор операций, а значения любого типа можно хранить, передавать и обрабатывать схожим образом. Более того, C# поддерживает как определяемые пользователями ссылочные типы, так и типы значений. C# позволяет динамически выделять объекты и хранить упрощённые структуры в стеке. C# поддерживает универсальные методы и типы, обеспечивающие повышенную безопасность типов и производительность. C# предоставляет итераторы, которые позволяют разработчикам классов коллекций определять пользовательские варианты поведения для клиентского кода [7].

Язык C# практически универсален. Можно использовать его для создания любого ПО: продвинутых бизнес-приложений, видеоигр, функциональных веб-приложений, приложений для Windows, macOS, мобильных программ для iOS и Android.

С# популярен за счет своей «простоты». Простоты для современных программистов и больших команд разработчиков, чтобы те могли в сжатые сроки создавать функциональные и производительные приложения. Этому способствуют нетипичные конструкции языка и специфичный синтаксис, помогающий максимально органично реализовать намеченные функции.

Популярность языка - ещё одно значимое преимущество. Большое количество поклонников C# способствуют его развитию. Также это благоприятно влияет на рост числа вакансий, связанных с разработкой на языке Microsoft. Программисты, хорошо знакомые с С#, востребованы в индустрии, несмотря на их большое и постоянно увеличивающееся количество [8].

Изм.

Лист

№ докум.

Подпись

Дата

Лист

УО «ВГТУ» ДП.006 1-40 05 01-01 ПЗ

Также в языке присутствует обилие синтаксического сахара, который делает тяжёлую жизнь программиста капельку слаще. Вместо того чтобы писать 100500 строк кода, присутствует возможность использовать готовую конструкцию, а компилятор сделает всю остальную работу. Но некоторые такие конструкции являются не самыми оптимальными с точки зрения производительности. Но все это перекрывается за счёт удобочитаемости кода и высокой скоростью разработки [9].

Сама разработка бэкэнд части выполнена при помощи платформы ASP.NET Core.

ASP.NET Core - это веб-инфраструктура с открытым исходным кодом, оптимизированная для облачных вычислений, для разработки современных веб-приложений, которые можно разрабатывать и запускать на Windows, Linux и Mac. Он включает в себя инфраструктуру MVC, которая теперь объединяет функции MVC и веб-API в единую среду веб-программирования. Он был переработан с нуля, чтобы быть быстрым, гибким, современным и работать на разных платформах. В дальнейшем ASP.NET Core - это фреймворк, который можно использовать для веб-разработки с .NET. Также ASP.NET Core имеет схожие функции с MVC и веб-API [10].

Бэкэнд часть является веб-API. Платформа веб-API ASP.NET позволяет с легкостью создавать службы HTTP для широкого диапазона клиентов, включая браузеры и мобильные устройства. ASP.NET Web API - это идеальная платформа для сборки REST-приложений на базе .NET Framework [11].

Изм.

Лист

№ докум.

Подпись

Дата

Лист

УО «ВГТУ» ДП.006 1-40 05 01-01 ПЗ

Для написания приложения была выбрана среда программирования Microsoft Visual Studio.

Microsoft Visual Studio является средой программирования, разработанной компанией Microsoft. Эта среда позволяет создавать кроссплатформенные проекты на различных языках программирования, таких как Visual Basic, Visual C#, Visual C++, Visual F# и другие. Также она позволяет создавать программы, использующие в своей работе платформу .NET, которая позволяет использовать большой набор сервисов, реализующихся в виде промежуточного, независящего от базовой архитектуры, кода. Основной целью создания платформы .NET является возможность реализации разработчиками специальных сервисно-ориентированных программ, работающих на любых платформах.

MS Visual Studio позволяет разработчику иметь доступ к огромной коллекции различных функций, которые позволяют вести разработки для любой версии операционной системы семейства Windows, для интернет-приложений и мобильных приложений. Также среда программирования открывает широкие возможности в области облачных технологий. Эта среда открывает разработчику широкие возможности для реализации самых разных проектов, реализуя высокую производительность и независимость от особенностей оборудования.

Microsoft Visual Studio позволяет осуществлять проектирование программ, используя любые по размеру команды. Эта среда разработки предоставляет инструменты планирования для возможности внедрения методов последовательной разработки, а также для гибкого планирования. Используя весь спектр возможностей, предоставляемых MS Visual Studio, можно реализовать максимально полную систему, наиболее удачно спроектировать любую архитектуру. Таким образом Microsoft Visual Studio представляет собой передовую среду разработки [12].

Фронтэед часть разработана при помощи JavaScript-библиотеки React при поддержке TypeScript.

Изм.

Лист

№ докум.

Подпись

Дата

Лист

УО «ВГТУ» ДП.006 1-40 05 01-01 ПЗ

React - библиотека, написанная на JavaScript, которая используется для работы с интерфейсами. В 2011 году её начали использовать для социальной сети Facebook, а уже в 2013 году библиотеку выложили в открытый доступ, и энтузиасты со всего мира начали создавать инструменты для расширения её возможностей. Разработчики используют React, чтобы создавать интерфейсы, которые способны менять контент без перезагрузки страницы. Благодаря этому сайты или нативные приложения быстро отзываются на действия пользователей. Можно добавлять товары в корзину без перезагрузки страницы или заполнять формы без переадресации.

React считается самой популярной библиотекой в мире, написанной на JS. Её популярность обеспечивается тем, что с помощью компонента можно решать разные задачи. Особенно, если использовать дополнительные инструменты, которые интегрируются в проекты и открывают доступ к нестандартным возможностям [13].

React предпочтительнее использовать по следующим причинам:

* технология SPA (single-page application, по-русски: “разработка одностраничных приложений”): React поможет вам создать одностраничное приложение. С помощью ReactJS вы сможете изменять (управлять/манипулировать) контент всей страницы с минимальным кодом;
* декларативный подход: React использует декларативный дизайн со всем синтаксическим сахаром, что помогает написать поддерживаемый код высокого уровня. Вам просто нужно определить цель, и React будет обрабатывать инструкции JavaScript DOM с учетом ситуаций, в которых они используются;
* компонент-управляемый пользовательский интерфейс: React основан на компонентной концепции. Компоненты являются многоразовыми строительными блоками в пользовательском интерфейсе. С помощью ReactJS вы можете создать инкапсулированный компонент, который управляет своими данными, и избежать сценария влияния на другие состояния и действия компонентов в DOM-дереве. Это только одна из характеристик компонент-управляемого интерфейса. Он также помогает повторно использовать код, разделять ответственность и избегать повторения [14].

Изм.

Лист

№ докум.

Подпись

Дата

Лист

УО «ВГТУ» ДП.006 1-40 05 01-01 ПЗ

TypeScript (TS, TScript или «тайпскрипт») - это язык программирования для веб-разработки, основанный на JavaScript. Делает код понятнее и надежнее, добавляет статическую типизацию (переменные привязаны к конкретным типам данных), а также может быть скомпилирован в JavaScript. TypeScript используют фронтенд- и бэкенд-разработчики.

TypeScript добавляет в язык строгую типизацию. Каждой переменной при создании присваивается определенный тип (type) — стандартный или созданный самим разработчиком. Создать тип можно в пределах возможностей языка: например, число от 1 до 31 для записи дня в месяц или массив из двух элементов для записи координат.

TypeScript помогает сократить время на выявление и устранение багов, которые иногда сложно найти в динамической среде JavaScript. С помощью TypeScript можно написать более понятный и читаемый код, который максимально описывает предметную область. Таким образом архитектура становится более выраженной.

Изм.

Лист

№ докум.

Подпись

Дата

Лист

УО «ВГТУ» ДП.006 1-40 05 01-01 ПЗ

Код, написанный на TypeScript, не выполнится напрямую в браузере. Поэтому TS - не самостоятельный язык, а именно языковая надстройка над JS.

Для его работы нужен дополнительный этап - транспиляция, когда программное обеспечение преобразует написанный на TypeScript код в «чистый» JavaScript.

JS не требует установки в систему: его по умолчанию поддерживает любой браузер. А вот TypeScript понадобится установить, потому что для транспиляции необходим модуль tsc [15].

Код для фронтэнд части приложения написан при помощи редактора кода Visual Studio Code.

Visual Studio Code - это «бесплатный редактор, который помогает программисту писать код, помогает в отладке и исправлении кода с помощью метода IntelliSense». В обычных условиях это облегчает пользователю написание кода простым способом. Многие говорят, что это половина IDE и редактора, но решение остается за программистами. Visual Studio Code поддерживает несколько языков программирования, имеет кроссплатформенную поддержку, огромное количество расширений для упрощения создания кода, а также встроенную поддержку системы контроля версий Git.

Visual Studio Code поддерживает несколько языков программирования. Так что раньше программистам требовалась веб-поддержка: другой редактор для разных языков, но он имеет встроенную многоязычную поддержку. Это также означает, что он легко обнаруживает, если есть какая-либо ошибка или ссылка на другой язык, он сможет легко обнаружить её [16].

## 4.2 Описание реализации вариантов использования

Изм.

Лист

№ докум.

Подпись

Дата

Лист

УО «ВГТУ» ДП.006 1-40 05 01-01 ПЗ

Вариант использования - это связный блок функциональности, которую предоставляет классификатор (система, подсистема или класс). Этот блок описывает последовательность сообщений, которыми обменивается система и один или несколько внешних пользователей (актантов), а также действия, осуществляемые при этом системой.

Вариант использования служит для определения некой части поведения классификатора (которым можно также считать подсистему и даже всю систему целиком), без указания на его внутреннюю структуру. Каждый вариант использования описывает некую услугу, которую предоставляет своим пользователям классификатор. Иначе говоря, это некоторый способ использования классификатора, который виден со стороны. Вариант использования описывает всю последовательность сообщений, которую начинает пользователь (и модели - актант), в терминах взаимодействия между пользователем и классификатором, включая ответы классификатора. К взаимодействию относятся только коммуникации между системой и актантами. Внутреннее поведение и реализация скрыты. Все множество вариантов использования какого-либо классификатора или системы разделяет и полностью описывает его поведение. Каждый вариант использования представляет собой некую разумную долю функциональности, которая доступна пользователям. Обратите внимание, что под термином пользователь следует понимать не только людей, но и компьютеры, а также прочие объекты. Актант представляет собой некую идеализацию намерений пользователя, а не самого этого пользователя. Один реальный пользователь может соответствовать нескольким актантам, а один актант может представлять одно и то же намерение сразу нескольких пользователей.

Изм.

Лист

№ докум.

Подпись

Дата

Лист

УО «ВГТУ» ДП.006 1-40 05 01-01 ПЗ

Вариант использования включает в себя описание основного поведения, осуществляемого в ответ на запрос пользователя, а также все возможные варианты этого поведения, например альтернативные последовательности, исключительное поведение и обработка ошибок. Для большего удобства варианты использования можно группировать в пакеты [17].

Диаграмма вариантов использования приложения пользователями представлена ниже (рисунок 4.1).

Изм.

Лист

№ докум.

Подпись

Дата

Лист

УО «ВГТУ» ДП.006 1-40 05 01-01 ПЗ

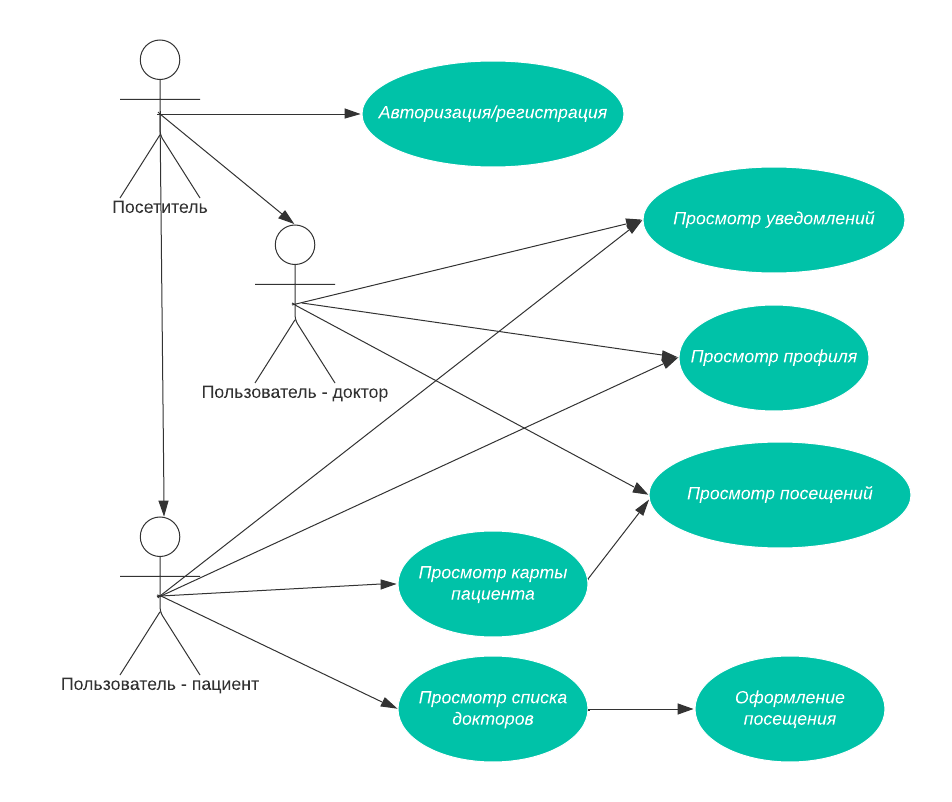


Рисунок 4.1 - Диаграмма вариантов использования

На схеме изображено 3 типа пользователей: анонимный посетитель, доктор, пациент. После авторизации или регистрации у анонимного пользователя появляется роль доктора или пациента. В зависимости от роли пользователям доступен различный функционал интерфейса, тем не менее, у всех типов пользователей имеются несколько одинаковых возможностей, например просмотр и редактирование своего профиля или просмотр уведомлений.

Из некоторых вариантов использования вытекают другие варианты, например из варианта просмотра докторов вытекает вариант оформления посещения для пациента.

В приложении реализована навигационная панель, таким образом пользователи с любой страницы всегда могут перейти на любую другую.

Изм.

Лист

№ докум.

Подпись

Дата

Лист

УО «ВГТУ» ДП.006 1-40 05 01-01 ПЗ

Изм.

Лист

№ докум.

Подпись

Дата

Лист

УО «ВГТУ» ДП.006 1-40 05 01-01 ПЗ