**Слайд 1.** Введение.

Здравствуйте, уважаемый председатель и члены Государственной аттестационной комиссии. Представляю вашему вниманию к защите выпускную квалификационную работу на тему: «Разработка серверной части автоматизированного рабочего места врача-терапевта».

**Слайд 2.** Актуальность работы.

На данный момент в нашей стране информатизация медицины является одним из наиболее активных направлений. Можно встретить большое количество решений, которые стремятся облегчить работу врачей. Однако на практике это не совсем так.

Крупные системы, которые используются в большинстве регионов, действительно пытаются автоматизировать некоторую работу врачей, но в то же время в таких интерфейсах достаточно тяжело работать.

Интеграция с такими системами также вызывает затруднения, поскольку стандарт обмена данными является уникальным для каждой системы.

Также недостаточно проработаны пользовательские сценарии для отдельных медицинских специальностей.

**Слайд 3.** Актуальность работы 2.

Разрабатываемая система позволит модернизировать работу врача-терапевта путем реализации удобного и понятного интерфейса, проработке пользовательских сценариев, а также внедрение систем помощи принятия решений.

Также разрабатываемая система позволит стандартизировано хранить и передавать данные. Что в свою очередь положительно повлияет на качество интеграции с другими медицинскими системами.

**Слайд 3.** Цель и задачи работы.

Целью дипломной работы является разработка медицинской информационной системы, которая позволит ускорить и упростить работу врача-терапевта.

В задачи входят:

1. Анализ и реализация взаимодействия врача терапевта с системой. В данном пункте необходимо учесть все тонкости в работе врача и реализовать их в системе.
2. Реализовать хранение и передачу данных в соответствии с международным стандартом FHIR.
3. Разработать архитектуру, которая позволит производить гибкую интеграции с несколькими медицинскими системами.
4. Учёт загруженности системы. Так как система будет иметь большую пользовательскую базу, необходимо обеспечить отказоустойчивость системы.

Объектом исследования выступала организация «ЗАО Калуга Астрал».

**Слайд 4.** Сравнительный анализ аналогов.

В научно-исследовательской части были рассмотрены существующие аналоги медицинских систем и произведен сравнительный анализ по следующим пунктам, представленным в таблице. Так как данные системы работают со всеми контекстами больницы, отдельные области, как работа врача-терапевта имеет большие недостатки в реализации интерфейса, поэтому практически все системы получили отрицательные оценки при анализе.

**Слайд 5.** Взаимодействие врача с системой.

Также в исследовательской части был проведен анализ взаимодействия врача с системой. Основные взаимодействия включают взаимодействие с электронной медицинской картой пациента и с медицинской историей. ЭМК включает в себя работу с основной, административной и медицинской информацией по пациенту, а также взаимодействие с реестром пациентов.

Медицинская история включает в себя все данные, которые относятся непосредственно для работы врача. Это взаимодействие с историей заболеваний пациента, по которой можно получить информацию по каждой болезни, по посещениям в рамках этой болезни, а также все необходимые результаты и действия, сделанные в рамках какого-либо заболевания (направления, анализы).

Также здесь находится взаимодействия с расписанием врача (по графику, без очереди, вызовы на дом), процесс приёма пациента и работу с диспансерным учётом.

Разрабатываемая система должна реализовывать все эти процессы.

**Слайд 6.** Выбранные средства разработки.

Для реализации цели курсовой работы было выбрано множество инструментов.

Так в свою очередь, для удобного масштабирования выступает Kubernetes. Это программное решение позволяет программным образом соединить выделенные физические сервера в одну инфраструктуру, а также при загруженности системы масштабировать нагруженные сервисы и распределять трафик. Docker – используется для упаковки каждого программного модуля системы в отдельные контейнеры, которые и будут горизонтально масштабироваться при нагрузках.

Данные в системе хранятся согласно международному стандарту FHIR, который описывает модели медицинских данных и их взаимодействие между собой. Aidbox – это программное решение, которое содержит базу данных и сервер для взаимодействия с данными по FHIR. Simplifier – это библиотека на C#, которая реализует интерфейс взаимодействия с сервером FHIR.

Для интерфейса взаимодействия с системой используется ASP NET Core. Это обычный сервер, который работает по HTTP протоколу. И Swagger – программный продукт, который позволяет выводить документация по API ASP NET Core приложения.

Для внутреннего взаимодействия используется брокер сообщений RabbitMQ, через него происходит общение между микросервисами. MassTransit – это реализация интерфейса взаимодействия с RabbitMQ на C#.

**Слайд 8.** Архитектура приложения.

Архитектура приложения состоит из микросервисов, каждый из который выполняет свои действия. Основные микросервисы работают с базой данных и с брокером сообщений, который позволяет общаться микросервисам друг с другом. Также в архитектуре используется микросервис сторонней системы, для синхронизации данных.

**Слайд 9.** Реализация микросервиса.

Для реализации микросервиса необходимо реализовать: внешний и внутренний интерфейсы взаимодействия, указать на каком этапе и в какие модели конвертируются данные, реализовать требуемую логику, настроить приложение, включая как микросервис будет развёртываться в системе и масштабироваться.

**Слайд 10.** Интеграция со сторонней системой.

Также в проектно-конструкторской части был приведен пример интеграции с системой РМИС. Описан протокол взаимодействия SOAP. На примере создания посещения, показана миграция данных. И как взаимодействует сторонняя система с внутренней логикой проекта.

**Слайд 11.** Проектно-технологическая часть.

В проектно-технологической части представлены минимальные и рекомендуемые системные требования, также представлено руководства по развёртыванию и использованию ПО, приведен расчёт стоимости разработки и окупаемости ПО. Также приведен анализ рабочего места на предмет соответствия нормам СанПин.

**Слайд 12.** Пути развития проекта.

В качестве путей развития проекта можно выделить улучшение проработки текущих процессов, внедрение смежных медицинских контекстов (регистратура больницы, работа скорой помощи). Внедрение искусственного интеллекта. И реализация системы от лица пациента.

**Слайд 13.** Недостатки разработки.

К недостаткам разработки можно отнести зависимость системы от интеграций. Так как без данных система не имеет значения. Также сложность в реализации и поддержки стандарта FHIR.

**Слайд 14.** Заключение.

В результате выполнения дипломной работы была выполнена поставленная цель, разработана медицинская информационная система, а также выполнены все поставленные задачи.

**Слайд 15. Спасибо за внимание!**.