***УТВЕРЖДАЮ***

\_\_Мазнина Ю.А\_\_\_ФИО

«\_\_\_»\_\_\_\_\_\_\_\_\_\_\_2020г.

**Наименование вида ИС**

**\_\_\_\_\_\_\_\_\_** «Хирургическое отделение» платной клиники\_\_\_\_\_\_\_

наименование вида ИС

**Сокращенное наименование**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**Хирургия**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

сокращенное наименование ИС

**Техническое задание**

Действует с\_18.11.2021\_

***СОГЛАСОВАНО РАЗРАБОТЧИК***

Мазнина Ю.А. Сергеева А.С.

Махиянова Э.В.

Магнитогорск 2020

**Оглавление**

[Выбор архитектуры 3](#_Toc90927114)

[Пользовательский интерфейс 4](#_Toc90927115)

[Технология работы с документами 5](#_Toc90927116)

[Выбор парадигмы разработки программного обеспечения 6](#_Toc90927117)

[Выбор языка программирования и разработки 7](#_Toc90927118)

[Декомпозиция разрабатываемого программного обеспечения 8](#_Toc90927119)

# Выбор архитектуры

Для данного приложения наиболее подходящей архитектурой будет многоуровневая архитектура.

Многоуровневая архитектура разделяет комплекс ПО на уровни по принципу взаимодействия “клиент-сервер”. Архитектура может иметь один, два и больше уровней, разделяющих ответственности между поставщиком данных и потребителем. Этот подход предлагает масштабируемость, которая может быть как горизонтальной (масштабирование сети с помощью высокопроизводительных узлов), так и вертикальной (масштабирование каждого узла путем повышения его производительности).

Преимущества:

1. Отсутствие дублирования кода программы-сервера программами-клиентами.
2. Так как все вычисления выполняются на сервере, то требования к компьютерам на которых установлен клиент снижаются.
3. Все данные хранятся на сервере, который, как правило, защищён гораздо лучше большинства клиентов. На сервере проще обеспечить контроль полномочий, чтобы разрешать доступ к данным только клиентам с соответствующими правами доступа.
4. Позволяет объединить различные клиенты. Использовать ресурсы одного сервера часто могут клиенты с разными аппаратными платформами, операционными системами и т. п.
5. Позволяет разгрузить сети за счёт того, что между сервером и клиентом передаются небольшие порции данных.

Недостатки:

1. Неработоспособность сервера может сделать неработоспособной всю вычислительную сеть. Неработоспособным сервером следует считать сервер, производительности которого не хватает на обслуживание всех клиентов, а также сервер, находящийся на ремонте, профилактике и т. п.
2. Поддержка работы данной системы требует отдельного специалиста — системного администратора.
3. Высокая стоимость оборудования.

# Пользовательский интерфейс

Интерфейс помогает двум объектам понимать друг друга и обмениваться информацией. Если каждое приложение или программа, установленная на компьютере, планшете или смартфоне, — это помощник, то интерфейс — это способ общаться (взаимодействовать) с ней, чтобы она помогала в вашем деле на работе и в жизни.

Всего различают два вида интерфейса: графический пользовательский и текстовые или интерфейсы командной строки

Для нашего приложения подходит графический пользовательский интерфейс

Преимущества:

1. Главным преимуществом GUI является то, что эти системы, доступны для людей всех уровней знаний, от абсолютного новичка до продвинутого разработчика или других технологов. Они позволяют простым пользователям открывать меню, перемещать файлы, запускать программы или искать в Интернете, не указывая компьютеру функции для выполнения через командную строку.
2. GUI также обеспечивают мгновенную обратную связь.

Недостатки:

1. Поскольку элементы представляют собой графику, а не текст, графические интерфейсы могут использовать гораздо большую вычислительную мощность по сравнению со стандартным текстовым интерфейсом.

В частности графический интерфейс состоит из нескольких подпунктов:

* Оконный интерфейс — способ организации полноэкранного интерфейса программы (разновидность графического интерфейса), в котором каждая интегральная часть располагается в графическом окне — собственном пространстве, находящемся в произвольном месте «над» основным экраном. Несколько окон, одновременно располагающихся на экране, могут перекрываться, виртуально находясь «выше» или «ниже» друг относительно друга.
* WIMP («windows, icons, menus, pointers» — окна, значки, меню, указатели) — в человеко-компьютерном взаимодействии означает взаимодействие с компьютером на базе этих элементов. Хотя его популярность постепенно падает, это слово часто используется в качестве приближённого синонима «графического интерфейса пользователя».
* Веб-интерфейс — веб-страница или совокупность веб-страниц, предоставляющая пользовательский интерфейс для взаимодействия с сервисом или устройством посредством протокола HTTP и веб-браузера. Веб-интерфейсы получили широкое распространение в связи с ростом популярности всемирной паутины и соответственно — повсеместного распространения веб-браузеров.
* Масштабируемый интерфейс пользователя (англ. zooming user interface или zoomable user interface, сокр. ZUI)) — графический интерфейс пользователя, где рабочее пространство представляет собой большую или неограниченную плоскость, на которой расположены основные элементы, свойства и содержимое которых становятся доступны по мере их «приближения» путём увеличения. Дальнейшее приближение содержимого делает доступным более глубокие уровни.
* Десктопное приложение – клиентское программное обеспечение, реализующее Windows Forms интерфейс. Приложение инсталлируется на рабочую станцию пользователя и запускается локально. Или запускается удаленно. ... Приложение не требует инсталляцию или загрузку программных модулей на рабочую станцию пользователя.

# Технология работы с документами

Приложения с графическим интерфейсом делятся на два типа однодокументный или многодокументный.

**Однодокументный интерфейс** (англ. Single document interface, SDI) — способ организации графического интерфейса приложений в отдельных окнах. Не существует «фонового» или «родительского» окна, содержащего меню или панели инструментов, по отношению к активному — каждое окно несёт в себе эти элементы. Такие приложения, позволяющие редактировать более одного документа одновременно, например, текстовые процессоры, могут создавать у пользователя впечатление, что запущена не одна копия программы, а несколько.

**Многодокументный интерфейс** (англ. multiple document interface, MDI) — способ организации графического интерфейса пользователя, предполагающий использование оконного интерфейса, в котором большинство окон (исключая, как правило, только модальные окна) расположены внутри одного общего окна. Этим он и отличается от SDI, в котором окна располагаются независимо друг от друга. Разработчики широко используют оба типа интерфейса, а зачастую и интерфейс смешанного типа. Например, Microsoft меняла интерфейс Microsoft Office от SDI к MDI, а потом вернулась обратно к SDI, хотя степень реализации включает и первое, и второе.

Данное приложение содержит в себе одну базу данных, которая нужна для работы с пациентами и сотрудниками хирургического отделения.

Для полноценной работы приложения необходимо иметь доступ к объекту хранения данных. Основными условиями должна быть поддержка асинхронность работы и высокая отказоустойчивость.

СУБД наиболее подходящая для наших требований являет MS SQL server. Основной язык манипулирования SQL и достаточно простая работа в данными.

# Выбор парадигмы разработки программного обеспечения

Объектно-ориентирование программирование (ООП):

Наиболее распространённая на данный момент парадигма. Это подвид императивного программирования — оно основано на последовательных вызовах команд, изменяющих данные, с которыми работает программа. Таким образом она оперирует объектами, и это удобно для многих приложений.

Новичков зачастую пугает аббревиатура ООП, но освоить парадигму объектно-ориентированного программирования не так сложно, как кажется. В своё время эта идея оказалась вирусной: создавать объекты, принадлежащие классам, а также использовать методы в качестве действий, которые может выполнить объект или которые можно произвести над ним. Много специалистов по Computer Science придерживаются такого подхода. Большое преимущество здесь в том, что программисту, использующему ООП, легко разобраться, что происходит в программе. Достаточно посмотреть, какие действия производит каждый из объектов.

Легче всего использовать ООП в Python, посложнее — в C++. Но если в этих языках у программиста ещё есть возможность увильнуть от ООП (например, для Python вполне подходит функциональное программирование), то в Java и C# всегда необходимо создавать классы, одних функций недостаточно.

Объектно-ориентированная парадигма имеет несколько принципов:

* Данные структурируются в виде объектов, каждый из которых имеет определенный тип, то есть принадлежит к какому-либо классу.
* Классы – результат формализации решаемой задачи, выделения главных ее аспектов.
* Внутри объекта инкапсулируется логика работы с относящейся к нему информацией.
* Объекты в программе взаимодействуют друг с другом, обмениваются запросами и ответами.
* При этом объекты одного типа сходным образом отвечают на одни и те же запросы.
* Объекты могут организовываться в более сложные структуры, например, включать другие объекты или наследовать от одного или нескольких объектов.

# Выбор языка программирования и разработки

Для разработки системы будет использоваться язык программирования Python и среда разработки Visual Studio Code.

**Python** — высокоуровневый язык программирования общего назначения с динамической строгой типизацией и автоматическим управлением памятью, ориентированный на повышение производительности разработчика, читаемости кода и его качества, а также на обеспечение переносимости написанных на нём программ.

Ориентированность на повышение производительность работы была направлена на удаление из языка так называемого «синтаксичего сахара», например, точек с запятыми и фигурных скобок. Обширная база библиотек позволяет писать разнообразные приложения и включать различные модули, написанные иными разработчиками.

**Visual Studio Code** (VS Code) — редактор исходного кода, разработанный Microsoft для Windows, Linux и macOS. Позиционируется как «лёгкий» редактор кода для кроссплатформенной разработки веб- и облачных приложений. Редактор иметь гибкую настройку: от назначения клавиш до полного набора пакетов для разработки под определенную платформу.

# Декомпозиция разрабатываемого программного обеспечения

