

Министерства науки и высшего образования Российской Федерации
федеральное государственное автономное образовательное учреждение
высшего образования «Национальный исследовательский университет
ИТМО»

Факультет программной инженерии и компьютерной техники

Информационные системы
Курсовая работа

Этап №1

Выполнili:

Вильданов Ильнур Наилевич
Бондарев Алексей Михайлович

Группа:
Р3312

Преподаватель:
Райла Мартин

Санкт-Петербург
2025 г.

Software Requirements Specification

(Спецификация требований к продукту)

1. Introduction (Введение)

Purpose (Назначение документа)

Данный документ описывает функциональные и нефункциональные требования к информационной системе «Система записи к врачу», разрабатываемой в рамках курсовой работы по дисциплине «Информационные системы».

Документ предназначен для:

- разработчиков, реализующих систему;
- тестировщиков, проверяющих соответствие реализованной системы требованиям.

Документ служит основой для всех последующих этапов разработки: проектирования базы данных, архитектуры, реализации и тестирования.

1.2 Scope (Область применения)

«Система записи к врачу» — это веб-приложение для одной медицинской организации (клиники/поликлиники), позволяющее:

- пациентам просматривать доступных врачей и свободные временные слоты и записываться на приём;
- врачам управлять своим расписанием и просматривать список записанных пациентов;
- администраторам управлять пользовательскими учетными записями, справочниками (врачи, специальности, кабинеты, услуги) и отслеживать общую загруженность.

Система предназначена для использования сотрудниками и пациентами одной клиники через веб-браузер.

Входит в функционал системы:

- регистрация и аутентификация пользователей (пациентов, врачей, администраторов);
- управление ролями и правами доступа;
- ведение справочников врачей, специальностей, кабинетов и услуг;
- управление расписанием врачей (создание и изменение временных слотов);
- запись пациента на приём, просмотр, отмена и, при необходимости, перенос записи;
- интеграция с внешним аптечным ресурсом для отображения пациенту перечня безрецептурных товаров/сервисов по выбранной симптоматике;
- уведомления в пользовательском интерфейсе;
- мобильный доступ в виде адаптивного веб-интерфейса с базовыми сценариями для пациента

- отображение простой статистики (например, количество записей по врачу/дню).

Не входит в функционал системы:

- ведение медицинских карт и истории заболеваний;
- хранение и обработка результатов анализов и назначений;
- финансовый учёт, выставление счетов, интеграция с платёжными системами;

1.3 Definitions, Acronyms and Abbreviations (Определения и аббревиатуры)

В рамках данного документа используются следующие термины и сокращения:

- **Система** — разрабатываемое веб-приложение «Система записи к врачу».
- **Пациент** — пользователь, который записывается на приём к врачу и управляет своими записями.
- **Врач** — пользователь, оказывающий медицинские услуги и использующий систему для работы с расписанием и списком записей.
- **Администратор** — пользователь с расширенными правами, управляющий пользователями, справочниками и имеющий доступ к общей информации о записях.
- **Запись (Appointment)** — сущность, описывающая назначенный приём: пациент, врач, дата, время, кабинет, статус.
- **Временной слот (Time slot)** — интервал времени в расписании врача, доступный или занятый для записи.
- **Расписание врача** — набор временных слотов врача в определённом периоде (день, неделя).
- **ИС (Информационная система)** — программный комплекс для автоматизации процессов записи к врачу.
- **DB (Database)** — база данных, в которой хранятся данные системы (PostgreSQL).
- **UI (User Interface)** — пользовательский интерфейс веб-приложения.
- **SRS (Software Requirements Specification)** — данный документ, спецификация требований к программному обеспечению.

1.4 References (Ссылки)

- ГОСТ 34.602-2018 "Информационные технологии. Комплекс стандартов на автоматизированные системы. Техническое задание на создание автоматизированной системы"
- IEEE 830-1998 "Recommended Practice for Software Requirements Specifications"
- Требования к курсовой работе по дисциплине "Информационные системы":
<https://se.ifmo.ru/courses/is/course-project>

1.5 Overview (Обзор документа)

Настоящий документ структурирован следующим образом:

- **Раздел 1 «Introduction (Введение)»** — описывает назначение документа, область применения разрабатываемой системы, используемые термины, внешние ссылки и структуру SRS.

- **Раздел 2 «Overall Description (Общее описание)»** — содержит общее описание системы: характеристики пользователей, ключевые бизнес-процессы, предполагаемую среду эксплуатации, предположения и ограничения.
- **Раздел 3 «Specific Requirements (Спецификация требований)»** — приводит детализированные функциональные требования (по ролям и прецедентам использования), а также нефункциональные требования: к удобству использования, надёжности, производительности, безопасности, интерфейсам и лицензированию.

2. Overall Description (Общее описание)

2.1 Product functions (Функционал продукта)

Система реализует базовые сценарии записи на приём для одной клиники.

Общие функции:

- регистрация нового пациента;
- вход и выход из системы (пациент, врач, администратор);
- проверка прав доступа в соответствии с ролью пользователя;
- мобильный доступ для базовых сценариев пациента;
- аптечные рекомендации — получение и отображение перечня безрецептурных товаров/сервисов через внешний аптечный сервис.

Функционал для пациента:

- просмотр списка врачей с простой фильтрацией по специальности;
- просмотр базовой информации о враче (ФИО, специальность, краткое описание);
- просмотр доступных временных слотов выбранного врача;
- создание записи на приём в свободный временной слот;
- просмотр списка своих записей (предстоящих и прошедших);
- отмена своей записи, если время приёма ещё не наступило (без переноса и листа ожидания).

Функционал для врача:

- вход в систему под учётной записью врача;
- просмотр своего расписания приёмов в виде списка записей по выбранной дате/диапазону дат;
- просмотр информации по отдельной записи (пациент, дата, время).

Функционал для администратора:

- вход в систему под учётной записью администратора;
- управление справочником врачей:
 - добавление нового врача (ФИО, специальность, базовые параметры);
 - изменение данных врача;
 - деактивация врача (чтобы его нельзя было выбирать при записи);
- управление расписанием врачей:
 - создание и удаление типовых временных слотов приёма;
- просмотр списка всех записей с возможностью фильтрации по врачу и дате.

Системные функции:

- проверка при создании записи, что:
 - выбранный временной слот существует и свободен;
 - у врача не возникает двух записей в один и тот же временной слот;
- базовый контроль целостности данных на уровне базы данных (ограничения, триггеры).

2.2 User characteristics (Описание пользователей)

В системе выделяются три группы пользователей.

Пациенты:

- назначение: самостоятельная запись на приём;
- уровень компьютерной грамотности: базовый;
- способ работы: через веб-браузер, несколько простых сценариев:
 - выбрать врача;
 - выбрать свободное время;
 - записаться, посмотреть или отменить запись.

Врачи:

- назначение: просмотр собственного расписания;
- уровень компьютерной грамотности: от базового до среднего;
- способ работы:
 - периодический вход в систему;
 - просмотр списка записей на выбранную дату;
 - просмотр краткой информации о пациенте и времени приёма.

Администраторы:

- назначение: поддержка справочников и расписаний;
- уровень компьютерной грамотности: уверенный пользователь;
- способ работы:
 - регулярное использование в рабочее время;
 - добавление/изменение врачей;
 - формирование типовых временных слотов;
 - просмотр всех записей.

2.3 Assumptions and dependencies (Влияющие факторы и зависимости)

2.3.1 Предположения

- Система используется в одной клинике, все пользователи работают в одной временной зоне.
- Доступ к системе осуществляется через веб-браузер с настольных ПК/ноутбуков и мобильных устройств.
- Система применяется для плановой записи на приём, а не для экстренной помощи.

- Расписание врачей ведётся преимущественно через систему и считается актуальным источником данных о записях.

2.3.2 Технические зависимости

- Использование СУБД PostgreSQL для хранения всех данных о пользователях, расписаниях и записях.
- Развёртывание backend-приложения и базы данных в Docker-контейнерах.
- Наличие стабильного сетевого соединения между компонентами системы (frontend/backend/БД).
- Использование внешнего аптечного REST API (наличие действующего API-ключа).

2.3.3 Влияние на требования

- Изменение организационных предположений (несколько клиник, разные часовые пояса, экстренные записи) потребует расширения функциональных требований раздела 3.1.
- Изменение СУБД или отказ от Docker повлияет на ограничения разработки (раздел 3.5) и требования к производительности и надёжности (разделы 3.4, 3.3).
- Недоступность аптечного API требует отображения локального справочника и влияет на требования к производительности и интерфейсам (разделы 3.4, 3.6).

2.4 Constraints (Ограничения)

На систему накладываются следующие ограничения:

- в качестве СУБД используется PostgreSQL;
- развёртывание системы выполняется с использованием Docker;
- backend разрабатывается с использованием технологий, разрешённых в рамках курса (Spring Boot);
- предоставляется мобильный доступ (Safari на iOS и Google Chrome на Android, реализация как адаптивный веб-интерфейс);
- внешний аптечный сервис: вызовы выполняются по HTTPS/JSON с авторизацией по API-ключу.

3. Specific Requirements (Спецификация требований)

3.1 Functionality (Функциональные требования)

3.1.1 Система должна обеспечивать регистрацию новых пациентов

- 3.1.1.1 Система должна предоставлять страницу регистрации, доступную неавторизованным пользователям.
- 3.1.1.2 Форма регистрации должна содержать поля: ФИО, адрес электронной почты (логин), пароль.
- 3.1.1.3 Система должна проверять заполненность обязательных полей, корректность формата email и минимальную длину пароля.
- 3.1.1.4 Система должна проверять уникальность email среди существующих пользователей.
- 3.1.1.5 При успешной регистрации система должна создавать учётную запись с ролью «Пациент» и уведомлять пользователя об успешной операции.

3.1.2 Система должна обеспечивать аутентификацию и выход пользователей

- 3.1.2.1 Система должна предоставлять страницу входа с полями логина (email) и пароля.
- 3.1.2.2 Система должна проверять соответствие пары логин/пароль и активность учётной записи.
- 3.1.2.3 При успешной аутентификации система должна открывать интерфейс в соответствии с ролью пользователя.
- 3.1.2.4 При ошибке аутентификации система должна отображать сообщение об ошибке без раскрытия деталей.
- 3.1.2.5 Система должна предоставлять возможность корректного выхода пользователя (завершения сессии).

3.1.3 Система должна обеспечивать пациенту просмотр списка врачей и информации о враче

- 3.1.3.1 Система должна отображать список врачей с указанием ФИО и специальности.
- 3.1.3.2 Система должна предоставлять фильтрацию списка по специальности.
- 3.1.3.3 Система должна предоставлять поиск врача по ФИО (по подстроке).
- 3.1.3.4 Система должна позволять открыть карточку врача с подробной информацией (ФИО, специальность, краткое описание, кабинет, статус активности).

3.1.4 Система должна предоставлять пациенту просмотр доступных временных слотов выбранного врача

- 3.1.4.1 Система должна отображать временные слоты приёма выбранного врача на указанную дату или диапазон дат.
- 3.1.4.2 Система должна отображать только слоты в будущем относительно текущего времени.
- 3.1.4.3 Система должна скрывать или помечать слоты, по которым уже существуют активные записи.
- 3.1.4.4 Система должна позволять пациенту выбрать свободный слот для записи.

3.1.5 Система должна обеспечивать создание записи пациента на приём к врачу

- 3.1.5.1 Система должна позволять пациенту создать запись на приём в выбранный свободный слот.
- 3.1.5.2 При создании записи система должна проверять существование и актуальную свободность слота.
- 3.1.5.3 Система должна сохранять запись с привязкой к пациенту, врачу, слоту и начальным статусом «Запланирована».
- 3.1.5.4 В случае успешного создания записи система должна отображать подтверждение и добавлять запись в список «Мои записи».
- 3.1.5.5 В случае ошибки (слот занят, врач деактивирован и т.п.) система должна отменять операцию и выводить сообщение об ошибке.

3.1.6 Система должна обеспечивать пациенту просмотр и отмену собственных записей

- 3.1.6.1 Система должна предоставлять страницу «Мои записи» со списком записей пациента.
- 3.1.6.2 Система должна отображать для каждой записи врача, дату, время и статус.
- 3.1.6.3 Система должна позволять отменять только предстоящие записи.
- 3.1.6.4 При попытке отмены система должна проверять, что время приёма ещё не наступило.
- 3.1.6.5 При успешной отмене система должна менять статус записи на «Отменена пациентом» и освобождать соответствующий слот.
- 3.1.6.6 Система должна уведомлять пациента об успешной или неуспешной отмене.

3.1.7 Система должна обеспечивать врачу просмотр собственного расписания и записей

- 3.1.7.1 Система должна предоставлять врачу страницу просмотра расписания.
- 3.1.7.2 Система должна позволять врачу выбирать дату или диапазон дат для отображения записей.
- 3.1.7.3 Система должна отображать по каждой записи пациента (ФИО), дату, время и статус.
- 3.1.7.4 Система не должна позволять врачу создавать, изменять или отменять записи через данный интерфейс (только просмотр).

3.1.8 Система должна обеспечивать администратору управление справочником врачей

- 3.1.8.1 Система должна предоставлять администратору список врачей.
- 3.1.8.2 Система должна позволять добавлять нового врача (ФИО, специальность, статус активности, при необходимости контакты).
- 3.1.8.3 Система должна позволять изменять данные существующего врача.
- 3.1.8.4 Система должна позволять деактивировать врача, чтобы его нельзя было выбирать при создании новых записей.
- 3.1.8.5 При деактивации врача система не должна удалять существующие записи, связанные с этим врачом.

3.1.9 Система должна обеспечивать администратору управление расписанием врачей

- 3.1.9.1 Система должна позволять создавать временные слоты приёма для конкретного врача (дата, время начала, время окончания).
- 3.1.9.2 Система должна проверять отсутствие пересечений по времени между слотами одного врача.
- 3.1.9.3 Система должна позволять удалять и изменять слоты, по которым ещё нет записей пациентов.
- 3.1.9.4 При наличии хотя бы одной записи по слоту система должна запрещать удаление или изменение этого слота и выводить сообщение об ошибке.
- 3.1.9.5 Система должна использовать сохранённые слоты при отображении доступного времени пациенту.

3.1.10 Система должна обеспечивать администратору просмотр всех записей пациентов

- 3.1.10.1 Система должна предоставлять администратору страницу со списком всех записей.
- 3.1.10.2 Система должна позволять фильтровать записи по врачу.

3.1.10.3 Система должна позволять фильтровать записи по дате или диапазону дат.
3.1.10.4 Система должна отображать для каждой записи пациента, врача, дату, время и статус.

3.1.11 Система должна обеспечивать контроль целостности данных расписания и записей

3.1.11.1 Система должна предотвращать создание двух разных записей в один и тот же временной слот одного и того же врача.
3.1.11.2 Система должна предотвращать создание записи для несуществующего или деактивированного врача.
3.1.11.3 Система должна предотвращать создание и изменение записей на время, которое уже прошло.
3.1.11.4 Система должна реализовывать часть проверок целостности на уровне базы данных (ограничения, триггеры), чтобы исключить некорректные операции независимо от кода приложения.
3.1.11.5 В случае нарушения целостности система должна отменять соответствующую операцию и информировать пользователя о причине отказа.

3.1.12 Система должна предоставлять пациенту аптечные рекомендации через внешний сервис

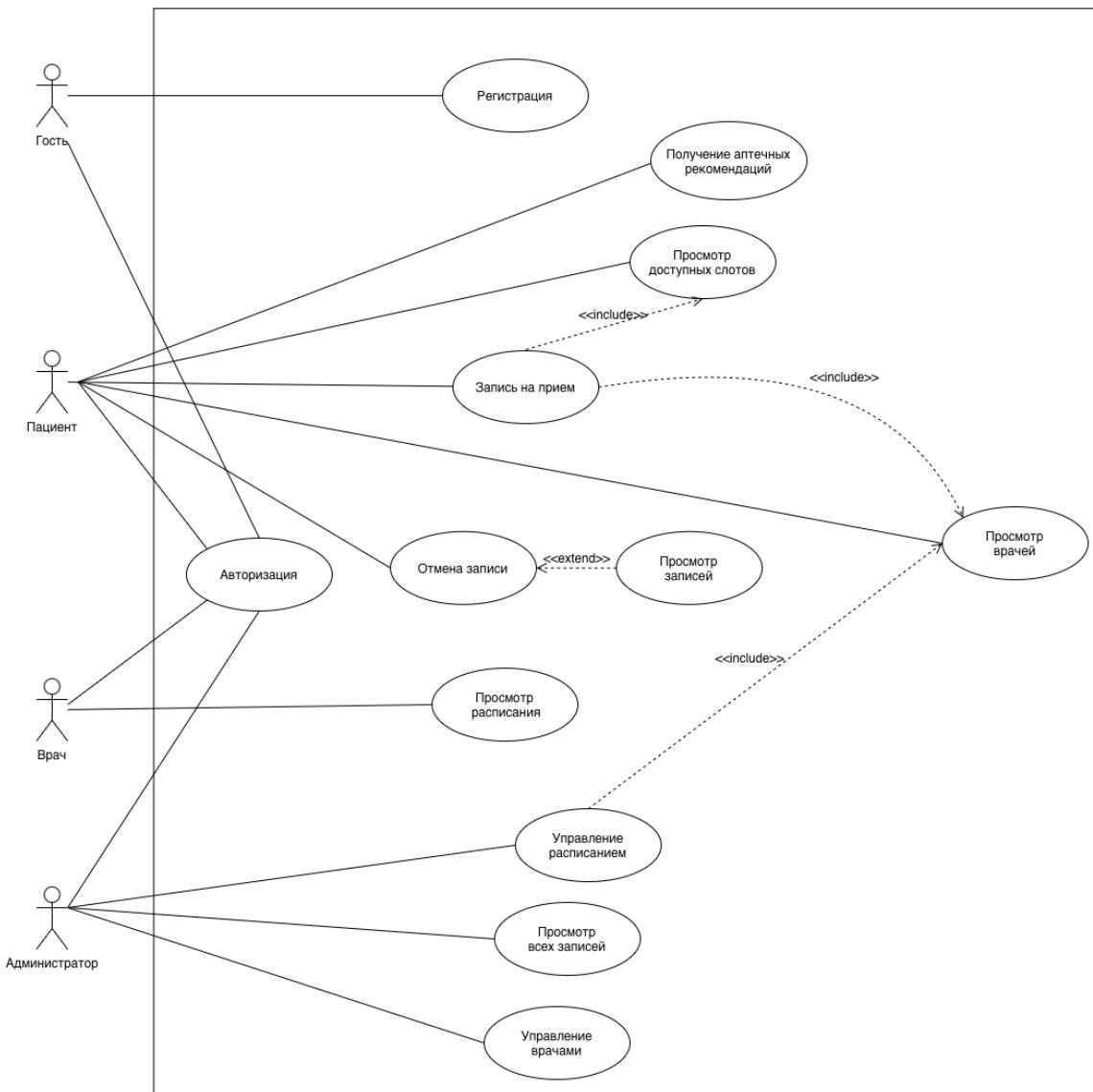
3.1.12.1 Система должна отправлять запрос к внешнему REST API с параметрами запроса (категория/симптом, возрастная группа при наличии).
3.1.12.2 Система должна принимать ответ в формате JSON и отображать пациенту список позиций
3.1.12.3 Система не должна сохранять в базе диагнозы, чувствительные медицинские данные и не должна формулировать медицинские рекомендации; отображаемая информация носит справочный характер.
3.1.12.4 При недоступности внешнего сервиса система должна отображать данные из локального справочника с пометкой «оффлайн».

3.1.13 Система должна обеспечивать мобильный доступ для iOS/Android как тонкий клиент

3.1.13.1 Мобильный клиент должен использовать те же серверные API, что и веб-интерфейс.
3.1.13.2 Должен быть обеспечен функциональный паритет базовых сценариев пациента: вход/выход, просмотр врачей и слотов, создание/отмена записи, просмотр «Моих записей».
3.1.13.3 При отсутствии сети мобильный клиент должен корректно уведомлять пользователя и не выполнять критичные операции (создание/отмена записи).

№ Требования	Название требования	Приоритетность	Трудоемкость в часах		Стабильность
			Мин	Макс	
3.1.1	Регистрация пациента	Высокая	32	64	Высокая
3.1.2	Аутентификация и выход пользователей	Высокая	32	64	Высокая

3.1.3	Просмотр списка врачей и информации о враче	Высокая	24	48	Высокая
3.1.4	Просмотр доступных временных слотов врача	Высокая	32	64	Средняя
3.1.5	Создание записи на приём	Высокая	48	96	Средняя
3.1.6	Просмотр и отмена записей пациентом	Средняя	32	64	Средняя
3.1.7	Просмотр расписания и записей врачом	Средняя	32	64	Средняя
3.1.8	Управление справочником врачей администратором	Высокая	40	80	Средняя
3.1.9	Управление расписанием врачей администратором	Высокая	48	96	Средняя
3.1.10	Просмотр всех записей администратором	Средняя	24	48	Средняя
3.1.11	Контроль целостности данных расписания и записей	Высокая	32	64	Высокая
3.1.12	Аптечные рекомендации	Средняя	56	120	Средняя
3.1.13	Мобильный доступ	Средняя	48	96	Высокая



Регистрация	
ID	UC-1
Краткое описание	Гость создаёт новую учётную запись пациента для дальнейшей работы в системе.
Главный актер	Гость
Второстепенные актеры	Система
Предусловия	1. Пользователь не авторизован в системе. 2. Для указанного email ещё не существует учётной записи.
Основной поток	1. Гость открывает страницу регистрации. 2. Система отображает форму регистрации (ФИО, email, пароль). 3. Гость заполняет форму и отправляет данные. 4. Система проверяет корректность формата данных и заполненность полей. 5. Система проверяет уникальность email.

	6. При успешной проверке система создаёт учётную запись с ролью «Пациент». 7. Система отображает сообщение об успешной регистрации и предлагает перейти к форме входа.
Постусловия	В системе создана активная учётная запись с ролью «Пациент».

Аутентификация	
ID	UC-2
Краткое описание	Пользователь входит в систему с использованием логина и пароля
Главный актер	Пациент / Врач / Администратор / Гость (при первичной авторизации)
Второстепенные актеры	Система
Предусловия	1. Для пользователя существует учётная запись с указанным email. 2. Учётная запись не заблокирована.
Основной поток	1. Пользователь открывает страницу входа. 2. Система отображает форму входа (email, пароль). 3. Пользователь вводит данные и отправляет форму. 4. Система проверяет корректность пары email/пароль и статус учётной записи. 5. При успешной проверке система создаёт сессию пользователя. 6. Система перенаправляет пользователя в стартовый раздел в соответствии с его ролью (пациент — к записям или врачам, врач — к расписанию, администратор — к панели администратора).
Постусловия	Пользователь авторизован и имеет доступ к функциям системы в соответствии со своей ролью.

Просмотр врачей	
ID	UC-3
Краткое описание	Пользователь просматривает список врачей и информацию о каждом враче.
Главный актер	Пациент
Второстепенные актеры	
Предусловия	Актёр авторизован в системе (для пациента) или имеет права администратора.
Основной поток	1. Актёр открывает раздел «Врачи». 2. Система отображает список врачей (ФИО, специальность, статус активности).

	<p>3. Актёр при необходимости задаёт фильтр по специальности и/или строку поиска по ФИО.</p> <p>4. Система обновляет список в соответствии с выбранными критериями.</p> <p>5. Актёр выбирает конкретного врача.</p> <p>6. Система отображает карточку врача с подробной информацией (ФИО, специальность, кабинет, краткое описание, статус активности).</p>
Постусловия	Актёр получил необходимую информацию о врачах и, при необходимости, выбрал врача для дальнейших действий (записи или настройки расписания).

Просмотр доступных слотов врача	
ID	UC-4
Краткое описание	Пациент просматривает свободные временные слоты выбранного врача для записи на приём.
Главный актер	Пациент
Второстепенные актеры	
Предусловия	<p>1. Пациент авторизован.</p> <p>2. Выбран конкретный врач (после UC-3).</p> <p>3. Для врача в системе заведены временные слоты.</p>
Основной поток	<p>1. Пациент на странице врача выбирает дату или диапазон дат.</p> <p>2. Система запрашивает и отображает список временных слотов врача на выбранный период.</p> <p>3. Система помечает слоты, находящиеся в прошлом, как недоступные для записи.</p> <p>4. Система скрывает или отмечает занятые слоты, по которым уже существуют активные записи.</p> <p>5. Пациент просматривает список и выбирает один свободный временной слот для записи.</p>
Постусловия	Выбран конкретный свободный временной слот, который может быть использован в precedente «Запись пациента на приём».

Запись пациента на приём	
ID	UC-5
Краткое описание	Пациент создаёт запись на приём к врачу в выбранный свободный временной слот.
Главный актер	Пациент

Второстепенные актеры		
Предусловия		1. Пациент авторизован. 2. Пациент выбрал врача (UC-3). 3. Пациент выбрал свободный временной слот (UC-4).
Основной поток		1. Пациент нажимает кнопку «Записаться» для выбранного временного слота. 2. Система повторно проверяет, что слот существует, относится к активному врачу и находится в будущем. 3. Система проверяет, что по этому слоту ещё не создана активная запись другого пациента. 4. При успешных проверках система создаёт запись на приём, связывая её с пациентом, врачом и временным слотом, и устанавливает статус «Запланирована». 5. Система отображает сообщение об успешной записи и добавляет её в список «Мои записи».
Постусловия		В системе существует запись на приём с корректными связями (пациент, врач, слот) и статусом «Запланирована».

Просмотр своих записей (пациент)	
ID	UC-6
Краткое описание	Пациент просматривает список всех своих записей на приём.
Главный актер	Пациент
Второстепенные актеры	
Предусловия	1. Пациент авторизован. 2. В системе могут существовать записи, связанные с этим пациентом.
Основной поток	1. Пациент открывает раздел «Мои записи». 2. Система запрашивает все записи пациента. 3. Система отображает список записей с указанием врача, даты, времени и статуса. 4. Система визуально различает предстоящие и прошедшие записи (например, группировкой или пометками). 5. Пациент просматривает список записей и при необходимости выбирает запись для дальнейших действий (например, отмены).
Постусловия	Пациент получил актуальную информацию о своих записях.

Отмена своей записи (пациент)	
ID	UC-7
Краткое описание	Пациент отменяет одну из своих предстоящих записей на приём.
Главный актер	Пациент
Второстепенные актеры	
Предусловия	<ol style="list-style-type: none"> 1. Пациент авторизован. 2. Выполнен прецедент UC-6 «Просмотр своих записей» (пациент видит список записей). 3. В списке есть хотя бы одна будущая запись.
Основной поток	<ol style="list-style-type: none"> 1. Пациент в списке «Мои записи» выбирает предстоящую запись для отмены. 2. Пациент инициирует отмену (нажимает кнопку «Отменить»). 3. Система проверяет, что время приёма ещё не наступило. 4. Система меняет статус записи на «Отменена пациентом». 5. Система освобождает соответствующий временной слот для возможности повторной записи другим пациентом. 6. Система отображает сообщение об успешной отмене записи.
Постусловия	Статус выбранной записи изменён, временной слот при необходимости становится доступным для других пациентов.

Просмотр расписания врачом	
ID	UC-8
Краткое описание	Врач просматривает список своих записей на выбранный день или период.
Главный актер	Врач
Второстепенные актеры	
Предусловия	<ol style="list-style-type: none"> 1. Врач авторизован. 2. В системе могут существовать записи, связанные с данным врачом.
Основной поток	<ol style="list-style-type: none"> 1. Врач открывает раздел «Моё расписание». 2. Система отображает записи врача на текущий день. 3. Врач при необходимости выбирает другую дату или диапазон дат. 4. Система обновляет список записей в соответствии с выбранным периодом. 5. Врач просматривает список записей (пациент, дата, время, статус).

Постусловия	Врач получил актуальную информацию о своих записях на указанный период.
-------------	---

Управление врачами (администратор)	
ID	UC-9
Краткое описание	Администратор управляет справочником врачей: добавляет, редактирует и деактивирует записи о врачах.
Главный актер	Администратор
Второстепенные актеры	
Предусловия	1. Администратор авторизован. 2. У администратора есть права на управление справочниками.
Основной поток	1. Администратор открывает раздел «Врачи (администрирование)». 2. Система отображает список врачей. 3. Администратор выбирает действие «Добавить врача». 4. Система отображает форму создания врача (ФИО, специальность, статус активности и др.). 5. Администратор заполняет форму и сохраняет данные. 6. Система создаёт запись врача и отображает её в списке.
Постусловия	Справочник врачей актуализирован: новые врачи добавлены, данные существующих изменены, неактуальные деактивированы.

Управление расписанием врача (администратор)	
ID	UC-10
Краткое описание	Администратор создаёт, изменяет и удаляет временные слоты приёма для выбранного врача.
Главный актер	Администратор
Второстепенные актеры	
Предусловия	1. Администратор авторизован. 2. В системе есть хотя бы один активный врач. 3. Администратор имеет доступ к прецеденту «Просмотр врачей» (UC-3) для выбора врача.
Основной поток	1. Администратор открывает раздел «Расписание (администрирование)». 2. Система предлагает выбрать врача из списка врачей (UC-3). 3. Администратор выбирает врача. 4. Система отображает существующие

	<p>временные слоты выбранного врача на выбранный период.</p> <p>5. Администратор выбирает действие «Добавить слот».</p> <p>6. Система отображает форму с полями: дата, время начала, время окончания.</p> <p>7. Администратор вводит данные и сохраняет форму.</p> <p>8. Система проверяет, что новый слот не пересекается по времени с существующими слотами врача.</p> <p>9. При успешной проверке система создаёт слот и отображает обновлённое расписание.</p>
Постусловия	Расписание выбранного врача обновлено; новые слоты сохранены, некорректные операции отклонены.

Просмотр всех записей (администратор)	
ID	UC-11
Краткое описание	Администратор просматривает список всех записей пациентов с возможностью фильтрации по врачу и дате.
Главный актер	Администратор
Второстепенные актеры	
Предусловия	Администратор авторизован. В системе существуют записи пациентов (необязательно в выбранный момент).
Основной поток	<p>1. Администратор открывает раздел «Записи (администрирование)».</p> <p>2. Система отображает список записей (например, за текущий день).</p> <p>3. Администратор при необходимости задаёт фильтр по врачу и/или дате/диапазону дат.</p> <p>4. Система применяет фильтр и обновляет список записей.</p> <p>5. Администратор просматривает список (пациент, врач, дата, время, статус).</p>
Постусловия	Администратор получил актуальную сводную информацию о записях пациентов.

Получение аптечных рекомендаций (пациент)	
ID	UC-12
Краткое описание	Пациент получает перечень аптечных позиций по выбранному

	симптому/категории через внешний сервис с переключением на локальный справочник при недоступности сервиса.
Главный актер	Пациент
Второстепенные актеры	
Предусловия	Пациент авторизован.
Основной поток	<ol style="list-style-type: none"> 1. Пациент открывает раздел «Аптечные рекомендации». 2. Пациент выбирает симптом/категорию (при наличии может выбрать возрастную группу). 3. Система отправляет запрос к внешнему аптечному сервису и ожидает ответ. 4. При получении ответа система отображает список позиций: наименование, краткое описание, форма выпуска, ориентировочная цена/ссылка (если предоставлена). 5. Если сервис недоступен или время ожидания превышено, система отображает данные локального справочника с пометкой «оффлайн».
Постусловия	Пациент получил справочную информацию об аптечных позициях; данные в базе системы не изменены.

3.2 Usability (Требования к удобству использования)

3.2.1 Система должна обеспечивать быстрое освоение интерфейса.

3.2.2 Система должна ограничивать количество шагов для типовых операций.

3.2.3 Система должна обеспечивать единообразие интерфейса.

3.2.4 Система должна обеспечивать понятную обратную связь.

3.3 Reliability (Требования к надежности)

3.3.1 Система должна обеспечивать доступность в учебной среде не ниже 95% времени демонстраций.

3.3.2 Система должна обеспечивать целостность данных.

3.3.3 Система должна устойчиво обрабатывать некорректные действия пользователя.

Некорректный ввод и повторная отправка форм не должны приводить к падению приложения или появлению дублирующих/частично сохранённых записей.

3.3.4 Система должна сохранять все критичные данные только в БД.

3.4 Performance (Требования к производительности)

3.4.1 Система должна обеспечивать приемлемое время ответа для типовых операций.

Для операций входа, просмотра списка врачей, просмотра слотов, создания/отмены записи и просмотра расписания время ответа сервера не должно превышать 3 секунд при нагрузке до 20 одновременных активных пользователей.

3.4.2 Система должна поддерживать работу с учебным объёмом данных.

Система должна корректно работать при объёме до 1000 пользователей и до 10 000 записей на приём без заметного ухудшения времени ответа по сравнению с требованием 3.4.1.

3.4.3 Время ответа внешнего аптечного сервиса по итогам измерений должно укладываться для девяноста пяти процентов запросов не более чем в две с половиной секунды. Если запрос выполняется дольше трёх секунд, система считает сервис недоступным и автоматически использует локальный справочник (оффлайн-режим).

3.5 Design Constraints (Ограничения разработки)

3.5.1 Система должна быть реализована с использованием согласованного стека технологий.

Backend разрабатывается на языке Java с использованием Spring Boot. В качестве СУБД используется PostgreSQL.

3.5.2 Система должна быть развёрнута в контейнерах Docker.

Приложение и база данных запускаются в отдельных Docker-контейнерах; допускается использование docker-compose для локального развёртывания.

3.5.3 Система должна иметь веб-интерфейс.

Уровень представления реализуется веб-фронтеном (React), взаимодействующим с бэкендом по HTTP API.

3.5.4 Интеграция с аптечным сервисом.

Интеграция выполняется через слой адаптера (PharmacyProviderAdapter) с возможностью подмены провайдера. Запросы выполняются по HTTPS/JSON.

3.5.5 Конфигурация.

Параметры подключения к БД и внешнему API (ключ) задаются переменными окружения/файлами конфигурации и не хранятся в репозитории.

3.6 Interfaces (Интерфейсы)

3.6.1 User Interfaces (Пользовательские интерфейсы)

- Адаптивный веб
- Основные экраны: «Врачи», «Слоты», «Мои записи», «Расписание (врач)», «Администрирование»
- Единый стиль уведомлений об успехе/ошибке

- Поддерживаемые мобильные браузеры: Safari (iOS 15+), Google Chrome (Android 12+)

3.6.2 Hardware Interfaces (Аппаратные интерфейсы)

- Обычный ПК, ноутбук или смартфон с доступом в интернет
- Специальные аппаратные интерфейсы не требуются

3.6.3 Software Interfaces (Программные интерфейсы)

- СУБД PostgreSQL — подключение из backend-приложения по стандартному протоколу PostgreSQL.
- При использовании отдельного фронтенда — взаимодействие с backend по REST/HTTP API (JSON).
- Интеграция с внешним аптечным сервисом

3.6.4 Communications Interfaces (Сетевые интерфейсы)

- Взаимодействие клиента с системой по HTTP/HTTPS.
- Взаимодействие серверного приложения с аптечным сервисом по протоколу HTTPS с обменом данными в формате JSON. Время ожидания ответа - три секунды, при временных ошибках допускается до двух повторных попыток. Если сервис недоступен, система автоматически использует локальный справочник.

3.7 Licensing Requirements (Требования к лицензированию)

Система предполагается распространяться по лицензии **MIT**.

- Разрешает свободное использование, копирование, модификацию и распространение кода
- Не требует открывать производные исходные коды
- Снимает ответственность с автора