

Министерство науки и высшего образования Российской Федерации
федеральное государственное автономное образовательное учреждение высшего образования
«НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО»

Отчет

по лабораторной работе №3 «Разработка технического задания»

по дисциплине «Проектирование информационных систем»

Автор: Ковалев Г. П.

Факультет: ПИН

Группа: К3341

Преподаватель: Духанов А.В.

ИТМО

Санкт-Петербург 2025

Я буду выполнять вариант номер 7 «HealthNavigator». Напомню, что это приложение для мониторинга здоровья, использующее носимые устройства для отслеживания жизненных показателей и предлагающее рекомендации по улучшению самочувствия.

Проектируемая информационная система по своему масштабу относится к малым (или средним) системам, так как ориентирована на ограниченный круг пользователей и решает узконаправленную задачу. Система не предполагает сложной структуры из независимых подсистем, для которых требовалась бы разработка отдельных технических заданий. В связи с этим в рамках данной работы разрабатывается частное техническое задание, охватывающее систему в целом.

1. Общие сведения о разработке

Разрабатываемая информационная система HealthNavigator предназначена для мониторинга состояния здоровья пользователей и поддержки принятия решений, направленных на улучшение самочувствия. Система собирает и обрабатывает данные о жизненных показателях, получаемые с носимых устройств, таких как умные часы и фитнес-браслеты. На основе собранных данных система выполняет анализ показателей и формирует рекомендации по образу жизни и состоянию здоровья пользователя. Взаимодействие с системой осуществляется через мобильное приложение, а также через веб-интерфейс для просмотра агрегированных данных. Обработка и хранение информации реализуются с использованием облачных технологий, что обеспечивает масштабируемость и доступность сервиса. Архитектура системы предполагает клиент-серверную модель с возможностью интеграции с внешними медицинскими информационными системами при согласии пользователя.

2. Основание для разработки

Основанием для разработки информационной системы HealthNavigator является учебное задание по дисциплине «Проектирование информационных систем», выполняемое в рамках образовательной программы. Разработка системы осуществляется в соответствии с методическими указаниями курса и направлена на закрепление навыков анализа и проектирования информационных систем.

3. Назначение разработки

Назначением разработки является создание информационной системы HealthNavigator, обеспечивающей мониторинг состояния здоровья пользователей и формирование рекомендаций по улучшению самочувствия на основе данных с носимых устройств. Система предназначена для использования частными лицами в повседневной жизни, а также может применяться врачами для ознакомления с агрегированными показателями состояния здоровья при согласии пользователя.

Ниже — аккуратно оформленный пункт 4. Требования к системе → Функциональные требования, в учебном формате и логически согласованный с ранее описанной системой HealthNavigator.

4. Требования к системе

4.1 Функциональные требования

№	Функция системы	Поддерживаемый бизнес-процесс	Краткое описание
1	Сбор данных о жизненных показателях	Мониторинг состояния здоровья пользователя	Система обеспечивает автоматический сбор данных о пульсе, уровне физической активности, качестве сна и других показателях с носимых устройств пользователя.
2	Анализ и обработка показателей здоровья	Оценка состояния здоровья	Система выполняет обработку и анализ полученных данных, выявляет отклонения от нормативных значений и формирует обобщённую оценку состояния здоровья пользователя.
3	Формирование персональных рекомендаций	Поддержка принятия решений пользователем	На основе результатов анализа система генерирует рекомендации по физической активности, режиму сна и образу жизни с целью улучшения самочувствия пользователя.
4	Интеграция с внешними системами	Обмен медицинскими данными	Система обеспечивает интеграцию с внешними сервисами и информационными системами носимых устройств, а также с медицинскими информационными системами для передачи данных при наличии согласия пользователя.

4.2 Требования к структуре

№	Компонент системы	Назначение
1	Модуль сбора данных	Обеспечивает получение данных о жизненных показателях пользователя с носимых устройств и их первичную обработку.
2	Модуль анализа данных	Выполняет анализ и обработку собранных показателей, выявляет отклонения и формирует обобщённую оценку состояния здоровья.
3	Модуль рекомендаций	Отвечает за формирование персонализированных рекомендаций на основе результатов анализа данных и пользовательских настроек.
4	Модуль интеграции и обмена данными	Обеспечивает взаимодействие с внешними сервисами носимых устройств и медицинскими информационными системами при наличии согласия пользователя.

4.3 Требования к интерфейсу

№	Интерфейсный элемент	Функциональность и поддерживаемый бизнес-процесс
1	Главная панель мониторинга	Отображает текущие и исторические показатели здоровья пользователя в виде графиков и числовых значений, поддерживает бизнес-процесс мониторинга состояния здоровья.
2	Форма детального анализа показателей	Предназначена для просмотра динамики жизненных показателей за выбранный период с использованием графиков и диаграмм, поддерживает процесс оценки состояния здоровья.

№	Интерфейсный элемент	Функциональность и поддерживаемый бизнес-процесс
3	Экран рекомендаций	Отображает персональные рекомендации по физической активности, режиму сна и образу жизни, поддерживает процесс принятия решений пользователем.
4	Экран настроек и согласий	Обеспечивает управление подключёнными устройствами и настройками обмена данными с внешними системами, поддерживает процесс интеграции и управления данными.

4.4 Технологические требования

№	Технология / метод	Обоснование применения
1	Облачные технологии хранения и обработки данных	Использование облачных технологий необходимо для централизованного хранения данных о здоровье пользователей, обеспечения масштабируемости системы и доступа к данным с различных устройств.
2	Механизмы интеграции через API	Применение API необходимо для взаимодействия системы с носимыми устройствами и внешними медицинскими информационными системами, а также для обмена данными при согласии пользователя.
3	Методы агрегации и фильтрации данных	Использование методов агрегации и фильтрации позволяет повысить качество обрабатываемых данных, исключить шум и некорректные значения, а также подготовить данные для последующего анализа и формирования рекомендаций.

4.5 Требования к качеству результатов

№	Метрика качества	Требуемое значение
1	Точность обработки данных	не менее 95 %
2	Время обновления показателей	не более 5 секунд
3	Актуальность данных для рекомендаций	не более 24 часов

4.6 Технические требования

№	Область	Требование	Пояснение
1	Клиентские платформы	iOS-приложение на Swift (iOS 17+), Android-приложение на Kotlin (Android 9+), веб-интерфейс на JavaScript/TypeScript (современные браузеры)	Обеспечивает доступ к системе с основных пользовательских устройств и покрывает основные сценарии использования
2	Серверная часть	Серверные сервисы реализуются на Go в клиент–серверной архитектуре	Go подходит для высоконагруженных сетевых сервисов и удобен для масштабирования и поддержки API.
3	Обмен данными	Взаимодействие клиент–сервер должно выполняться по HTTPS; формат данных — JSON	Единые форматы обеспечивают совместимость разных клиентов и корректную обработку временных рядов показателей.

№	Область	Требование	Пояснение
4	Хранение данных	Использование реляционной СУБД PostgreSQL для хранения транзакционных и справочных данных; использование нереляционного хранилища или БД временных рядов для телеметрии	Использование реляционной СУБД PostgreSQL для хранения транзакционных и справочных данных; использование нереляционного хранилища или БД временных рядов для телеметрии
5	Интеграция	Поддержка интеграции с внешними сервисами/устройствами через API (SDK/REST), обмен данными — только при согласии пользователя	Необходимо для получения данных сносимых устройств и (опционально) передачи агрегированных данных во внешние системы.
6	Безопасность и доступ	Аутентификация по токенам (OAuth2/OIDC, JWT), разграничение прав доступа (пользователь/врач/администратор)	Требуется для защиты персональных (в т.ч. медицинских) данных и разделения сценариев использования.
7	Совместимость и сопровождение	Версионирование API (например, v1/v2), обратная совместимость при минорных изменениях	Позволяет обновлять клиенты и сервер независимо и снижает риск «поломки» приложений при обновлениях.
8	Развертывание и эксплуатация	Возможность развёртывания в контейнерах (Docker) и централизованное логирование	Упрощает переносимость между средами (учебная/тест/прод) и помогает диагностировать ошибки.

5. Требования к программной документации

№	Вид документа	Назначение документа	Регламентирующий стандарт
1	Описание программы	Содержит сведения о назначении системы, выполняемых функциях, логической структуре и основных алгоритмах работы	ГОСТ 19.402–78
2	Описание применения	Определяет условия и порядок использования системы, основные сценарии работы пользователя	ГОСТ 19.502–78
3	Руководство системного программиста	Описывает порядок установки, настройки, обновления и сопровождения программного обеспечения	ГОСТ 19.503–79
4	Программа и методика испытаний	Устанавливает порядок и методы проверки соответствия системы требованиям технического задания	ГОСТ 19.301–79

6. Порядок выполнения работ

1. Анализ требований и постановка задачи

На данном этапе выполняется анализ предметной области, требований пользователей и условий эксплуатации системы. Формируется и утверждается техническое задание на разработку.

Результат этапа: утверждённое техническое задание.

2. Проектирование архитектуры и структуры системы

Разрабатывается общая архитектура системы, определяется состав модулей, способы взаимодействия компонентов и интеграции с внешними системами.

Результат этапа: архитектурное описание системы и структурная схема.

3. Разработка программных компонентов

Выполняется реализация серверной и клиентских частей системы в соответствии с утверждённой архитектурой и техническими требованиями.

Результат этапа: реализованные программные модули системы.

4. Интеграция и предварительное тестирование

Проводится интеграция всех компонентов системы, а также модульное и интеграционное тестирование для выявления и устранения ошибок.

Результат этапа: интегрированная версия системы, готовая к испытаниям.

5. Испытания системы

Испытания проводятся в соответствии с программой и методикой испытаний, включая функциональные и приёмочные испытания, направленные на проверку соответствия системы требованиям ТЗ.

Результат этапа: протоколы испытаний и заключение о готовности системы.

6. Приёмка и ввод в эксплуатацию

Осуществляется приёмка системы по результатам испытаний и передача её в эксплуатацию с сопроводительной программной документацией.

Результат этапа: акт приёмки и комплект программной документации.