**Спецификация**

1. **Введение** 
   1. **Назначение**

Программное средство фитнес-клуба включает в себя модули регистрации, онлайн-записи на тренировки, просмотра расписания, расчета калорий, персональных программ тренировок и питания, а также систему отслеживания прогресса.

Документация составлена для разработчиков, тестировщиков, дизайнеров и менеджеров проекта.

* 1. **Соглашения и стандарты**

В документе используются следующие соглашения:

* Обязательные требования выделены жирным текстом.
* Названия модулей и компонентов программного обеспечения именуются с начальной заглавной буквы, например: Модуль регистрации.
* Все функциональные требования пронумерованы по шаблону FR-XX.
  1. **Границы проекта**

Приложение предназначено улучшить взаимодействие клиентов с услугами фитнес-клуба. Оно позволяет пользователям легко записываться на тренировки, следить за своим прогрессом и пользоваться персонализированными программами.

Основная цель приложения - увеличить вовлечённость клиентов и упростить процесс бронирования и коммуникации между клиентами и тренерами.

* 1. **Ссылки**

[**https://diskstation.belstu.by:5001**](https://diskstation.belstu.by:5001)

[**https://metanit.com**](https://metanit.com)

**https://krazytech.com/projects**

1. **Общее описание**

**2.1 Общий взгляд на продукт**

Приложение представляет собой современное программное решение, созданное с нуля для улучшения цифрового взаимодействия клиентов с сетью фитнес-клубов. Это полностью новый продукт, не основанный на предыдущих версиях или обновлениях других систем.

Программное средство предоставляет пользователю возможность:

* записываться на тренировки,
* отслеживать прогресс,
* получать индивидуальные планы тренировок и питания от тренеров,
* рассчитывать калории.

**2.2 Классы и характеристики пользователей**

Программа ориентирована на несколько типов пользователей:

* Новички — клиенты, недавно записавшиеся в клуб. Им необходим интуитивно понятный интерфейс, помощь в регистрации и рекомендации по тренировкам.
* Постоянные клиенты — активно занимающиеся клиенты, использующие расширенные функции, такие как трекеры прогресса, расписание, анализ достижений.
* Тренеры — сотрудники, управляющие расписанием, просматривающие информацию по клиентам и составляющие индивидуальные программы.
* Администраторы клуба — сотрудники, которые управляют учетными записями, контролируют бронирования, статистику и загруженность залов.

**2.3 Операционная среда**

Приложение предназначено для работы на персональных компьютерах и ноутбуках под управлением операционных систем семейства Windows (версии не ниже Windows 10). Приложение должно корректно функционировать как на 64-битных, так и на 32-битных системах, однако рекомендуется использование 64-битной архитектуры для наилучшей производительности.

Минимальные системные требования:

* Оперативная память: не менее 4 ГБ;
* Свободное место на диске: минимум 500 МБ для установки;

Подключение к сети Интернет для доступа к облачным функциям (авторизация, обновление, синхронизация данных и т. п.).

Приложение должно обеспечивать стабильную работу в различных географических локациях, но основными регионами являются Беларусь, Россия и страны СНГ.

**2.4 Ограничения дизайна и реализации**

* Технические требования: Использование языка программирования С# и фреймворка WPF.
* Интерфейс разрабатывается только под Windows.
* Все тексты должны быть доступны на русском и английском языках.
* Программа должна быть понятна для аудитории, что подразумевает разработку ненагруженного и простого интерфейса.

**2.5 Предположения и зависимости**

* Приложение зависит от функционирования сервера базы данных, где хранятся расписания, данные тренировок и пользовательские профили.
* Ожидается наличие у пользователя аккаунта в системе, зарегистрированного через приложение.
* Пользователи будут работать в среде, соответствующей техническим требованиям, указанным в разделе 2.3.
* Все изменения в функциональности и интерфейсе должны соответствовать бренду фитнес-клуба и быть одобрены его руководством.

1. **Функции системы**

**3.1 Потск расписания тренировок**

**3.1.1 Описание**

Приоритет: высокий

Функция позволяет пользователю фильтровать доступные тренировки по типу, дню недели, времени, локации и тренеру для быстрого поиска подходящего варианта.

**3.1.2 Функциональные требования**

* FR-3.1.2.1: Пользователь может выбрать тип тренировки из выпадающего списка.
* FR-3.1.2.2: Пользователь может отфильтровать тренировки по дню недели и времени.
* FR-3.1.2.3: Пользователь может отфильтровать тренировки по тренеру.
* FR-3.1.2.4: Приложение должно отображать только те тренировки, которые соответствуют установленным фильтрам.
* FR-3.1.2.5: При отсутствии результатов по выбранным фильтрам отображается соответствующее сообщение ("Нет доступных тренировок").
* FR-3.1.2.6: Пользователь может сбросить фильтры одной кнопкой.
* FR-3.1.2.7: В случае ввода некорректных данных или сбоя фильтрации должно отображаться сообщение об ошибке.

**3.2 Добавление приема пищи в калькулятор калорий**

**3.2.1 Описание**

Функция позволяет пользователю вносить информацию о съеденных продуктах в калькулятор калорий для расчета дневной нормы и отслеживания баланса.

Приоритет: высокий

**3.2.2 Функциональные требования**

* FR-3.2.2.1: Пользователь может выбрать прием пищи.
* FR-3.2.2.2: Пользователь может добавить один или несколько продуктов из встроенного справочника.
* FR-3.2.2.3: Пользователь может указать массу каждого продукта вручную (в граммах).
* FR-3.2.2.4: Система автоматически рассчитывает калорийность и пищевую ценность по введённым данным.
* FR-3.2.2.5: Пользователь может редактировать или удалить добавленный прием пищи.
* FR-3.2.2.6: В случае отсутствия продукта в базе пользователь может добавить новый продукт вручную.
* FR-3.2.2.7: При ошибке (например, пустое поле или отрицательное значение массы) отображается предупреждение.

**3.3 Регистрация пользователя**

**3.3.1 Описание**

Приоритет**:** высокий.

Функция позволяет новому пользователю создать личный аккаунт в системе с последующим доступом к персонализированному контенту.

**3.3.2 Функциональные требования**

* FR-3.3.2.1: Пользователь должен ввести имя, фамилию, адрес электронной почты, пароль, подтверждение пароля.
* FR-3.3.2.2: Адрес электронной почты должен быть проверен на корректный формат.
* FR-3.3.2.3: Пароль должен содержать не менее 8 символов, включая хотя бы одну заглавную букву и одну цифру.
* FR-3.3.2.4: При несовпадении пароля и его подтверждения должно отображаться предупреждающее сообщение.
* FR-3.3.2.5: После успешной регистрации пользователь должен автоматически переходить на экран входа или в личный кабинет.
* FR-3.3.2.6: В случае повторной попытки регистрации с уже используемой почтой пользователь получает сообщение об ошибке.
* FR-3.3.2.7: Все поля формы должны быть обязательными для заполнения.
* FR-3.3.2.8: Данные пользователя должны быть сохранены в базе данных с использованием безопасного хэширования пароля.
* FR-3.3.2.9: В случае сбоев соединения или других ошибок регистрации должно отображаться сообщение с инструкцией повторить попытку позже.

И т д…..

1. **Требования к данным**

**4.1 Логическая модель данных**

Система обрабатывает несколько ключевых сущностей и их взаимосвязи:

* Пользователь. Связан с персональными данными, планами тренировок, питанием, прогрессом.
* Тренировка. Связана с пользователем, расписанием, типом активности, длительностью, калорийностью.
* Прием пищи. Включает дату, тип приема (завтрак, обед, ужин, перекус), продукты, калорийность.
* Продукты. Содержат данные о пищевой ценности, весе, составе, категории.
* Результаты. Содержат показатели физической активности, веса, объёмов тела и прочее.

**4.2 Словарь данных**

Словарь данных содержит описание ключевых структур и элементов данных, используемых в системе. Он помогает определить, какие данные обрабатываются приложением, как они структурированы, а также какие типы и ограничения применяются к ним.

Пользователь — основная сущность системы. Для каждого пользователя сохраняются следующие атрибуты:

* user\_id — уникальный идентификатор пользователя, представленный в формате UUID.
* name — имя пользователя, строка длиной до 50 символов.
* email — адрес электронной почты, соответствующий стандарту RFC 5322, обязательный для регистрации.
* password\_hash — хеш пароля, зашифрованный с использованием безопасного алгоритма (например, bcrypt или SHA256).
* role — роль пользователя в системе

**Тренировка** — объект, описывающий физическую активность пользователя. Атрибуты:

* training\_id — уникальный идентификатор тренировки.
* training\_type — тип тренировки, выбираемый из ограниченного набора значений, например: "кардио", "силовая", "йога".
* duration — продолжительность тренировки в минутах.
* calories\_burned — количество сожжённых калорий, вычисляется автоматически на основе типа тренировки и длительности.

**Прием пищи** — запись о том, что пользователь съел. Для каждого приёма пищи фиксируются:

* meal\_id — уникальный идентификатор.
* date — дата приема пищи в формате YYYY-MM-DD.
* meal\_type — тип приёма пищи: завтрак, обед, ужин или перекус.
* product\_list — список продуктов, связанных с приемом пищи.

**Продукт** — пищевая единица, содержащая:

* product\_name — наименование продукта, строка до 100 символов.
* calories — энергетическая ценность (ккал).
* proteins, fats, carbohydrates — пищевые значения, выраженные в граммах.
* portion\_size — размер порции в граммах или миллилитрах.

Все перечисленные поля имеют строго определённые типы данных и проходят валидацию при вводе. Некоторые поля, такие как e-mail и идентификаторы, должны быть уникальными, а обязательные поля не могут быть пустыми.

**4.3 Отчеты**

Приложение генерирует следующие логические отчеты:

1. Отчет о калорийности за день. Отображает суммарную калорийность всех приемов пищи за дату.

* Сортировка: по времени приема пищи
* Сводка: итог по белкам, жирам, углеводам и калориям

1. Отчет о прогрессе пользователя. Сравнение текущих параметров тела и активности за выбранный период.

* Сортировка: по дате
* Отображение: график веса, объёмов, количества тренировок

1. Отчет о выполненных тренировках. Список всех завершенных тренировок с указанием длительности и типа.

* Сортировка: по дате
* Фильтрация: по типу тренировки и дате

**4.4 Получение, целостность, хранение и утилизация данных**

* Система получает данные как вручную от пользователя, так и автоматически через БД. К пользовательскому вводу относятся регистрационные данные, информация о приёмах пищи, тренировках и параметрах тела.
* Целостность данных обеспечивается на нескольких уровнях. Все идентификаторы и адреса электронной почты должны быть уникальными. Форматы данных проверяются как на клиентской стороне, так и на сервере. Обязательные поля, такие как e-mail или пароль, не допускаются к сохранению без заполнения.
* Для хранения информации используется надёжная СУБД Microsoft SQL Server.
* Кэширование данных осуществляется в ограниченном объеме и очищается автоматически после выхода пользователя из системы.

### ****5. Требования к внешним интерфейсам****

### ****5.1 Пользовательские интерфейсы****

Приложение должно предоставлять интуитивно понятный графический интерфейс, соответствующий следующим требованиям:

* Использование единого стиля оформления экранов: единая цветовая палитра, шрифты (по умолчанию — Segoe UI), иконки, размеры элементов.
* Интерфейс должен соответствовать рекомендациям по UI-дизайну для Windows.
* Экранные формы должны поддерживать разрешение не ниже 1366×768 пикселей.
* Элементы должны быть стандартизированы и использоваться во всех модулях приложения.
* Во всех экранах должна присутствовать панель навигации с кнопками «Главная», «Профиль», «Настройки», «Справка».
* Поддержка клавиш быстрого доступа (например, Enter – ввод данных через поля, Esc — закрыть окно).
* Встроенная проверка корректности данных в формах: числа должны вводиться только в числовые поля, проверка на обязательные поля перед сохранением
* Поддержка локализации: все текстовые элементы должны быть вынесены в отдельный ресурсный файл.

### ****5.2 Интерфейсы программного обеспечения****

* Приложение должно взаимодействовать с локальной базой данных MS SQL Server для хранения информации о пользователях, приемах пищи и настройках.
* Используемые сторонние библиотеки: Newtonsoft.Json (для сериализации), System.Data.SQLite (для работы с БД).
* Безопасность: Для интерфейса предусмотрено шифрование данных. Это обеспечивает защиту данных и предотвращает взлом личной информации.

### ****5.3 Интерфейсы оборудования****

* Приложение работает с клавиатурой и мышью как основными средствами управления.
* Не требуется подключение специализированного оборудования.

### ****5.4 Коммуникационные интерфейсы****

* Приложение должно иметь возможность отправлять отчеты пользователю на электронную почту
* Поддержка обновления приложения через встроенную систему обновлений (при наличии интернет-соединения) — проверка версии на удаленном сервере.
* Скачивание обновленных таблиц продуктов и упражнений: По запросу приложение может загружать обновлённые базы данных (в формате JSON) с сервера разработчика

### ****6. Атрибуты качества****

### ****6.1 Удобство использования****

**Приоритет: высокий**

* Интерфейс должен быть интуитивно понятным и не требовать специального обучения.
* Все действия пользователя должны сопровождаться подсказками, уведомлениями об ошибках и возможностью отмены последних операций.
* Предусмотрена тёмная тема для приложения

### ****6.2 Производительность****

**Приоритет: средний**

* Приложение должно запускаться не более чем за 3 секунды на системе со средними характеристиками (4 ГБ RAM, двухъядерный процессор).
* Загрузка и фильтрация расписания из локальной БД — не более 1 секунды при объёме до 1000 записей.
* Максимальное потребление оперативной памяти — не более 200 МБ при нормальной эксплуатации.

### ****6.3 Безопасность****

**Приоритет: высокий**

* Пользовательские данные, такие как пароли и конфиденциальные данные, должны храниться в зашифрованном виде
* Доступ к данным должен быть возможен только после аутентификации пользователя.
* Все вводимые данные проходят валидацию для предотвращения SQL-инъекций и других уязвимостей.

### ****6.4 Техника безопасности****

**Приоритет: низкий**

* Приложение не должно вызывать сбоев системы или мешать работе других приложений.
* Все операции с файлами и локальной БД должны проходить через проверку путей и прав доступа.
* В случае аварийного завершения работы, приложение должно корректно сохранять промежуточные данные для предотвращения потери информации.

### ****6.5 Надёжность****

**Приоритет: высокий**

* Приложение должно сохранять пользовательские данные автоматически при изменении (например, при добавлении пищи).
* В случае сбоя программа должна корректно восстанавливаться и продолжать работу без потери данных.

**7.Требования по интернационализации и локализации**

Язык: Приложение должно поддерживать многоязычный пользовательский интерфейс, позволяя пользователю выбирать язык отображения. Базовая версия должна включать поддержку русского и английского языков. Все текстовые элементы интерфейса должны храниться в отдельных ресурсных файлах, что обеспечит удобство при добавлении новых языков

Форматы отображения даты, времени, чисел, валют и единиц измерения должны адаптироваться под региональные настройки операционной системы или вручную выбранную локаль. Например, формат даты должен отображаться как «DD.MM.YYYY» для русской локали и как «MM/DD/YYYY» для американской. Также должны корректно обрабатываться десятичные разделители (запятая или точка) и форматы времени (24-часовой и 12-часовой).

Также необходимо предусмотреть поддержку выбора единиц измерения: метрических (граммы, литры) и британских (унции, фунты), с возможностью автоматической конверсии при смене локали или настроек пользователя.

Все действия, связанные с планированием приёмов пищи и расписания, должны учитывать текущий часовой пояс устройства.

**8. Остальные требования**

* Юридические требования: соблюдение ФЗ-152 и GDPR; шифрование пользовательских данных; безопасный обмен по HTTPS.
* Установка: инсталлятор для Windows с проверкой системных требований (.NET), поддержка полного удаления.
* Первый запуск: настройка языка, часового пояса, единиц измерения.
* Поддержка и мониторинг: логирование ошибок, возможность выгрузки логов; предусмотрено автообновление с проверкой безопасности.
* Устойчивость: корректная обработка сбоев, информативные ошибки, восстановление состояния при аварийном завершении.

Приложение A. Словарь терминов

|  |  |
| --- | --- |
| **Термин** | **Определение** |
| БЖУ | Белки, жиры, углеводы. |
| UI | Пользовательский интерфейс. |
| GDPR | Общий регламент по защите данных в ЕС. |
| Локализация | Адаптация под региональные особенности. |
| Интернационализация | Подготовка к локализации. |
| Кэш | Временное хранилище данных. |
| UTC | Универсальное координированное время |

Приложение Б. Модели анализа

* ER-диаграмма — структура данных (Пользователь, Продукт, Приём пищи и т.д.).
* Диаграмма состояний — переходы между экранами.
* DFD — потоки данных от ввода до вывода отчёта.
* Дерево функций — структура основных функций приложения.