

Техническое задание на разработку сервиса «WallHub»

Версия 1.0

2025

История изменения документа

Версия	Дата	Авторы	Изменения
1.0	10.10.2025	Медведев Д.В., Анисимов Н.С.	Начало написания документа.

Содержание

- [Общие сведения](#)
- [Плановые сроки начала и окончания работы по созданию системы](#)
- [Назначение и цели создания системы](#)
- [Требования к системе](#)
 - [Функциональные требования](#)
 - [Нефункциональные требования](#)
- [Архитектура системы](#)
- [Модель данных системы](#)
- [Приложения](#)

1. Общие сведения

Система — настольное приложение «WallHub», сервис, который представляет собой платформу для поиска, управления и установки динамических и интерактивных обоев для рабочего стола. В отличие от стандартных средств ОС, WallHub позволяет использовать видеоролики, HTML5, приложения и 3D-сцены в качестве живых обоев, а также предоставляет обширную галерею контента от сообщества.

Заказчиком и исполнителем выступают Медведев Д.В. и Анисимов Н.С.

2. Плановые сроки начала и окончания работы по созданию системы

Этапы создания информационной системы с плановыми сроками представлены в таблице.

Этап	Сроки
Планирование и анализ требований	15 сентября — 28 сентября (2 недели)
Определение требований	29 сентября — 5 октября (1 неделя)
Проектирование системы	6 октября — 26 октября (3 недели)
Разработка системы	27 октября — 27 декабря (8 недель)
Тестирование системы	24 ноября — 21 декабря (4 недели)
Развертывание системы	12 января — 31 января (3 недели)

Стоимость разработки системы включает в себя заработную плату разработчику и затраты на инфраструктуру, общая стоимость составляет 250 000 рублей.

3. Назначение и цели создания системы

Система предназначена для предоставления пользователям мощного и удобного инструмента для персонализации рабочего стола с помощью динамических обоев.

Цели создания системы:

- Создать стабильное и производительное приложение, которое не оказывает значительного влияния на ресурсы системы.
- Предоставить пользователям интуитивно понятный интерфейс для установки и настройки обоев.
- Реализовать встроенную галерею (магазин) для просмотра, поиска и загрузки обоев, созданных сообществом.
- Обеспечить инструменты для создателей контента по публикации и распространению своих работ.
- Поддерживать различные типы обоев: видео, 3D, приложения, веб-страницы.

4. Требования к системе

4.1. Функциональные требования

【Основной функционал приложения】

- Приложение должно запускаться вместе с ОС и работать в фоновом режиме.
- Пользователь должен иметь возможность выбирать обои из локальной библиотеки или встроенной галереи.
- Пользователь должен иметь возможность настраивать параметры обоев (например, громкость видео, отзывчивость на системные события, потребление ресурсов).
- Пользователь должен иметь возможность приостанавливать работу обоев или устанавливать расписание их работы (например, отключать ночью).

【Управление библиотекой обоев】

- Пользователь должен иметь возможность импортировать собственные файлы обоев (видео, HTML, исполняемые файлы) в локальную библиотеку.
- Система должна автоматически определять тип обоев и назначать им соответствующие настройки.
- Пользователь должен иметь возможность создавать коллекции и плейлисты обоев для автоматической смены по времени или расписанию.

【Галерея / Магазин обоев】

- В приложении должен быть раздел для просмотра и поиска обоев, опубликованных сообществом.
- Пользователь должен иметь возможность фильтровать и сортировать обои (по рейтингу, дате, популярности, типу).
- Пользователь должен иметь возможность загружать и устанавливать обои напрямую из галереи в один клик.
- Пользователь должен иметь возможность оценивать и оставлять отзывы на обои.

【Система публикации обоев (для создателей)】

- Авторизованный пользователь должен иметь возможность загружать свои обои в галерею.

- При загрузке необходимо указать название, описание, теги, тип обоев, предварительно настроить параметры производительности и указать возрастной рейтинг.
- Загруженные обои проходят проверку на соответствие правилам платформы перед публикацией.

【Система пользователей】

- Пользователь должен иметь возможность создать учетную запись для синхронизации своих настроек и библиотеки между устройствами.
- Система должна поддерживать роли: Обычный пользователь, Создатель контента, Модератор.

4.2. Нефункциональные требования

【Требования к производительности】

- Потребление памяти приложением в фоновом режиме не должно превышать 100 МБ.
- Потребление ресурсов ЦП и GPU динамическими обоями должно быть настраиваемым пользователем.
- Приложение не должно влиять на производительность системных игр и приложений в полноэкранном режиме.

【Требования к безопасности】

- Все исполняемые обои (приложения, скрипты) должны запускаться в изолированной песочнице (sandbox) для защиты системы пользователя.
- Пароли пользователей должны храниться в зашифрованном виде.
- Загружаемые файлы должны проверяться антивирусным модулем.

【Требования к интерфейсу】

- **Стиль:** Модерновый, технологичный, с акцентами.
- **Цветовая схема:** Черно-оранжевая.
 - **Оранжевый (#FF6A00), используется для:**
 - Активных кнопок и призывов к действию.
 - Выделения выбранных элементов.
 - Акцентных линий и границ.
 - **Черный / Темно-серый (#121212, #1E1E1E), используется для:**
 - Фона интерфейса.
 - Панелей и карточек.

- **Градиент (от #FF6A00 к #FF3D00), используется для:**
 - Логотипа.
 - Кнопок главного меню.
 - Индикаторов загрузки.
- Интерфейс должен поддерживать светлую и темную темы, но темная тема является приоритетной и основной.
- Все элементы интерфейса должны быть выполнены с помощью современного UI-фреймворка для обеспечения единообразия и скорости разработки.

【Требования к совместимости】

- Приложение должно быть совместимо с ОС Windows 10 и Windows 11.
- Должна быть обеспечена поддержка многомониторных конфигураций.

5. Архитектура системы

【Основные сведения】

WallHub представляет собой настольное приложение, использующее клиент-серверную архитектуру для работы галереи и синхронизации данных.

【Компоненты архитектуры】

- **Клиентское приложение:** Разрабатывается на **C++** с использованием фреймворка **Qt** для рендеринга интерфейса и управления обоями. Для рендеринга сложных 3D-обоев будет использоваться **Unity Engine** как встраиваемый модуль.
- **Бэкенд-сервисы:** Реализуются на **Node.js** (или **Python Django**), обеспечивают работу галереи, аутентификацию и синхронизацию.
 - **REST API** для коммуникации между клиентом и сервером.
- **База данных:** Для хранения данных пользователей, информации об обоях и метаданных используется **PostgreSQL**.
- **Файловое хранилище:** Для хранения загруженных обоев (видео, архивы, ассеты) используется **S3-совместимое облачное хранилище**.
- **Микросервис модерации:** Отдельный сервис на **Python**, который проверяет загружаемые обои на соответствие правилам с помощью автоматического анализа метаданных и превью.

6. Модель данных системы

(Здесь будет размещена ER-диаграмма и описание основных сущностей, таких как User, Wallpaper, Tag, Category, Review, Download и т.д.)

7. Приложения

- **Приложение А: Цветовая схема** (Визуальное представление палитры #121212, #1E1E1E, #FF6A00, #FF3D00)
- **Приложение В: Макеты интерфейса** (Прототипы главного окна, галереи, настроек)
- **Приложение С: Технические требования к обоям** (Спецификации для создателей контента)