

**ПРАВИТЕЛЬСТВО РОССИЙСКОЙ ФЕДЕРАЦИИ
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ
«ВЫСШАЯ ШКОЛА ЭКОНОМИКИ»**

Факультет компьютерных наук
Образовательная программа «Программная инженерия»

СОГЛАСОВАНО

Приглашенный преподаватель ФКН
НИУ ВШЭ

_____ Н. И. Веселко
«__» _____ 2026 г.

УТВЕРЖДЕНО

Академический руководитель
образовательной программы
"Программная инженерия",
старший преподаватель департамента
программной инженерии

_____ Н. А. Павлочев
«__» _____ 2026 г.

**ГЕЙМИФИЦИРОВАННОЕ МОБИЛЬНОЕ
ПРИЛОЖЕНИЕ ДЛЯ УПРАВЛЕНИЯ ЗАДАЧАМИ И
ВИЗУАЛИЗАЦИИ ПРОГРЕССА - TASKORIA.**

Техническое задание

ЛИСТ УТВЕРЖДЕНИЯ

RU.17701729.06.05-01 ТЗ 01-1-ЛУ

Исполнители:

Студент группы БПИ247

_____/ В. А. Путинцева /

«__» _____ 2026 г.

Москва 2026

Инва.№ подп	Подп. и дата	Взам. инв.№	Инва.№ дубл.	Подп. и дата

УТВЕРЖДЕН

RU.17701729.06.05-01 ТЗ 01-1-ЛЮ

**ГЕЙМИФИЦИРОВАННОЕ МОБИЛЬНОЕ
ПРИЛОЖЕНИЕ ДЛЯ УПРАВЛЕНИЯ ЗАДАЧАМИ И
ВИЗУАЛИЗАЦИИ ПРОГРЕССА - TASKORIA**

Техническое задание

RU.17701729.06.05-01 ТЗ 01-1

Листов 39

Инов.№ подп	Подп. и дата	Взам. инв.№	Инов.№ дубл.	Подп. и дата

АННОТАЦИЯ

Техническое задание - это основной документ, оговаривающий набор требований и порядок создания программного продукта, в соответствии с которым производится разработка программы ее тестирование и приемка.

Настоящее Техническое задание на разработку «Геймифицированное мобильное приложение для управления задачами и визуализации прогресса - Taskoria» содержит следующие разделы: «Введение», «Основания для разработки», «Назначение разработки», «Требования к программе», «Требования к программной документации», «Технико-экономические показатели», «Стадии и этапы разработки», «Порядок контроля и приемки», приложения [7].

В разделе «Введение» указано наименование и краткая характеристика области применения программы.

В разделе «Основания для разработки» указан документ, на основании которого ведется разработка, и наименование темы разработки.

В разделе «Назначение разработки» указано функциональное и эксплуатационное назначение создаваемого программного продукта.

Раздел «Требования к программе» содержит указание на основные требования к функциональным характеристикам программы, к её надежности и к условиям эксплуатации, к составу и параметрам технических средств, к информационной и программной совместимости, к маркировке и упаковке, к транспортировке и хранению, а также специальные требования.

Раздел «Требования к программным документам» содержит указание на предварительный состав программной документации и специальные требования к ней.

Раздел «Технико-экономические показатели» содержит информацию об ориентировочной экономической эффективности разработки, экономические преимущества разработки программы.

Раздел «Стадии и этапы разработки» содержит информацию о стадиях разработки, этапах и содержании работ.

В разделе «Порядок контроля и приемки» указаны общие требования к приемке работы.

Настоящий документ разработан в соответствии с требованиями:

1. ГОСТ 19.101-77 [1]: Виды программ и программных документов.
2. ГОСТ 19.102-77 [2]: Стадии разработки.
3. ГОСТ 19.103-77 [3]: Обозначения программ и программных документов.

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.06.05-01 ТЗ 01-1				
Инв. № подл.	Подп. и дата	Взам. Инв. №	Инв. № дубл.	Подп. и дата

RU.17701729.06.05-01 ТЗ 01-1

4. ГОСТ 19.104-78 [4]: Основные надписи.
5. ГОСТ 19.105-78 [5]: Общие требования к программным документам.
6. ГОСТ 19.106-78 [6]: Требования к программным документам, выполненным печатным способом.
7. ГОСТ 19.201-78 [7]: Техническое задание.

Требования к содержанию и оформлению. Изменения к данному Техническому заданию оформляются согласно ГОСТ 19.603-78 [12], ГОСТ 19.604-78 [13].

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.06.05-01 ТЗ 01-1				
Инв. № подл.	Подп. и дата	Взам. Инв. №	Инв. № дубл.	Подп. и дата

СОДЕРЖАНИЕ

АННОТАЦИЯ.....	2
СОДЕРЖАНИЕ.....	4
ТЕРМИНЫ И ОПРЕДЕЛЕНИЯ.....	6
1. ВВЕДЕНИЕ	8
1.1. Наименование программы	8
1.2. Краткая характеристика области применения программы	8
2. ОСНОВАНИЯ ДЛЯ РАЗРАБОТКИ	9
2.1. Документ(ы), на основании которого(ых) ведется разработка	9
2.2. Наименование темы разработки	9
3. НАЗНАЧЕНИЕ РАЗРАБОТКИ	10
3.1. Функциональное назначение	10
3.2. Эксплуатационное назначение.....	10
4. ТРЕБОВАНИЯ К ПРОГРАММЕ	11
4.1. Требования к функциональным характеристикам	11
4.1.1. Требования к составу выполняемых функций	11
4.1.1.1. Общие требования к функциональным характеристикам	11
4.1.1.2. Состав компонентов программного продукта.....	11
4.1.1.3. Требования к серверной части.....	11
4.1.1.4. Требования к клиентскому приложению.....	12
4.1.2. Требования к организации входных данных	14
4.1.3. Требования к организации выходных данных	15
4.1.4. Требования к временным характеристикам	16
4.1.5. Требования к интерфейсу.....	17
4.2. Требования к надежности.....	18
4.3. Условия эксплуатации	19
4.3.1. Климатические условия эксплуатации	19
4.3.2. Требования к видам обслуживания	19
4.3.3. Требования к численности и квалификации персонала.....	19
4.4. Требования к составу и параметрам технических средств.....	20
4.4.1. Программные средства	20

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.06.05-01 ТЗ 01-1				
Инв. № подл.	Подп. и дата	Взам. Инв. №	Инв. № дубл.	Подп. и дата

4.4.2. Технические средства.....	20
4.5. Требования к информационной и программной совместимости.....	21
4.5.1. Требования к информационным структурам и методам решения.....	21
4.5.2. Требования к программным средствам, используемым программой.....	21
4.5.3. Требования к исходным кодам и языкам программирования.....	22
4.5.4. Требования к защите информации и программы.....	23
4.6. Требования к маркировке и упаковке.....	24
4.7. Требования к транспортированию и хранению.....	24
4.8. Специальные требования.....	24
5. ТРЕБОВАНИЯ К ПРОГРАММНОЙ ДОКУМЕНТАЦИИ.....	26
5.1. Состав программной документации.....	26
5.2. Специальные требования к программной документации.....	26
6. ТЕХНИКО-ЭКОНОМИЧЕСКИЕ ПОКАЗАТЕЛИ.....	27
6.1. Предполагаемая потребность.....	27
6.2. Целевая аудитория.....	27
6.3. Преимущества перед аналогами.....	27
7. СТАДИИ И ЭТАПЫ РАЗРАБОТКИ.....	31
7.1. Стадии разработки, этапы и содержание работ.....	31
7.2. Сроки разработки и исполнители.....	32
8. ПОРЯДОК КОНТРОЛЯ И ПРИЕМКИ.....	33
8.1. Виды испытаний.....	33
8.2. Общие требования к приемке работы.....	35
СПИСОК ИСПОЛЬЗУЕМОЙ ЛИТЕРАТУРЫ.....	37
ПРИЛОЖЕНИЕ. ССЫЛКИ НА АНАЛОГИ.....	38

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.06.05-01 ТЗ 01-1				
Инв. № подл.	Подп. и дата	Взам. Инв. №	Инв. № дубл.	Подп. и дата

ТЕРМИНЫ И ОПРЕДЕЛЕНИЯ

1. API (Application Programming Interface) - программный интерфейс приложения, набор протоколов и инструментов для создания программного обеспечения, определяющий способы взаимодействия различных компонентов системы.
2. Backend - серверная часть приложения, отвечающая за обработку запросов, бизнес-логику и взаимодействие с базой данных.
3. CRUD - аббревиатура от Create, Read, Update, Delete (создание, чтение, обновление, удаление), обозначающая базовые операции работы с данными.
4. Coins - игровая валюта в приложении, начисляемая пользователю за выполнение задач и используемая для постройки и улучшения зданий в виртуальном городе.
5. FastAPI - современный веб-фреймворк для создания API на языке Python, отличающийся высокой производительностью и автоматической генерацией документации.
6. Flutter - кроссплатформенный фреймворк для разработки мобильных приложений от компании Google, использующий язык программирования Dart.
7. Frontend - клиентская часть приложения, отвечающая за пользовательский интерфейс и взаимодействие с пользователем.
8. Геймификация - применение игровых механик и элементов игрового дизайна в неигровых контекстах для повышения мотивации и вовлеченности пользователей.
9. HTTP (HyperText Transfer Protocol) - протокол передачи гипертекста, используемый для обмена данными между клиентом и сервером в сети Интернет.
10. ИИ (Искусственный интеллект) - технология создания компьютерных систем, способных выполнять задачи, обычно требующие человеческого интеллекта, такие как понимание естественного языка, принятие решений и обучение.
11. JSON (JavaScript Object Notation) - текстовый формат обмена данными, основанный на синтаксисе JavaScript, широко используемый для передачи структурированных данных между клиентом и сервером.
12. JWT (JSON Web Token) - открытый стандарт для безопасной передачи информации в виде JSON-объекта между сторонами, используемый для аутентификации и авторизации пользователей.
13. LLM (Large Language Model) - большая языковая модель, тип искусственного

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.06.05-01 ТЗ 01-1				
Инв. № подл.	Подп. и дата	Взам. Инв. №	Инв. № дубл.	Подп. и дата

RU.17701729.06.05-01 ТЗ 01-1

- интеллекта, обученная на больших объемах текстовых данных для понимания и генерации естественного языка.
14. PostgreSQL - свободная объектно-реляционная система управления базами данных с открытым исходным кодом.
 15. Продуктивность - эффективность выполнения задач и достижения целей, измеряемая количеством и качеством выполненной работы за определенный период времени.
 16. RPG (Role-Playing Game) - ролевая игра, жанр игр, в котором игрок управляет персонажем в вымышленном мире, выполняя различные задания и развивая своего персонажа.
 17. Streak - серия последовательных дней активности пользователя, в течение которых он выполняет задачи без пропусков.
 18. Swagger - набор инструментов для проектирования, создания, документирования и тестирования RESTful API.
 19. Синхронизация - процесс согласования данных между клиентским приложением и сервером, обеспечивающий актуальность информации на всех устройствах пользователя.
 20. XP (Experience Points) - очки опыта, игровая метрика, начисляемая пользователю за выполнение задач и используемая для повышения уровня персонажа.
 21. Docker Compose - средство контейнерного запуска серверной части и базы данных.
 22. Mock-режим - режим работы клиентского приложения с локальными тестовыми данными без подключения к серверу.
 23. Footprint - область ячеек карты, занимаемая зданием или объектом.
 24. Изометрический город - визуальная игровая карта, на которой прогресс пользователя отображается через здания и декоративные объекты.
 25. Anchor-точка - точка изображения здания или объекта, которая привязывается к нижнему углу выбранной ячейки изометрической карты.

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.06.05-01 ТЗ 01-1				
Инв. № подл.	Подп. и дата	Взам. Инв. №	Инв. № дубл.	Подп. и дата

1. ВВЕДЕНИЕ

1.1. Наименование программы

Наименование программы - «Геймифицированное мобильное приложение для управления задачами и визуализации прогресса - Taskoria».

Наименование программы на английском языке - «Gamified mobile app for task management and progress visualization - Taskoria».

1.2. Краткая характеристика области применения программы

Программный продукт «Геймифицированное мобильное приложение для управления задачами и визуализации прогресса - Taskoria» - мобильное приложение, предназначенное для планирования и выполнения повседневных задач с элементами геймификации и игровой прогрессии. Приложение помогает пользователю фиксировать задачи, делить их на подзадачи, отслеживать выполнение, получать игровые награды и наблюдать прогресс через развитие виртуального изометрического города.

Приложение применяется в области управления личной продуктивностью, самоорганизации и планирования. Оно ориентировано на пользователей, испытывающих трудности с мотивацией к выполнению повседневных задач, включая студентов, школьников, молодых специалистов, а также любых людей, стремящихся улучшить привычки и снизить уровень прокрастинации.

В отличие от традиционных планировщиков, приложение использует игровую механику: уровни, достижения, игровые валюты, улучшение виртуального города. Дополнительным преимуществом является интеграция ИИ-ассистента, который автоматически разбивает задачи на подзадачи и определяет сложность, что снижает когнитивную нагрузку при планировании.

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.06.05-01 ТЗ 01-1				
Инв. № подл.	Подп. и дата	Взам. Инв. №	Инв. № дубл.	Подп. и дата

2. ОСНОВАНИЯ ДЛЯ РАЗРАБОТКИ

2.1. Документ, на основании которого ведется разработка

Разработка ведется на основании учебного плана подготовки бакалавров по направлению 09.03.04 «Программная инженерия» и утвержденной академическим руководителем программы темы курсового проекта.

2.2. Наименование темы разработки

Наименование темы разработки - «Геймифицированное мобильное приложение для управления задачами и визуализации прогресса - Taskoria».

Наименование темы разработки на английском языке - «Gamified mobile app for task management and progress visualization - Taskoria».

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.06.05-01 ТЗ 01-1				
Инв. № подл.	Подп. и дата	Взам. Инв. №	Инв. № дубл.	Подп. и дата

3. НАЗНАЧЕНИЕ РАЗРАБОТКИ

3.1. Функциональное назначение

Программа обеспечивает пользователю следующие функции:

- регистрацию и вход пользователя;
- создание, просмотр, редактирование, удаление и завершение задач;
- указание категории, приоритета, сложности и дедлайна задачи;
- автоматическое разбиение задач на подзадачи с помощью ИИ-ассистента;
- начисление XP и Coins за выполнение задач;
- отображение уровня пользователя, streak и краткой статистики;
- визуализацию прогресса через изометрический город;
- развитие зданий в зависимости от прогресса по категориям задач;
- покупку и размещение декоративных объектов из магазина;
- отображение достижений и счетчика открытых достижений;
- просмотр календаря активности;
- просмотр списка активных и завершенных задач;
- фильтрация задач по категориям (учеба, работа, здоровье, личное);

Программа позволяет повысить мотивацию пользователей к выполнению задач благодаря геймификации и визуальному отображению прогресса.

3.2. Эксплуатационное назначение

Программный продукт используется конечными пользователями на мобильных устройствах с операционной системой Android. Программа не требует специальных знаний или профессиональной подготовки, так как интерфейс ориентирован на массового пользователя.

Приложение предназначено для индивидуальной работы одного пользователя с личным аккаунтом. Для полноценной работы требуется подключение к сети Интернет для синхронизации данных с сервером и использования ИИ-функций. Для демонстрации и разработки клиентское приложение может быть запущено в mock-режиме без подключения к серверной части.

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.06.05-01 ТЗ 01-1				
Инв. № подл.	Подп. и дата	Взам. Инв. №	Инв. № дубл.	Подп. и дата

4. ТРЕБОВАНИЯ К ПРОГРАММЕ

4.1. Требования к функциональным характеристикам

4.1.1. Требования к составу выполняемых функций

4.1.1.1. Общие требования к функциональным характеристикам

- 4.1.1.1.1. Должна проверяться корректность всех вводимых данных, указанных в последующих пунктах п. 4.1.2, в соответствии с требованиями, указанными в разделе «Требования к организации входных данных».

4.1.1.2. Состав компонентов программного продукта

Программный продукт включает следующие компоненты:

1. Мобильное клиентское приложение для Android, разработанное на Flutter.
2. Серверная часть на FastAPI, предоставляющая REST API.
3. База данных PostgreSQL для хранения пользователей, задач, подзадач, игрового прогресса, достижений и покупок магазина.
4. Подсистема интеграции с GigaChat API для формирования подзадач.
5. Docker Compose конфигурация для локального контейнерного запуска серверной части и базы данных.

4.1.1.3. Требования к серверной части

Серверная часть должна обеспечивать:

1. Предоставлять API регистрации и входа пользователя.
2. Предоставлять API получения данных текущего пользователя
3. Хранение и CRUD-операции для всех сущностей: пользователи, задачи, подзадачи, здания, достижения, статистика активности, товары магазина и покупки.
4. Поддержку категорий задач: учёба, работа, здоровье, личное; приоритетов, уровней сложности и дедлайна.
5. API для задач: создание, получение, изменение, завершение, удаление.
6. API для подзадач: получение списка, создание, изменение, удаление, отметка о начале и завершении.
7. API для автоматического формирования подзадач через LLM.

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.06.05-01 ТЗ 01-1				
Инв. № подл.	Подп. и дата	Взам. Инв. №	Инв. № дубл.	Подп. и дата

RU.17701729.06.05-01 ТЗ 01-1

8. Начисление XP и Coins при завершении задачи (расчёт на основе сложности); обновление уровня пользователя, streak и статистики. Повторное начисление за уже завершённую задачу недопустимо.
9. API игрового города: состояние города (здания, уровни, декоративные объекты); обновление уровней зданий на основе прогресса по категориям задач.
10. API магазина: список товаров, покупка предмета (с проверкой достаточности Coins и без повторного списания), размещение купленного предмета на карте.
11. API достижений: список достижений и признаки их открытия.
12. Endpoint GET /health для проверки доступности сервера; корневой endpoint GET / с информацией о версии и ссылкой на документацию.
13. Автоматическую документацию API через Swagger/OpenAPI.
14. Возврат данных в формате JSON с HTTP-кодами состояния, соответствующими результату операции (200, 201, 400, 401, 403, 404, 500).

Ограничения бизнес-логики (реализованные в серверной части): выполненные задачи не удаляются и не редактируются (для сохранения истории и статистики); подзадачи завершённых или отменённых задач не создаются и не редактируются; выполненные подзадачи не удаляются и не редактируются.

4.1.1.4. Требования к клиентскому приложению

Клиентское приложение должно предоставлять:

1. Onboarding.
2. Экран авторизации и регистрации.
3. Экран изометрического города с отображением зданий и их уровней.
4. Отображение ратуши как экран краткой сводки профиля и прогресса.
5. Отображение декоративных объектов, купленных и размещенных пользователем.
6. Экран списка задач с фильтрацией по категориям и статусу.
7. Поддержка фильтрации задач по категории, приоритету, сложности и дедлайну.
8. Разделение активных и выполненных задач.

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.06.05-01 ТЗ 01-1				
Инв. № подл.	Подп. и дата	Взам. Инв. №	Инв. № дубл.	Подп. и дата

9. Экран создания/редактирования задачи.
10. Отображение подзадач и прогресса их выполнения.
11. Завершение задачи и получение визуальной обратной связи о награде.
12. Экран достижений с отображением полученных и доступных ачивок.
13. Экран календаря задач.
14. Экран магазина с товарами и ценами.
15. Экран локальных уведомлений о событиях приложения.
16. Экран настроек и выхода из аккаунта.
17. Экран профиля с информацией об уровне, XP, Coins, streak, краткой статистикой и счетчиком достижений.

4.1.1.5. Требования к изометрическому городу

Виртуальный город должен строиться на основе изометрической карты. Карта должна иметь условную сетку 28 x 28 ячеек. Дороги должны задаваться как набор дорожных блоков, а здания и декоративные объекты должны размещаться по заранее заданным спецификациям.

Для каждого здания или объекта должны задаваться:

- путь к графическому ресурсу;
- тип или категория объекта;
- уровень здания, если объект является зданием;
- размер вывода изображения;
- исходный размер изображения;
- координаты anchor-точки внутри изображения;
- координаты ячейки карты, к которой привязывается anchor-точка;
- footprint области, занимаемой объектом;
- параметры порядка отрисовки.

Позиция изображения должна вычисляться так, чтобы измеренная anchor-точка изображения совпадала с нижним углом заданной ячейки карты.

Порядок отрисовки объектов должен учитывать глубину расположения на карте, чтобы объекты, находящиеся визуально ниже, перекрывали объекты, находящиеся выше.

Для отдельных случаев допускается использование дополнительных параметров слоя.

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.06.05-01 ТЗ 01-1				
Инв. № подл.	Подп. и дата	Взам. Инв. №	Инв. № дубл.	Подп. и дата

Базовые здания города должны соответствовать категориям задач:

- учеба - школа;
- работа - офис;
- здоровье - госпиталь;
- личное - личное здание
- ратуша - сводка профиля и прогресса

4.1.2. Требования к организации входных данных

Входные данные включают:

1. Данные пользователя:
 - логин (email) и пароль при регистрации/авторизации (пароль не менее 8 символов);
 - имя пользователя (обязательное при регистрации, строка от 3 до 50 символов).
2. Данные задачи:
 - название задачи (обязательное поле, строка до 200 символов);
 - описание задачи (опциональное поле, текст до 1000 символов);
 - категория (обязательное поле: учеба, работа, здоровье, личное);
 - приоритет (обязательное поле: низкий, средний, высокий);
 - дедлайн (опциональное поле, дата).
 - сложность (легкая, средняя, сложная).
 - статус задачи, изменяемый при выполнении пользовательских действий.
3. Данные подзадачи:
 - название подзадачи (обязательное поле, строка до 200 символов);
 - описание (опциональное поле);
 - оценка времени выполнения (сформирована системой, в минутах);
 - порядок отображения (order_index);
 - статус подзадачи(не начата, выполнена);
4. Запросы к ИИ-подсистеме:
 - текст задачи для разбиения на подзадачи;
 - описание задачи и пользовательские параметры задачи;
5. Данные города:
 - идентификатор здания для постройки/апгрейда;

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.06.05-01 ТЗ 01-1				
Инв. № подл.	Подп. и дата	Взам. Инв. №	Инв. № дубл.	Подп. и дата

RU.17701729.06.05-01 ТЗ 01-1

- тип здания (школа, офис, госпиталь, личное здание);
- координаты и параметры размещения здания на изометрической карте;
- данные ратуши как сводки профиля и прогресса;

6. Данные магазина:

- идентификатор товара;
- тип товара;
- цена товара в Coins;
- действие покупки товара;
- действие размещения товара магазина;

Все входные данные должны проверяться на соответствие типам и ограничениям:

- обязательные поля не должны быть пустыми;
- формат email должен быть корректным;
- формат даты должен быть корректным;
- длина строк не должна превышать установленные лимиты;
- значения категорий, приоритетов, сложности и статусов должны соответствовать допустимым вариантам;
- покупка товара должна выполняться только при достаточном количестве Coins;
- размещение объекта магазина должно выполняться только после покупки товара;
- повторное завершение уже выполненной задачи не должно приводить к повторному начислению награды.

4.1.3. Требования к организации выходных данных

Выходные данные включают:

1. Данные пользователя:

- идентификатор пользователя;
- имя пользователя
- уровень пользователя;
- текущий XP и прогресс до следующего уровня;
- баланс Coins;
- текущий streak (количество дней подряд);
- список полученных достижений.

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.06.05-01 ТЗ 01-1				
Инв. № подл.	Подп. и дата	Взам. Инв. №	Инв. № дубл.	Подп. и дата

2. Данные задач:

- Список задач с полной информацией (название, описание, категория, приоритет, сложность, статус, дедлайн, награды XP и Coins);
- Список подзадач для задачи (название, описание, оценка времени в минутах, порядок, статус: не начата, выполнена);
- Результат завершения задачи (начисленные XP, Coins).

3. Данные города:

- Текущее состояние города (построенные здания, их уровни);
- Список зданий;
- Уровни зданий;
- Расположение зданий и декоративных объектов на изометрической карте;

4. Статистика:

- Общее количество выполненных задач;
- Активность пользователя по дням в календаре;
- История streak.

5. Достижения:

- Список всех достижений с описанием;
- Статус получения (получено/не получено);
- Награды за достижения.
- Количество открытых достижений.

Все выходные данные передаются в формате JSON через HTTP API. Ответы сервера содержат соответствующие HTTP-коды состояния (200 - успех, 400 - ошибка валидации, 401 - не авторизован, 404 - не найдено, 500 - ошибка сервера).

4.1.4. Требования к временным характеристикам

- Время отклика сервера на стандартные запросы получения и изменения данных при локальном запуске и стабильном соединении не должно превышать 500 мс;
- Время обработки запроса к ИИ-подсистеме для разбиения задачи на подзадачи не должно превышать 10 секунд при доступности внешнего сервиса;
- Время первичной загрузки основных экранов приложения на устройстве пользователя при стабильном соединении не должно превышать 2 секунд;

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.06.05-01 ТЗ 01-1				
Инв. № подл.	Подп. и дата	Взам. Инв. №	Инв. № дубл.	Подп. и дата

- Время обновления данных после создания, изменения, завершения задачи или покупки товара не должно превышать 1 секунды без учета задержек внешней сети;

4.1.5. Требования к интерфейсу

1. Пользовательский интерфейс должен быть интуитивно понятным и не требовать обучения.
2. Интерфейс должен поддерживать русский язык.
3. Все экраны должны быть адаптированы для работы на мобильных устройствах с различными размерами экранов (от 4.5 до 7 дюймов).
4. Интерфейс должен включать:
 - навигационное меню для перехода между основными разделами (задачи, город, календарь, магазин, профиль);
 - визуальное отображение игровых метрик (XP, уровень, Coins, streak);
 - анимации апгрейде здания;
 - визуализацию виртуального города с пиксельной графикой;
 - индикаторы прогресса для задач и уровня пользователя.
5. Интерфейс должен обеспечивать быстрый доступ к основным функциям:
 - создание новой задачи (не более 2 нажатий);
 - просмотр списка задач с фильтрацией;
 - переход к экрану города;
 - просмотр статистики.
6. Действия пользователя должны сопровождаться визуальной обратной связью.

Интерфейс должен включать следующие экраны:

1. Стартовый экран и onboarding.
2. Экран входа.
3. Экран регистрации.
4. Экран города.
5. Карточка здания.
6. Карточка ратуши.
7. Экран списка задач.
8. Экран фильтрации задач.
9. Экран создания задачи.
10. Экран редактирования задачи.

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.06.05-01 ТЗ 01-1				
Инв. № подл.	Подп. и дата	Взам. Инв. №	Инв. № дубл.	Подп. и дата

11. Экран детального просмотра задачи.
12. Экран календаря.
13. Экран магазина.
14. Экран достижений.
15. Экран профиля.
16. Экран уведомлений.
17. Экран настроек.

На экране создания задачи должны быть доступны поля названия, описания, выбора приоритета, категории, сложности и дедлайна. Карточки выбора должны быть визуально различимыми и не выходить за границы экрана.

На экране задач должны быть доступны поиск, переключение между активными и выполненными задачами, фильтрация и переход к карточке задачи.

На экране города пользователь должен видеть изометрическую карту, здания, декоративные объекты, панель профиля, Coins и кнопки перехода к основным разделам.

4.2. Требования к надежности

1. Устойчивость функционирования:

- программа должна обеспечивать стабильную работу при корректном использовании;
- клиентское приложение должно корректно обрабатывать ошибки сети и недоступность серверной части;
- при ошибках загрузки данных пользователь должен иметь возможность повторить запрос;
- критические операции изменения данных должны выполняться через серверную часть и базу данных.

2. Контроль входных данных:

- все входные данные должны проверяться на соответствие типам и ограничениям;
- обязательные поля должны быть валидированы перед отправкой на сервер или на стороне сервера;
- некорректные значения должны отклоняться с сообщением об ошибке;
- доступ к базе данных должен выполняться через ORM, что снижает риск SQL-инъекций;
- операции покупки, размещения объекта и завершения задачи должны проверять бизнес-ограничения.

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.06.05-01 ТЗ 01-1				
Инв. № подл.	Подп. и дата	Взам. Инв. №	Инв. № дубл.	Подп. и дата

3. Время восстановления:

- при временной недоступности серверной части клиентское приложение должно отображать состояние ошибки;
- после восстановления доступности сервера пользователь должен иметь возможность повторить загрузку данных;

4. Обработка ошибок:

- серверная часть должна логировать необработанные исключения и ошибки интеграции с внешними сервисами;
- пользователю должны предоставляться понятные сообщения или состояния ошибки;
- критические ошибки не должны приводить к аварийному завершению клиентского приложения;
- серверная часть должна возвращать HTTP-коды состояния, соответствующие результату обработки запроса.

4.3. Условия эксплуатации

4.3.1. Климатические условия эксплуатации

Требования к климатическим условиям отсутствуют, так как приложение работает на мобильных устройствах в стандартных условиях эксплуатации бытовой техники.

4.3.2. Требования к видам обслуживания

Обслуживание программы включает:

1. Обновление приложения через магазин приложений или APK-файл.
2. Обновление серверной части без прерывания работы (деплой через Docker).
3. Резервное копирование базы данных не реже одного раза в сутки.
4. Мониторинг работы сервера и ИИ-API для обеспечения стабильности.

4.3.3. Требования к численности и квалификации персонала

Приложение не требует специальной подготовки для использования. Требуемая квалификация пользователя - обычный пользователь мобильных приложений, умеющий работать с сенсорным экраном и базовыми функциями смартфона.

Для разработки и поддержки требуется:

- разработчик Backend (Python, FastAPI, PostgreSQL);
- разработчик Frontend (Dart, Flutter);
- DevOps-инженер для настройки серверной инфраструктуры (опционально).

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.06.05-01 ТЗ 01-1				
Инв. № подл.	Подп. и дата	Взам. Инв. №	Инв. № дубл.	Подп. и дата

4.4. Требования к составу и параметрам технических средств

4.4.1. Программные средства

Для работы клиентского приложения требуется:

- операционная система Android версии 8.0 (API level 26) или выше;
- подключение к сети Интернет для синхронизации данных и работы с ИИ-API.

Для работы серверной части требуется:

- Python 3.11 или выше;
- PostgreSQL версии 15 или выше;
- Docker и Docker Compose при контейнерном запуске.
- Alembic;
- Pydantic;
- Uvicorn;

Для разработки и запуска клиентской части необходимы:

- Flutter 3.0 или выше;
- Dart 3.0 или выше;
- Android SDK;
- Android-устройство или Android-эмулятор

4.4.2. Технические средства

Для работы клиентского приложения:

- смартфон с ОС Android 8.0 или выше;
- минимум 2 ГБ оперативной памяти;
- минимум 200 МБ свободного места на устройстве;
- подключение к сети Интернет или подключение к локальной сети, в которой доступна серверная часть;
- при запуске на физическом устройстве в режиме отладки допускается подключение по USB с использованием ADB

Для локального развертывания серверной части:

- персональный компьютер или ноутбук с ОС Windows, Linux или macOS;
- минимум 4 ГБ оперативной памяти;
- минимум 2 vCPU;
- минимум 5 ГБ свободного места на диске;
- установленный Python 3.11 для запуска backend без контейнеризации;

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.06.05-01 ТЗ 01-1				
Инв. № подл.	Подп. и дата	Взам. Инв. №	Инв. № дубл.	Подп. и дата

- установленный Docker Desktop и Docker Compose для контейнерного запуска backend и PostgreSQL;
- доступ к сети Интернет для установки зависимостей и обращения к GigaChat API.

Для демонстрационного запуска:

- backend может быть запущен локально на компьютере разработчика;
- PostgreSQL может быть запущен локально или в Docker-контейнере;
- Android-приложение может подключаться к backend через локальный IP-адрес компьютера, адрес эмулятора 10.0.2.2 или проброс порта через ADB;

4.5. Требования к информационной и программной совместимости

4.5.1. Требования к информационным структурам и методам решения

1. База данных:

- использование реляционной СУБД PostgreSQL для хранения структурированных данных;
- нормализация данных до 3NF для обеспечения целостности;
- использование индексов для оптимизации запросов;
- реализация транзакций для критических операций.
- наличие предопределённых достижений (seed-данные), загружаемых при инициализации приложения.

2. API:

- RESTful архитектура для взаимодействия клиента и сервера;
- префикс маршрутов API: /api (например, /api/tasks, /api/auth, /api/users);
- использование HTTP-методов (GET, POST, PATCH, DELETE) согласно их семантике;
- формат обмена данными - JSON;
- документирование API через Swagger.

3. ИИ-интеграция:

- использование GigaChat API для разбиения пользовательских задач на подзадачи и назначения награды за задачу;
- возможность замены ИИ-провайдера (Strategy Pattern) для использования другого LLM при необходимости;

4.5.2. Требования к программным средствам, используемым программой

1. Backend:

- Python 3.11 - язык программирования серверной части;
- FastAPI - веб-фреймворк для создания API;

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.06.05-01 ТЗ 01-1				
Инв. № подл.	Подп. и дата	Взам. Инв. №	Инв. № дубл.	Подп. и дата

RU.17701729.06.05-01 ТЗ 01-1

- SQLAlchemy - ORM для работы с базой данных;
- Pydantic - валидация данных;
- Python-dotenv - управление переменными окружения;
- GigaChat SDK - интеграция с ИИ-API;
- Uvicorn - ASGI-сервер для запуска приложения;
- Alembic - миграции базы данных.
- Pytest - автоматизированное тестирование серверной части.

2. Frontend:

- Flutter SDK версии 3.0 или выше;
- Dart версии 3.0 или выше;
- GoRouter - маршрутизация между экранами;
- HTTP-клиент для взаимодействия с API;
- State management (Provider, Riverpod);

3. База данных:

- PostgreSQL 15+;

4. Инфраструктура:

- Docker и Docker Compose для контейнерного запуска backend и PostgreSQL;
- Git для контроля версий;
- Swagger/OpenAPI для просмотра и проверки серверного API;

4.5.3. Требования к исходным кодам и языкам программирования**1. Языки программирования:**

- Python 3.11 - для серверной части (Backend);
- Dart - для клиентской части (Frontend);
- SQL - для работы с базой данных PostgreSQL.

2. Стандарты кодирования:

- Python: PEP 8;
- Dart: официальные рекомендации Dart Style Guide;
- использование type hints в Python;
- использование строгой типизации в Dart.
- код должен быть структурирован по слоям и функциональным областям.

3. Структура проекта:

- разделение на модули по функциональности;
- следование принципам SOLID;

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.06.05-01 ТЗ 01-1				
Инв. № подл.	Подп. и дата	Взам. Инв. №	Инв. № дубл.	Подп. и дата

RU.17701729.06.05-01 ТЗ 01-1

- серверная часть должна включать слои API, схем, моделей, репозитория, сервисов, фабрик, строителей и стратегий;
- клиентская часть должна включать слои экранов, моделей, провайдеров, сетевого клиента, управления сессией и общих виджетов;
- крупные спецификации изометрической карты, зданий и объектов магазина должны быть отделены от кода отрисовки карты;
- использование паттернов проектирования: Repository (доступ к данным), Strategy (выбор ИИ-провайдера), Template Method (расчёт наград), State (управление streak), Observer (Event Bus для событий), Factory (создание зданий), Builder (построение города), Facade (сложные операции), Dependency Injection, принципы DDD (агрегаты, доменные сущности)

4. Версионирование:

- исходный код должен храниться в репозитории Git;
- публичная версия исходного кода должна быть размещена на GitHub;

4.5.4. Требования к защите информации и программы**1. Аутентификация и авторизация:**

- использование JWT (JSON Web Tokens) для аутентификации пользователей;
- выдача токена после успешного входа пользователя;
- проверка токена при выполнении защищенных запросов;
- автоматический переход клиента в состояние неавторизованного пользователя при получении ответа 401 Unauthorized;
- очистка локального состояния сессии при выходе пользователя из аккаунта.

2. Защита данных:

- валидация всех входных данных на сервере;
- защита от SQL-инъекций за счет использования ORM и параметризованного доступа к данным;
- хранение секретных ключей, токенов внешних API и параметров подключения к базе данных в переменных окружения;
- недопущение хранения секретов в исходном коде и публичном репозитории;
- для локального демонстрационного запуска допускается использование HTTP;

3. Безопасность API:

- сервер должен возвращать HTTP-коды ошибок, соответствующие причине отказа;
- CORS-политика должна настраиваться на стороне серверной части;

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.06.05-01 ТЗ 01-1				
Инв. № подл.	Подп. и дата	Взам. Инв. №	Инв. № дубл.	Подп. и дата

- операции завершения задачи, покупки товара и размещения объекта должны проверять бизнес-ограничения;
- использование переменных окружения для хранения секретных ключей (API keys, токены);
- недопущение хранения секретов в исходном коде.

4. Защита персональных данных:

- приложение должно собирать минимальный набор пользовательских данных, необходимый для регистрации и работы с задачами;
- пользовательские данные должны быть доступны только после авторизации;

4.6. Требования к маркировке и упаковке

Программный продукт передается в виде исходного кода, отчетной документации, ссылки на GitHub-репозиторий и инструкции по локальному запуску.

Клиентская часть передается в виде исходного кода Flutter-приложения. Установка и запуск Android-приложения выполняются локально через Flutter SDK на физическом Android-устройстве или Android-эмуляторе.

Серверная часть передается в виде исходного кода FastAPI-приложения, файла примера переменных окружения и Docker Compose-конфигурации для локального запуска backend и PostgreSQL.

Публичное серверное развертывание, публикация приложения в магазине приложений и обязательная передача APK-файла в рамках текущей версии проекта не предусматриваются.

Специальные требования к физической упаковке отсутствуют, так как программный продукт распространяется в цифровом виде.

4.7. Требования к транспортированию и хранению

Программа хранится в репозитории GitHub. Исходный код должен быть доступен для просмотра и клонирования. В репозитории должны храниться:

- исходный код серверной части;
- исходный код клиентской части;
- README-файлы с командами запуска;
- Docker Compose-конфигурация;
- примеры переменных окружения без реальных секретов;
- тесты backend и frontend.

4.8. Специальные требования

1. Все документы должны соответствовать ГОСТ 19.106-78.

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.06.05-01 ТЗ 01-1				
Инв. № подл.	Подп. и дата	Взам. Инв. №	Инв. № дубл.	Подп. и дата

RU.17701729.06.05-01 ТЗ 01-1

2. Все материалы проекта должны быть загружены в систему LMS SmartLMS.
3. Приложение должно поддерживать локальный демонстрационный запуск backend и подключение Android-клиента к локально запущенной серверной части.
4. ИИ-подсистема должна корректно обрабатывать задачи на русском языке и возвращать подзадачи в структуре, пригодной для отображения в мобильном приложении.
5. Визуальный стиль города должен быть выполнен с использованием pixel-art ресурсов для обеспечения единого игрового восприятия.
6. Изометрическая карта должна поддерживать точное размещение зданий и декоративных объектов через anchor-точки, footprint и координаты ячеек.
7. Документация API должна быть доступна через Swagger UI при запущенной серверной части.
8. README-файлы должны содержать команды запуска backend, frontend, Docker Compose, mock режима и core_api-режима.

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.06.05-01 ТЗ 01-1				
Инв. № подл.	Подп. и дата	Взам. Инв. №	Инв. № дубл.	Подп. и дата

5. ТРЕБОВАНИЯ К ПРОГРАММНОЙ ДОКУМЕНТАЦИИ

5.1. Состав программной документации

1. «Геймифицированное мобильное приложение для управления задачами и визуализации прогресса - Taskoria». Техническое задание (ГОСТ 19.201-78 [7]).
2. «Геймифицированное мобильное приложение для управления задачами и визуализации прогресса - Taskoria». Пояснительная записка (ГОСТ 19.404-79 [10]).
3. «Геймифицированное мобильное приложение для управления задачами и визуализации прогресса - Taskoria». Программа и методика испытаний (ГОСТ 19.301-79[8]).
4. «Геймифицированное мобильное приложение для управления задачами и визуализации прогресса - Taskoria». Текст программы (ГОСТ 19.401-78 [9]).
5. «Геймифицированное мобильное приложение для управления задачами и визуализации прогресса - Taskoria». Руководство оператора (ГОСТ 19.505-79 [11]).

5.2. Специальные требования к программной документации

Документы должны соответствовать ГОСТ 19.106-78. Все материалы подлежат загрузке в LMS SmartLMS. Документация должна быть написана на русском языке с использованием терминологии, соответствующей области применения программы.

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.06.05-01 ТЗ 01-1				
Инв. № подл.	Подп. и дата	Взам. Инв. №	Инв. № дубл.	Подп. и дата

6. ТЕХНИКО-ЭКОНОМИЧЕСКИЕ ПОКАЗАТЕЛИ

6.1. Предполагаемая потребность

Потребность в разработке приложения обусловлена отсутствием на рынке гибридного продукта, сочетающего функции планировщика задач, элементы геймификации и интеграцию ИИ-ассистента для автоматического разбиения задач.

Существующие аналоги (Habitica, Forest и др.) либо не используют ИИ для автоматизации планирования, либо не имеют развитой системы визуального развития виртуального города, что ограничивает их эффективность в мотивации пользователей.

Целевая аудитория включает студентов, школьников, молодых специалистов и всех людей, испытывающих трудности с мотивацией к выполнению повседневных задач. По оценкам, потенциальная аудитория составляет миллионы пользователей в русскоязычном сегменте рынка мобильных приложений.

6.2. Целевая аудитория

Целевая аудитория приложения включает:

1. Студенты высших учебных заведений - для планирования учебных задач, подготовки к экзаменам, выполнения курсовых работ.
2. Школьники - для организации домашних заданий и внеучебной деятельности.
3. Молодые специалисты - для управления рабочими задачами и личными проектами.
4. Люди, испытывающие проблемы с прокрастинацией - для повышения мотивации через геймификацию.
5. Пользователи, интересующиеся продуктивностью и саморазвитием - для отслеживания прогресса и формирования полезных привычек.

Общие характеристики целевой аудитории:

- возраст от 14 до 35 лет;
- активные пользователи смартфонов;
- знакомы с базовыми функциями мобильных приложений;
- заинтересованы в игровых механиках и визуальном прогрессе.

6.3. Преимущества перед аналогами

Основные преимущества приложения «Геймифицированное мобильное приложение для управления задачами и визуализации прогресса - Taskoria» перед существующими аналогами:

1. Уникальное сочетание ИИ + продуктивность + развитие виртуального города - ни один существующий продукт не объединяет все три компонента. В отличие от Habitica, Do It Now, LifeUp и других приложений, наше приложение использует искусственный

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.06.05-01 ТЗ 01-1				
Инв. № подл.	Подп. и дата	Взам. Инв. №	Инв. № дубл.	Подп. и дата

интеллект для автоматизации планирования.

2. Автоматическое разбиение задач на подзадачи с помощью ИИ снижает когнитивную нагрузку при планировании и помогает пользователю лучше структурировать работу. Ни одно из анализируемых приложений (Habitica, Forest, Avocation, Habit Hunter, Do It Now, LifeUp, Todoist Karma) не предоставляет такой функциональности.
3. Визуальная мотивация через развитие виртуального города - пользователь видит прямой результат своей продуктивности в виде растущего и развивающегося виртуального города. В отличие от Avocation (рост растения), Forest (выращивание дерева) или других приложений, наш город предоставляет более глубокую систему развития с различными типами зданий, привязанными к категориям задач.
4. Интеграция категорий задач с типами зданий - создает дополнительную связь между реальными задачами и игровым прогрессом. Например, выполнение задач в категории «Учеба» развивает школу, задачи «Здоровье» - госпиталь. Такая связь отсутствует у всех конкурентов.
5. Адаптивная система наград на основе ИИ-анализа сложности - обеспечивает справедливое вознаграждение за выполнение задач различной сложности. В отличие от Habitica, Do It Now, LifeUp, где награды фиксированы или определяются пользователем, наше приложение автоматически рассчитывает оптимальные награды.
6. Система предоставляет аналитику продуктивности с календарем активности, статистику по задачам.
7. Отсутствие социального давления - в отличие от Habitica с её системой групп и квестов, наше приложение фокусируется на индивидуальном прогрессе без необходимости участия в социальных активностях, что снижает стресс для пользователей, испытывающих проблемы с прокрастинацией.
8. Уникальная визуализация прогресса - виртуальный город с пиксельной графикой предоставляет более наглядное и привлекательное отображение прогресса по сравнению с простыми графиками в Todoist Karma или базовыми визуализациями в других приложениях.
9. Гибкая система категорий - четыре основные категории (учеба, работа, здоровье, личное) с привязкой к типам зданий создают более структурированный подход к организации задач по сравнению с универсальными системами в Do It Now или LifeUp.
10. Повышенная удерживаемость пользователей благодаря комбинации геймификации и практической пользы - сочетание игровых механик с реальной пользой для

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.06.05-01 ТЗ 01-1				
Инв. № подл.	Подп. и дата	Взам. Инв. №	Инв. № дубл.	Подп. и дата

RU.17701729.06.05-01 ТЗ 01-1

продуктивности обеспечивает более высокую мотивацию и долгосрочное использование по сравнению с приложениями, фокусирующимися только на одной из этих сторон.

Функция	Taskoria	Habitica	Forest	Avocation	Habit Hunter	Do It Now	LifeUp	Todoist Karma
ИИ-разбиение задач на подзадачи	+	-	-	-	-	-	-	-
Виртуальный город с развитием	+	-	-	-	-	-	-	-
Система уровней и XP	+	+	-	-	+	+	+	+
Игровая валюта	+	+	-	-	+	+	+	-
Streak-система	+	+	+	-	+	+	±	+
Достижения	+	+	-	-	+	+	+	-
Статистика продуктивности	+	+	±	±	±	+	+	+
Категории задач	+	+	-	+	+	+	+	+
Подзадачи	+	+	-	-	+	+	-	+
Календарь/планирование	+	±	-	+	+	+	+	+
Напоминания	+	+	+	+	+	+	+	+
Визуализация прогресса	+	+	+	+	+	+	+	+

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.06.05-01 ТЗ 01-1				
Инв. № подл.	Подп. и дата	Взам. Инв. №	Инв. № дубл.	Подп. и дата

RU.17701729.06.05-01 ТЗ 01-1

Социальные функции	+	+	-	-	-	-	+	-
Адаптивная система наград	+	±	-	-	±	±	±	±
Итого	14	10	3	4	9	10	9	8

Таблица 1. Сравнение функциональных характеристик

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.06.05-01 ТЗ 01-1				
Инв. № подл.	Подп. и дата	Взам. Инв. №	Инв. № дубл.	Подп. и дата

7. СТАДИИ И ЭТАПЫ РАЗРАБОТКИ

7.1. Стадии разработки, этапы и содержание работ

Стадии и этапы разработки были выявлены с учетом ГОСТ 19.102-77 [2].

Таблица 2 – Стадии и этапы разработки

Стадия разработки	Этап работ	Содержание работ	Сроки выполнения
Проектирование	Анализ конкурентов	Анализ конкурентов (Habitica, Forest, EpicWin, SimCity и др.), формирование концепции приложения	10.11.25 - 23.11.25
	Проектирование базы данных	Проектирование базы данных, определение структуры данных	24.11.25 - 30.11.25
	API и прототип	Определение API-структуры, создание Figma-прототипа	1.12.25 - 17.12.25
Разработка Backend	Базовая архитектура и CRUD	Создание базовой архитектуры проекта (FastAPI+ PostgreSQL + SQLAlchemy), модель пользователя, CRUD задач	18.12.25 - 15.01.26
	Игровая механика и ИИ	Механика начисления XP и Coins, интеграция ИИ-API для разбиения задач	16.01.26 - 23.01.26
	API достижений и статистики	API для достижений, статистики, streak-системы, документация Swagger	24.01.26 – 31.01.26
Разработка Frontend	Основные экраны	Разработка экранов: задачи, город, статистика, профиль, авторизация	1.02.26 - 20.02.26
	Интеграция с API	Интеграция с API: получение и обновление данных, локальное сохранение	21.02.26 – 28.03.26
	Визуализация города	Отображение города и уровня развития пользователя, UI-анимации	1.03.26 - 26.03.26
Интеграция	Связь клиента и сервера	Связь клиента и сервера, тестирование взаимодействия, устранение ошибок	27.03.26 - 10.04.26
Тестирование	Функциональное тестирование	Функциональное тестирование всех модулей, проверка ИИ и разбиения задач	11.04.26 - 18.04.26
	UX-полировка	Финальное оформление интерфейса, UX-полировка, оптимизация производительности	19.04.26 - 3.05.26
Подготовка проекта	Финальная подготовка	Презентация, финальная корректировка документации, подготовка к защите	4.05.26 - 11.05.26

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.06.05-01 ТЗ 01-1				
Инв. № подл.	Подп. и дата	Взам. Инв. №	Инв. № дубл.	Подп. и дата

7.2. Сроки разработки и исполнители

Программный продукт должен быть завершен не позднее утвержденного срока загрузки программы и программной документации в Smart LMS.

Исполнитель - Путинцева Вера Александровна

Критерии приемки каждого этапа:

- выполнение всех задач этапа согласно содержанию работ;
- отсутствие критических ошибок;
- соответствие кода стандартам программирования;
- наличие необходимой документации для этапа.

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.06.05-01 ТЗ 01-1				
Инв. № подл.	Подп. и дата	Взам. Инв. №	Инв. № дубл.	Подп. и дата

8. ПОРЯДОК КОНТРОЛЯ И ПРИЕМКИ

8.1. Виды испытаний

1. Функциональное тестирование:

- тестирование регистрации, авторизации и выхода пользователя из аккаунта;
- проверка создания, просмотра, редактирования, удаления и завершения задач;
- проверка работы с подзадачами: получение списка, создание, запуск, завершение и удаление;
- проверка корректности начисления XP и Coins при выполнении задач;
- проверка защиты от повторного завершения задачи с повторным начислением награды;
- тестирование системы уровней и прогрессии пользователя;
- проверка работы streak-системы;
- проверка отображения краткой статистики профиля;
- проверка фильтрации задач по категории, приоритету, сложности и дедлайну;
- проверка поиска задач;
- проверка системы достижений и счетчика открытых достижений в профиле;
- тестирование функционала города: отображение зданий, уровней зданий и ратуши;
- проверка магазина: отображение товаров, покупка товара, отказ при недостатке Coins;
- проверка размещения купленных декоративных объектов на изометрической карте;
- проверка календаря активности.

2. Тестирование ИИ-подсистемы:

- проверка корректности разбиения задач на подзадачи;
- проверка обработки задач на русском языке;
- проверка сохранения сформированных подзадач;
- проверка поведения системы при недоступности ИИ-сервиса;
- проверка fallback-механизма при ошибке внешнего API.

3. Тестирование API:

- проверка соответствия API документации Swagger;
- тестирование эндпоинтов авторизации;
- тестирование эндпоинтов задач и подзадач;

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.06.05-01 ТЗ 01-1				
Инв. № подл.	Подп. и дата	Взам. Инв. №	Инв. № дубл.	Подп. и дата

RU.17701729.06.05-01 ТЗ 01-1

- тестирование эндпоинтов пользователя и профиля;
- тестирование эндпоинтов города;
- тестирование эндпоинтов магазина;
- проверка корректности HTTP-кодов ответов;
- тестирование валидации входных данных;
- проверка обработки ошибок;
- проверка доступа к защищенным эндпоинтам только при наличии корректного JWT-токена.

4. Тестирование интерфейса:

- проверка отображения основных экранов приложения на Android-устройствах с разными размерами экрана;
- тестирование навигации между экранами через нижнее меню и внутренние переходы;
- проверка экрана входа и регистрации;
- проверка экрана списка задач, фильтрации и поиска;
- проверка экрана создания и редактирования задачи;
- проверка экрана детального просмотра задачи;
- проверка визуализации изометрического города;
- проверка карточек зданий и ратуши;
- проверка экрана магазина;
- проверка экрана достижений;
- проверка экрана профиля;
- проверка календаря активности;
- проверка отображения состояний загрузки, пустого результата и ошибки.

5. Тестирование производительности:

- проверка времени отклика сервера на стандартные запросы;
- проверка времени обработки ИИ-запросов;
- проверка поведения приложения при временной недоступности серверной части;
- проверка отсутствия аварийного завершения приложения при ошибках сети;
- проверка корректного повторного запроса данных после ошибки загрузки.

6. Тестирование безопасности:

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.06.05-01 ТЗ 01-1				
Инв. № подл.	Подп. и дата	Взам. Инв. №	Инв. № дубл.	Подп. и дата

- тестирование аутентификации и авторизации;
- проверка обработки запросов без JWT-токена;
- проверка очистки локальной сессии при выходе пользователя;
- проверка валидации входных данных на сервере;
- проверка снижения риска SQL-инъекций за счет использования ORM;
- проверка отсутствия секретных ключей и файлов окружения в публичном репозитории.

8.2. Общие требования к приемке работы

Программа принимается при выполнении следующих условий:

1. Корректная работа всех функций, указанных в техническом задании:

- регистрация, авторизация и выход пользователя;
- создание, просмотр, редактирование, удаление и завершение задач;
- работа с подзадачами;
- разбиение задач на подзадачи через ИИ-подсистему;
- начисление XP и Coins;
- система уровней и прогрессии;
- streak-система;
- визуализация изометрического города;
- отображение уровней зданий;
- отображение ратуши как сводки профиля и прогресса;
- система достижений;
- магазин декоративных объектов;
- покупка и размещение объектов на карте;
- календарь активности;
- профиль пользователя и краткая статистика.

2. Успешное прохождение всех видов испытаний, указанных в разделе «Виды испытаний».

3. Наличие всей программной документации:

- техническое задание (ГОСТ 19.201-78);
- пояснительная записка (ГОСТ 19.404-79);
- программа и методика испытаний (ГОСТ 19.301-79);

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.06.05-01 ТЗ 01-1				
Инв. № подл.	Подп. и дата	Взам. Инв. №	Инв. № дубл.	Подп. и дата

RU.17701729.06.05-01 ТЗ 01-1

- текст программы (ГОСТ 19.401-78);
 - руководство оператора (ГОСТ 19.505-79).
4. Отсутствие критических ошибок, препятствующих нормальной работе приложения.
 5. Соответствие кода стандартам программирования и наличие комментариев в критических местах.
 6. Наличие документации API (Swagger) с описанием всех эндпоинтов.
 7. Успешный локальный запуск проекта:
 - запуск backend локально или через Docker Compose;
 - запуск PostgreSQL локально или через Docker Compose;
 - запуск Android-приложения на физическом устройстве или эмуляторе;
 - подключение Android-приложения к локальной серверной части.
 8. Успешное прохождение автоматизированных проверок:
 - backend-тесты Pytest;
 - flutter analyze;
 - flutter test.
 9. Загрузка всех материалов проекта в систему LMS SmartLMS.

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.06.05-01 ТЗ 01-1				
Инв. № подл.	Подп. и дата	Взам. Инв. №	Инв. № дубл.	Подп. и дата

СПИСОК ИСПОЛЬЗУЕМОЙ ЛИТЕРАТУРЫ

1. ГОСТ 19.101-77: Виды программ и программных документов. // Единая система программной документации. – М.: ИПК Издательство стандартов, 2001.
2. ГОСТ 19.102-77: Стадии разработки. // Единая система программной документации. – М.: ИПК Издательство стандартов, 2001.
3. ГОСТ 19.103-77: Обозначения программ и программных документов. // Единая система программной документации. – М.: ИПК Издательство стандартов, 2001.
4. ГОСТ 19.104-78: Основные надписи. // Единая система программной документации. – М.: ИПК Издательство стандартов, 2001.
5. ГОСТ 19.105-78: Общие требования к программным документам. // Единая система программной документации. – М.: ИПК Издательство стандартов, 2001.
6. ГОСТ 19.106-78: Требования к программным документам, выполненным печатным способом. // Единая система программной документации. – М.: ИПК Издательство стандартов, 2001.
7. ГОСТ 19.201-78: Техническое задание. Требования к содержанию и оформлению. // Единая система программной документации. – М.: ИПК Издательство стандартов, 2001.
8. ГОСТ 19.301-79: Программа и методика испытаний. Требования к содержанию и оформлению. // Единая система программной документации. – М.: ИПК Издательство стандартов, 2001.
9. ГОСТ 19.401-78: Текст программы. Требования к содержанию и оформлению. // Единая система программной документации. – М.: ИПК Издательство стандартов, 2001.
10. ГОСТ 19.404-79: Пояснительная записка. Требования к содержанию и оформлению. // Единая система программной документации. – М.: ИПК Издательство стандартов, 2001.
11. ГОСТ 19.505-79: Руководство оператора. Требования к содержанию и оформлению. // Единая система программной документации. – М.: ИПК Издательство стандартов, 2001.
12. ГОСТ 19.603-78: Общие правила внесения изменений. // Единая система программной документации. – М.: ИПК Издательство стандартов, 2001.
13. ГОСТ 19.604-78: Правила внесения изменений в программные документы, выполненные печатным способом. // Единая система программной документации. – М.: ИПК Издательство стандартов, 2001.

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.06.05-01 ТЗ 01-1				
Инв. № подл.	Подп. и дата	Взам. Инв. №	Инв. № дубл.	Подп. и дата

ПРИЛОЖЕНИЕ. ССЫЛКИ НА АНАЛОГИ

Приложение	Ссылка
Habitica	https://habitica.com/
Forest	https://www.forestapp.cc/
Avocation	https://avocation.app/
Habit Hunter	https://habithunter.net/
Do It Now	https://do-it-now.app/
LifeUp	https://www.lifeupapp.fun/en/index.html
Todoist Karma	https://www.todoist.com/ru/karma

Дата обращения: 01.02.26.

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.06.05-01 ТЗ 01-1				
Инв. № подл.	Подп. и дата	Взам. Инв. №	Инв. № дубл.	Подп. и дата

ЛИСТ РЕГИСТРАЦИИ ИЗМЕНЕНИЙ

[illegible]