

**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. Павлов
«__» _____ 2026 г.

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

Программа и методика испытаний

ЛИСТ УТВЕРЖДЕНИЯ

RU.17701729.06.05-01 51 01-1-ЛУ

Исполнитель:

Студент группы БПИ247

_____/ В. А. Путинцева /

«__» _____ 2026 г.

Подп. и дата	
Инв.№ дубл.	
Взам. инв.№	
Подп. и дата	
Инв.№ подл	

УТВЕРЖДЕН

RU.17701729.06.05-01 51 01-1-ЛУ

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

Программа и методика испытаний

RU.17701729.06.05-01 51 01-1

Листов 31

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

АННОТАЦИЯ

Настоящая программа и методика испытаний определяет порядок проверки программного продукта «Геймифицированное мобильное приложение для управления задачами и визуализации прогресса - Taskoria».

Документ содержит объект и цель испытаний, требования к проверяемой программе, состав программной документации, средства испытаний, порядок проведения проверок и методику испытаний основных функций: авторизации, задач, подзадач, ИИ-разбиения, начисления наград, города, магазина, достижений, календаря, профиля, Swagger API и локального запуска через Docker Compose.

Документ разработан с учетом требований ГОСТ 19.301-79 «Программа и методика испытаний. Требования к содержанию и оформлению».

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.06.05-01 51 01-1				
Инв. № подл.	Подп. и дата	Взам. Инв. №	Инв. № дубл.	Подп. и дата

СОДЕРЖАНИЕ

АННОТАЦИЯ.....	2
СОДЕРЖАНИЕ.....	3
1. ОБЪЕКТ ИСПЫТАНИЙ.....	5
1.1. Наименование программы.....	5
1.2. Краткая характеристика области применения	5
2. ЦЕЛЬ ИСПЫТАНИЙ.....	6
3. ТРЕБОВАНИЯ К ПРОГРАММЕ	7
3.1. Требования к функциональным характеристикам.....	7
3.1.1. Состав выполняемых функций.....	7
3.1.1.1.Регистрация и авторизация пользователя	7
3.1.1.2.Управление задачами.....	7
3.1.1.3.Работа с подзадачами	7
3.1.1.4.Завершение задач и начисление наград.....	7
3.1.1.5.Изометрический город	9
3.1.1.6.Магазин.....	9
3.1.1.7.Достижения, календарь и профиль	9
3.1.2. Организация входных данных	9
3.1.3. Организация выходных данных.....	9
3.2. Требования к временным характеристикам.....	10
3.3. Требования к интерфейсу	10
3.4. Требования к надежности.....	10
3.5. Условия эксплуатации	10
3.6. Требования к составу и параметрам технических средств.....	11
3.7. Требования к информационной и программной совместимости	11
3.8. Требования к транспортировке и упаковке.....	11
4. ТРЕБОВАНИЯ К ПРОГРАММНОЙ ДОКУМЕНТАЦИИ.....	12
4.1. Состав программной документации	12
4.2. Специальные требования к программной документации	12
5. СРЕДСТВА И ПОРЯДОК ИСПЫТАНИЙ	13
5.1. Технические средства	13

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.06.05-01 51 01-1				
Инв. № подл.	Подп. и дата	Взам. Инв. №	Инв. № дубл.	Подп. и дата

5.2.	Программные средства.....	13
5.3.	Порядок проведения испытаний.....	13
5.4.	Требования к персоналу	13
6.	МЕТОДЫ ИСПЫТАНИЙ.....	14
6.1.	Подготовка к проведению испытаний.....	14
6.2.	Проверка требований к программной документации.....	14
6.3.	Проверка требований к интерфейсу.....	14
6.4.	Проверка требований к функциональным характеристикам	14
6.4.1.	Проверка запуска, регистрации и входа	14
6.4.2.	Проверка работы с задачами.....	15
6.4.3.	Проверка подзадач и ИИ-разбиения	15
6.4.4.	Проверка начисления наград	15
6.4.5.	Проверка города	15
6.4.6.	Проверка магазина	15
6.4.7.	Проверка достижений, календаря и профиля.....	16
6.4.8.	Проверка API и контейнерного запуска	16
6.5.	Основные тестовые сценарии.....	16
6.5.1.	Проверка регистрации и авторизации пользователя	16
6.5.2.	Проверка создания задачи и отображения списка задач.....	18
6.5.3.	Проверка поиска и фильтрации задач	19
6.5.4.	Проверка валидации формы создания задачи	20
6.5.5.	Проверка разбиения задачи на подзадачи	20
6.5.6.	Проверка завершения задачи и обновления прогресса	21
6.5.7.	Проверка изометрического города и карточек зданий.....	22
6.5.8.	Проверка покупки объекта магазина	23
6.5.9.	Проверка отказа покупки при недостатке Coins	24
6.5.10.	Проверка календаря, достижений и профиля.....	25
6.5.11.	Проверка поведения при недоступности серверной части	26
6.6.	Основные тестовые сценарии.....	27
7.	ОФОРМЛЕНИЕ РЕЗУЛЬТАТОВ ИСПЫТАНИЙ.....	29
	СПИСОК ИСПОЛЬЗУЕМОЙ ЛИТЕРАТУРЫ	30

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.06.05-01 51 01-1				
Инв. № подл.	Подп. и дата	Взам. Инв. №	Инв. № дубл.	Подп. и дата

1. ОБЪЕКТ ИСПЫТАНИЙ

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».

Краткое наименование программы - «Taskoria».

1.2. Краткая характеристика области применения

«Taskoria» - Android-приложение для управления личными задачами и отслеживания прогресса пользователя в игровом формате. Программа предназначена для индивидуального планирования учебных, рабочих, личных задач и задач, связанных со здоровьем.

Особенностью приложения является визуализация прогресса через изометрический город. Выполнение задач приносит пользователю XP и Coins, влияет на статистику, достижения и развитие зданий, связанных с категориями задач.

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.06.05-01 51 01-1				
Инв. № подл.	Подп. и дата	Взам. Инв. №	Инв. № дубл.	Подп. и дата

2. ЦЕЛЬ ИСПЫТАНИЙ

Целью испытаний является проверка корректности выполнения программных функций, изложенных в разделе «Требования к программе» документа «Техническое задание» из комплекта документации в соответствии с ЕСПД.

Испытания должны подтвердить, что клиентское приложение корректно взаимодействует с серверной частью, основные пользовательские сценарии выполняются без критических ошибок, а данные задач, профиля, города, магазина и достижений обрабатываются согласно требованиям.

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.06.05-01 51 01-1				
Инв. № подл.	Подп. и дата	Взам. Инв. №	Инв. № дубл.	Подп. и дата

3. ТРЕБОВАНИЯ К ПРОГРАММЕ

3.1. Требования к функциональным характеристикам

3.1.1. Состав выполняемых функций

3.1.1.1. Регистрация и авторизация пользователя

Программа должна обеспечивать регистрацию пользователя с указанием имени, email и пароля. После успешной регистрации пользователь должен получать доступ к основным разделам приложения.

Программа должна обеспечивать вход пользователя по email и паролю. При успешном входе открывается главный экран города. При выходе из аккаунта пользовательская сессия должна завершаться, а приложение должно возвращаться к экрану входа.

3.1.1.2. Управление задачами

Приложение должно предоставлять функционал создания, просмотра, редактирования, удаления и завершения задач. Параметры задачи включают название, описание, категорию, приоритет, сложность и дедлайн. Название задачи является обязательным полем и не должно быть пустым.

Список задач должен отображать активные и завершенные задачи. Для удобства работы должны быть доступны поиск и фильтрация. При выборе задачи должна открываться карточка с полной информацией о задаче, наградами XP и Coins, а также списком подзадач.

3.1.1.3. Работа с подзадачами

Приложение должно обеспечивать отображение подзадач внутри карточки задачи. Подзадачи используются для разбиения крупной задачи на более понятные шаги. Пользователь должен иметь возможность просматривать подзадачи и изменять их состояние (активная/выпавшая подзадача).

Серверная часть должна поддерживать разбиение задачи на подзадачи с использованием ИИ-подсистемы. Полученные подзадачи должны сохраняться и отображаться в клиентском приложении.

3.1.1.4. Завершение задач и начисление наград

При завершении задачи программа должна начислять пользователю XP и Coins. Завершенная задача должна получать соответствующий статус и отображаться среди выполненных задач.

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.06.05-01 51 01-1				
Инв. № подл.	Подп. и дата	Взам. Инв. №	Инв. № дубл.	Подп. и дата

RU.17701729.06.05-01 51 01-1

После завершения задачи должны обновляться профиль пользователя, статистика, достижения и состояние города.

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.06.05-01 51 01-1				
Инв. № подл.	Подп. и дата	Взам. Инв. №	Инв. № дубл.	Подп. и дата

3.1.1.5. Изометрический город

Приложение должно отображать изометрическую карту города. На карте должны отображаться основные здания, связанные с категориями задач: школа, офис, госпиталь, личное здание и ратуша.

Уровни зданий должны изменяться в зависимости от прогресса пользователя по соответствующим категориям задач. Карточки зданий должны открываться при нажатии и содержать краткую информацию о здании и его уровне.

3.1.1.6. Магазин

Приложение должно содержать магазин декоративных объектов. Пользователь должен иметь возможность купить объект при достаточном количестве Coins. После покупки объект должен отображаться на карте города.

Если Coins недостаточно, покупка не должна выполняться, а баланс пользователя должен оставаться без изменений.

3.1.1.7. Достижения, календарь и профиль

Приложение должно отображать список достижений и статус их получения. Количество открытых достижений должно отображаться в профиле пользователя.

Календарь активности должен отображать задачи по датам. Профиль должен содержать основные показатели пользователя: уровень, XP, Coins, streak, количество выполненных задач и количество открытых достижений.

3.1.2. Организация входных данных

Все требования ко входным данным подробно описаны в функциональных требованиях. Данные, введенные пользователем в клиентском приложении, должны передаваться серверной части через разработанное API в теле HTTP-запроса.

Обязательные поля должны проверяться до отправки данных или на стороне серверной части. Некорректные данные не должны приводить к аварийному завершению программы.

3.1.3. Организация выходных данных

Все требования к выходным данным подробно описаны в функциональных требованиях. Данные, полученные от серверной части через API, должны отображаться в пользовательском интерфейсе приложения.

К выходным данным относятся сведения о пользователе, список задач и подзадач, состояние города, список достижений, календарь активности, товары магазина и результаты выполнения действий пользователя.

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.06.05-01 51 01-1				
Инв. № подл.	Подп. и дата	Взам. Инв. №	Инв. № дубл.	Подп. и дата

3.2. Требования к временным характеристикам

Приложение должно обеспечивать отклик на основные пользовательские действия в течение не более 2 секунд при штатной работе клиентского приложения и локально запущенной серверной части. Операции, связанные с обращением к ИИ-подсистеме, могут выполняться дольше из-за обращения к внешнему сервису.

3.3. Требования к интерфейсу

Интерфейс приложения должен включать основные разделы, доступные через нижнюю панель навигации: город, задачи, календарь, магазин и профиль. Дополнительно из основных экранов должны быть доступны достижения, уведомления и настройки.

Главный экран должен отображать изометрический город, баланс Coins, уровень пользователя и кнопку перехода к уведомлениям. Раздел задач должен предоставлять список задач, поиск, фильтрацию, форму создания задачи и карточку подробного просмотра. Раздел календаря должен отображать задачи по датам. Раздел магазина должен отображать товары и состояние покупки. Раздел профиля должен отображать показатели пользователя и статистику.

Интерфейс должен корректно отображаться на Android-устройствах с разными размерами экрана. Текстовые элементы, карточки и кнопки не должны перекрывать друг друга и препятствовать выполнению основных действий.

3.4. Требования к надежности

Приложение не должно аварийно завершаться при вводе некорректных данных пользователем. Ошибки загрузки данных, авторизации, создания задач и покупки товаров должны обрабатываться с переходом приложения в безопасное состояние.

Серверная часть должна проверять корректность входных данных и принадлежность данных текущему пользователю. Критические операции, такие как завершение задачи и покупка товара, не должны приводить к повторному начислению наград или некорректному списанию Coins.

3.5. Условия эксплуатации

Специального обслуживания для использования мобильного приложения не требуется. Пользователю достаточно владеть базовыми навыками работы с Android-устройством.

Для полной проверки программы необходимо запустить серверную часть локально или через Docker Compose. Для демонстрации интерфейса допускается запуск клиентского приложения в mock-режиме.

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.06.05-01 51 01-1				
Инв. № подл.	Подп. и дата	Взам. Инв. №	Инв. № дубл.	Подп. и дата

3.6. Требования к составу и параметрам технических средств

Для работы клиентского приложения требуется:

1. Мобильное устройство или эмулятор Android версии 8.0 или выше.
2. Не менее 200 МБ свободной памяти на устройстве.
3. Подключение к локальной сети или интернету при работе с серверной частью.

Для работы серверной части требуется персональный компьютер с возможностью запуска Python-приложений и PostgreSQL либо Docker Desktop.

3.7. Требования к информационной и программной совместимости

Клиентское приложение должно быть реализовано на языке Dart с использованием Flutter. Серверная часть должна быть реализована на Python с использованием FastAPI. В качестве базы данных должна использоваться PostgreSQL. Обмен данными между клиентской и серверной частями должен выполняться через HTTP API в формате JSON.

3.8. Требования к транспортировке и упаковке

Программный продукт передается в виде исходного кода, программной документации и ссылки на репозиторий GitHub. Серверная часть передается в виде исходного кода и конфигурации Docker Compose для локального запуска.

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.06.05-01 51 01-1				
Инв. № подл.	Подп. и дата	Взам. Инв. №	Инв. № дубл.	Подп. и дата

4. ТРЕБОВАНИЯ К ПРОГРАММНОЙ ДОКУМЕНТАЦИИ

4.1. Состав программной документации

На испытание должна быть представлена следующая программная документация:

1. «Геймифицированное мобильное приложение для управления задачами и визуализации прогресса - Taskoria». Техническое задание.
2. «Геймифицированное мобильное приложение для управления задачами и визуализации прогресса - Taskoria». Пояснительная записка.
3. «Геймифицированное мобильное приложение для управления задачами и визуализации прогресса - Taskoria». Программа и методика испытаний.
4. «Геймифицированное мобильное приложение для управления задачами и визуализации прогресса - Taskoria». Текст программы.
5. «Геймифицированное мобильное приложение для управления задачами и визуализации прогресса - Taskoria». Руководство оператора.
6. Ссылка на репозиторий с исходным кодом.

4.2. Специальные требования к программной документации

Программная документация должна быть выполнена в соответствии с требованиями ЕСПД и ГОСТами к каждому виду документа. Документация и программа сдаются в электронном виде.

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.06.05-01 51 01-1				
Инв. № подл.	Подп. и дата	Взам. Инв. №	Инв. № дубл.	Подп. и дата

5. СРЕДСТВА И ПОРЯДОК ИСПЫТАНИЙ

5.1. Технические средства

Для проведения испытаний требуется персональный компьютер для запуска серверной части и Android-устройство или эмулятор для запуска клиентского приложения.

5.2. Программные средства

Во время испытаний используются:

1. Python 3.11.
2. PostgreSQL 15 или Docker Compose с контейнером PostgreSQL.
3. Flutter SDK и Dart.
4. Android Studio или другое средство запуска Flutter-приложения.
5. Pytest для проверки серверной части.
6. Flutter analyze и Flutter test для проверки клиентской части.
7. Swagger UI для проверки доступности API.

5.3. Порядок проведения испытаний

Испытания проводятся совместно для клиентской и серверной частей, так как основная функциональность приложения связана с обменом данными через API.

Порядок проведения испытаний:

1. Проверить наличие программной документации.
2. Клонировать репозиторий с исходным кодом.
3. Установить зависимости серверной и клиентской частей.
4. Запустить серверную часть локально или через Docker Compose.
5. Проверить доступность Swagger UI.
6. Запустить автоматические тесты backend.
7. Запустить Flutter analyze и Flutter test.
8. Запустить Android-приложение в core_api-режиме.
9. Выполнить сценарии, описанные в разделе «Методы испытаний».
10. Зафиксировать фактические результаты испытаний.

5.4. Требования к персоналу

Для проведения базовых испытаний достаточно одного тестировщика с навыками запуска Android-приложений и локальной серверной части. Испытания проводятся в соответствии с настоящим документом и руководством оператора.

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.06.05-01 51 01-1				
Инв. № подл.	Подп. и дата	Взам. Инв. №	Инв. № дубл.	Подп. и дата

6. МЕТОДЫ ИСПЫТАНИЙ

6.1. Подготовка к проведению испытаний

Испытания направлены на установление следующих показателей:

- соответствие функционала приложения требованиям технического задания;
- корректность взаимодействия клиентской и серверной частей;
- корректность отображения и обработки пользовательских данных;
- корректность работы игровых механик: XP, Coins, достижений, магазина и города;
- устойчивость приложения при ошибках ввода и ошибках соединения.

6.2. Проверка требований к программной документации

Проверка осуществляется на соответствие документации требованиям ГОСТ и ее полноту. Проверяется наличие технического задания, пояснительной записки, руководства оператора, программы и методики испытаний, текста программы и ссылки на репозиторий.

6.3. Проверка требований к интерфейсу

Проверка интерфейса проводится сравнением фактических экранов приложения с описанием в руководстве оператора. Проверяется наличие следующих разделов:

- экран входа и регистрации;
- главный экран города;
- экран задач;
- форма создания задачи;
- карточка задачи;
- календарь активности;
- магазин;
- достижения;
- профиль;
- уведомления;
- настройки.

Также проверяется корректность переходов между разделами, работа нижней панели навигации и адаптация интерфейса под разные размеры Android-устройств.

6.4. Проверка требований к функциональным характеристикам

6.4.1. Проверка запуска, регистрации и входа

При открытии приложения пользователь должен попасть на экран входа или основной экран в зависимости от состояния сессии. При отсутствии аккаунта пользователь должен иметь возможность перейти на экран регистрации, заполнить необходимые поля и создать аккаунт.

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.06.05-01 51 01-1				
Инв. № подл.	Подп. и дата	Взам. Инв. №	Инв. № дубл.	Подп. и дата

После успешного входа приложение должно открыть главный экран города. При неверных данных входа или недоступности серверной части приложение должно отобразить ошибку и не завершаться аварийно.

6.4.2. Проверка работы с задачами

На экране задач должен отображаться список задач пользователя. Пользователь должен иметь возможность создать задачу, открыть карточку задачи, отредактировать данные, удалить задачу или завершить ее.

Валидация формы не должна позволять создать задачу без обязательного названия. После сохранения новая задача должна появляться в списке. После завершения задача должна получать статус выполненной и отображаться среди завершенных задач.

6.4.3. Проверка подзадач и ИИ-разбиения

При выполнении разбиения задачи на подзадачи приложение должно получить список подзадач от серверной части. Подзадачи должны отображаться в карточке задачи в корректном порядке.

При повторном разбиении задачи с уже существующими подзадачами программа должна обрабатывать ситуацию согласно логике приложения и не создавать неконтролируемое дублирование.

6.4.4. Проверка начисления наград

При завершении задачи пользователю должны начисляться XP и Coins. Значения в профиле должны обновляться после выполнения действия. Повторное завершение уже выполненной задачи не должно повторно начислять награды.

6.4.5. Проверка города

На главном экране должен отображаться изометрический город. При нажатии на здание должна открываться карточка здания. При выполнении задач определенной категории должен обновляться уровень соответствующего здания.

Ратуша должна отображать краткую сводку прогресса пользователя: уровень, количество выполненных задач и уровни зданий.

6.4.6. Проверка магазина

В магазине должны отображаться доступные декоративные объекты. При достаточном количестве Coins пользователь должен иметь возможность купить объект. После покупки баланс должен уменьшиться, а объект должен отображаться на карте города.

При недостаточном количестве Coins покупка не должна выполняться.

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.06.05-01 51 01-1				
Инв. № подл.	Подп. и дата	Взам. Инв. №	Инв. № дубл.	Подп. и дата

6.4.7. Проверка достижений, календаря и профиля

Экран достижений должен отображать список достижений и статус их получения. При выполнении условий достижения оно должно отображаться как открытое. Количество открытых достижений должно совпадать с показателем в профиле.

Календарь должен отображать задачи по датам.

Профиль должен отображать уровень, XP, Coins, streak, количество выполненных задач и количество открытых достижений.

6.4.8. Проверка API и контейнерного запуска

При запущенной серверной части должна быть доступна Swagger-документация API. При запуске через Docker Compose должны запускаться backend и PostgreSQL, а backend должен проходить проверку доступности.

6.5. Основные тестовые сценарии

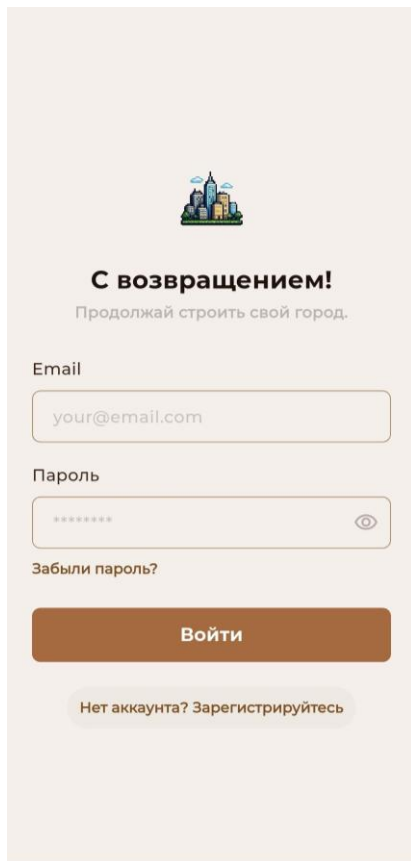
6.5.1. Проверка регистрации и авторизации пользователя

Порядок проведения проверки:

1. Открыть приложение.
2. Перейти на экран регистрации.
3. Ввести имя пользователя, email и пароль.
4. Отправить форму регистрации.
5. Проверить переход к основному интерфейсу приложения.
6. Выполнить выход из аккаунта через настройки.
7. Ввести данные созданного пользователя на экране входа.
8. Проверить открытие главного экрана города после входа.

Ожидаемый результат: пользователь может зарегистрироваться, войти в аккаунт и выйти из него. После повторного входа открывается основной интерфейс приложения.

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.06.05-01 51 01-1				
Инв. № подл.	Подп. и дата	Взам. Инв. №	Инв. № дубл.	Подп. и дата



С возвращением!
Продолжай строить свой город.

Email
your@email.com

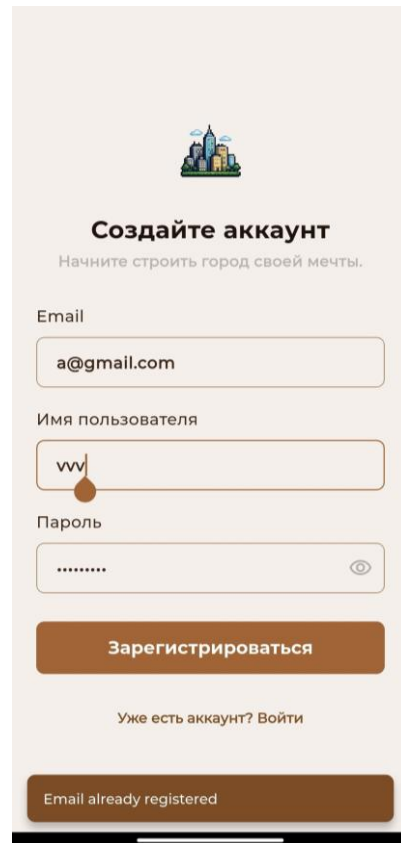
Пароль

Забыли пароль?

Войти

Нет аккаунта? Зарегистрируйтесь

Рис. 1. Экран входа в приложение



Создайте аккаунт
Начните строить город своей мечты.

Email
a@gmail.com

Имя пользователя
vvv

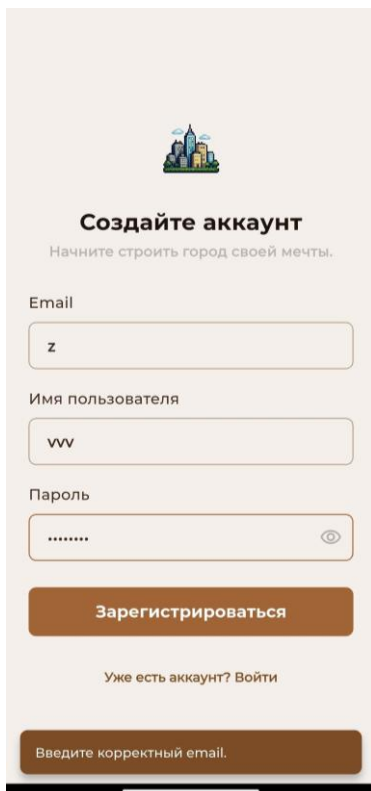
Пароль

Зарегистрироваться

Уже есть аккаунт? Войти

Email already registered

Рис. 2. Сообщение при использовании уже зарегистрированного email



Создайте аккаунт
Начните строить город своей мечты.

Email
z

Имя пользователя
vvv

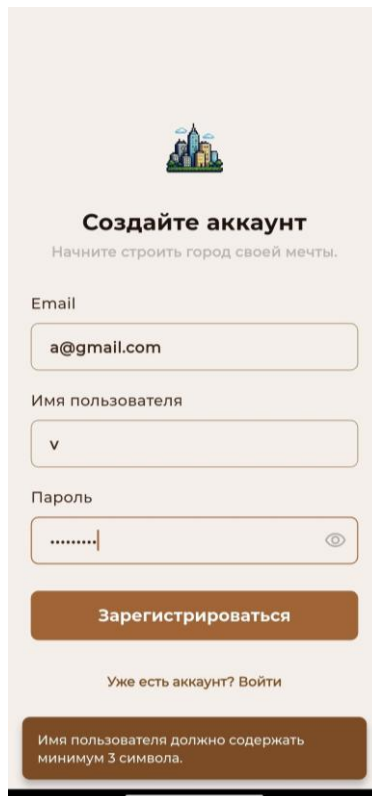
Пароль

Зарегистрироваться

Уже есть аккаунт? Войти

Введите корректный email.

Рис. 3. Неверный email



Создайте аккаунт
Начните строить город своей мечты.

Email
a@gmail.com

Имя пользователя
v

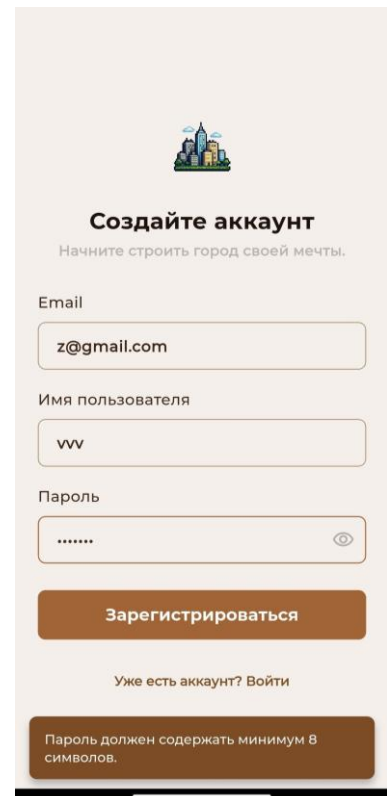
Пароль

Зарегистрироваться

Уже есть аккаунт? Войти

Имя пользователя должно содержать минимум 3 символа.

Рис. 4. Некорректный логин



Создайте аккаунт
Начните строить город своей мечты.

Email
z@gmail.com

Имя пользователя
vvv

Пароль

Зарегистрироваться

Уже есть аккаунт? Войти

Пароль должен содержать минимум 8 символов.

Рис. 5. Некорректный пароль

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.06.05-01 51 01-1				
Инв. № подл.	Подп. и дата	Взам. Инв. №	Инв. № дубл.	Подп. и дата

6.5.2. Проверка создания задачи и отображения списка задач

Исходные условия: пользователь авторизован, открыт основной интерфейс приложения.

Порядок проведения проверки:

1. Перейти в раздел задач.
2. Открыть форму создания задачи.
3. Заполнить название задачи, описание, категорию, приоритет, сложность и дедлайн.
4. Сохранить задачу.
5. Вернуться к списку задач.
6. Проверить наличие созданной задачи в списке активных задач.
7. Открыть карточку созданной задачи.
8. Сравнить данные в карточке задачи с данными, введенными при создании.

Ожидаемый результат: задача создается, сохраняется и отображается в списке. Карточка задачи содержит название, описание, категорию, приоритет, сложность, дедлайн и рассчитанные награды XP и Coins.

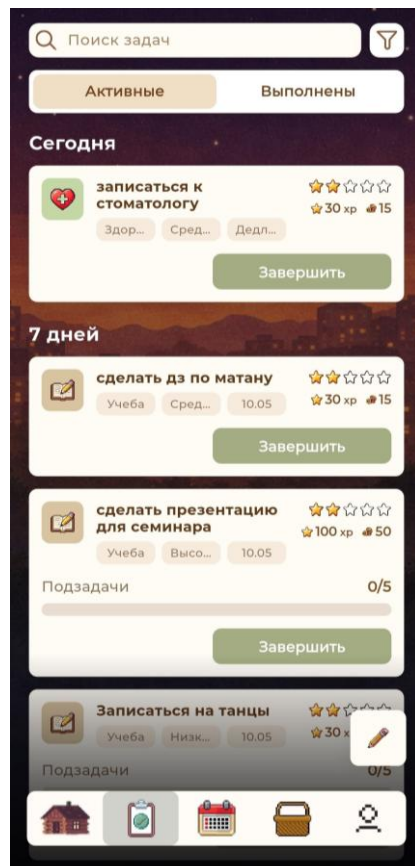


Рис. 6. Форма создания задачи с заполненными полями

Рис. 7. Список задач после создания задач

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.06.05-01 51 01-1				
Инв. № подл.	Подп. и дата	Взам. Инв. №	Инв. № дубл.	Подп. и дата

6.5.3. Проверка поиска и фильтрации задач

Исходные условия: пользователь авторизован, в приложении есть несколько задач с разными категориями, статусами, приоритетами и сложностью.

Порядок проведения проверки:

1. Перейти в раздел задач.
2. Ввести часть названия задачи в строку поиска.
3. Проверить, что в списке остаются задачи, соответствующие поисковому запросу.
4. Открыть панель фильтрации.
5. Выбрать категорию, приоритет, сложность или режим отображения завершенных задач.
6. Применить фильтр и проверить содержимое списка.
7. Сбросить фильтры и убедиться, что список задач возвращается к исходному состоянию.

Ожидаемый результат: поиск и фильтрация изменяют список задач в соответствии с выбранными параметрами. После сброса фильтров отображается полный список задач, доступных пользователю.

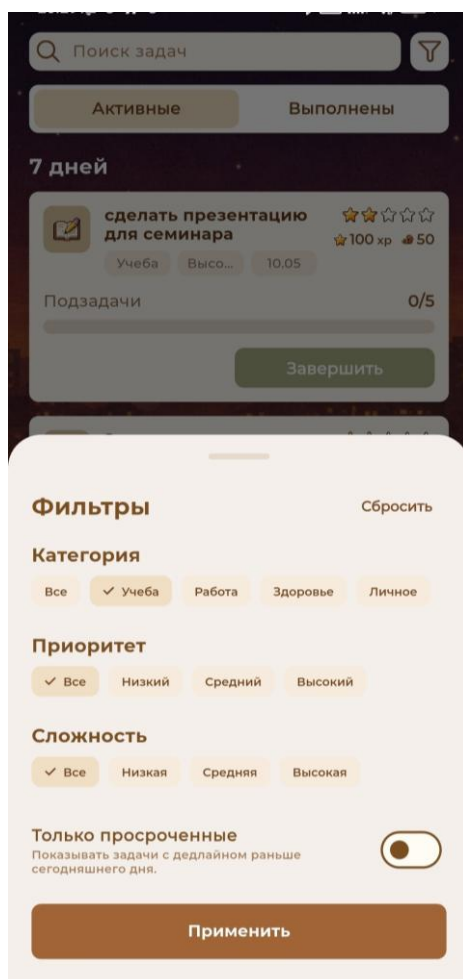


Рис. 8. Панель выбора параметров фильтрации

Рис. 9. Список задач после применения фильтров

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.06.05-01 51 01-1				
Инв. № подл.	Подп. и дата	Взам. Инв. №	Инв. № дубл.	Подп. и дата

6.5.4. Проверка валидации формы создания задачи

Исходные условия: пользователь находится на экране создания задачи. Порядок проведения проверки:

1. Оставить обязательное поле названия задачи пустым.
2. Заполнить или не заполнять остальные поля формы.
3. Нажать кнопку создания задачи.
4. Проверить реакцию приложения.

Ожидаемый результат: кнопка создания заблокирована, задача не создается. Пользователь остается на форме создания задачи, приложение не завершается аварийно.

6.5.5. Проверка разбиения задачи на подзадачи

Исходные условия: создана задача с названием и описанием, пользователь находится в карточке задачи.

Порядок проведения проверки:

1. Открыть карточку задачи.
2. Запустить действие разбиения задачи на подзадачи.
3. Дождаться ответа серверной части.
4. Проверить список подзадач в карточке задачи.
5. Повторно открыть задачу и проверить сохранение подзадач.

Ожидаемый результат: в карточке задачи отображается список подзадач. Подзадачи имеют названия и описания, связаны с исходной задачей и сохраняются после повторного открытия карточки.

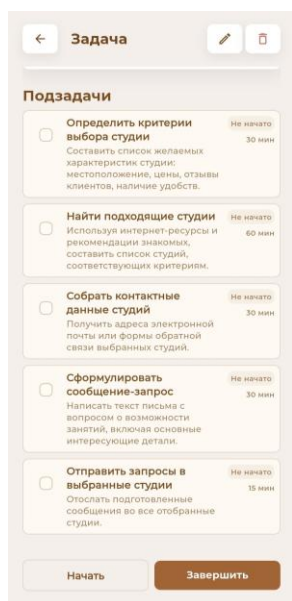


Рис. 10. Карточка задачи со списком подзадач

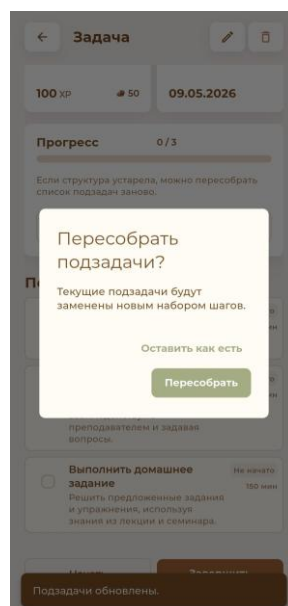


Рис. 11. Проверка повторного формирования подзадач

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.06.05-01 51 01-1				
Инв. № подл.	Подп. и дата	Взам. Инв. №	Инв. № дубл.	Подп. и дата

6.5.6. Проверка завершения задачи и обновления прогресса

Исходные условия: создана активная задача, пользователь находится в карточке задачи или в списке задач.

Порядок проведения проверки:

1. Зафиксировать текущие значения XP, Coins, уровня пользователя и количества выполненных задач в профиле.
2. Открыть активную задачу.
3. Завершить задачу.
4. Вернуться в список задач и проверить изменение статуса.
5. Открыть профиль и проверить изменение XP, Coins и количества выполненных задач.
6. Открыть экран достижений и проверить наличие новых открытых достижений.
7. Открыть главный экран города и проверить изменение соответствующего здания.

Ожидаемый результат: задача получает статус выполненной, награды начисляются один раз, профиль и город обновляются. При выполнении условий достижения оно отображается как открытое.



Рис. 12. Список выполненных задач



Рис. 13. Профиль пользователя после выполнения задач

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.06.05-01 51 01-1				
Инв. № подл.	Подп. и дата	Взам. Инв. №	Инв. № дубл.	Подп. и дата



Рис. 14. Состояние здания до выполнения задачи «личное»



Рис. 15. Состояние здания после выполнения задачи «личное»

6.5.7. Проверка изометрического города и карточек зданий

Исходные условия: пользователь авторизован, состояние города загружено с серверной части или из mock-данных.

Порядок проведения проверки:

1. Открыть главный экран города.
2. Проверить отображение основных зданий: школы, офиса, госпиталя, личного здания и ратуши.
3. Выполнить прокрутку карты.
4. Проверить визуальное соответствие зданий и дорог изометрической сетке.
5. Нажать на каждое основное здание.
6. Проверить открытие карточки здания с названием, описанием и уровнем.
7. Нажать на ратушу и проверить отображение сводной информации о профиле и прогрессе.

Ожидаемый результат: карта отображается корректно, здания не препятствуют

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.06.05-01 51 01-1				
Инв. № подл.	Подп. и дата	Взам. Инв. №	Инв. № дубл.	Подп. и дата

использованию интерфейса, карточки зданий открываются по нажатию, нижняя навигация не перекрывает содержимое карточек.



Рис. 16. Карточка ратуши со сводной информацией

6.5.8. Проверка покупки объекта магазина

Исходные условия: пользователь авторизован, баланс Coins достаточен для покупки хотя бы одного объекта магазина.

Порядок проведения проверки:

1. Открыть магазин.
2. Выбрать доступный декоративный объект.
3. Выполнить покупку.
4. Проверить изменение баланса Coins.
5. Вернуться на главный экран города.

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.06.05-01 51 01-1				
Инв. № подл.	Подп. и дата	Взам. Инв. №	Инв. № дубл.	Подп. и дата

6. Проверить отображение купленного объекта на карте.

Ожидаемый результат: объект покупается, баланс уменьшается на стоимость товара, декоративный объект отображается в городе.

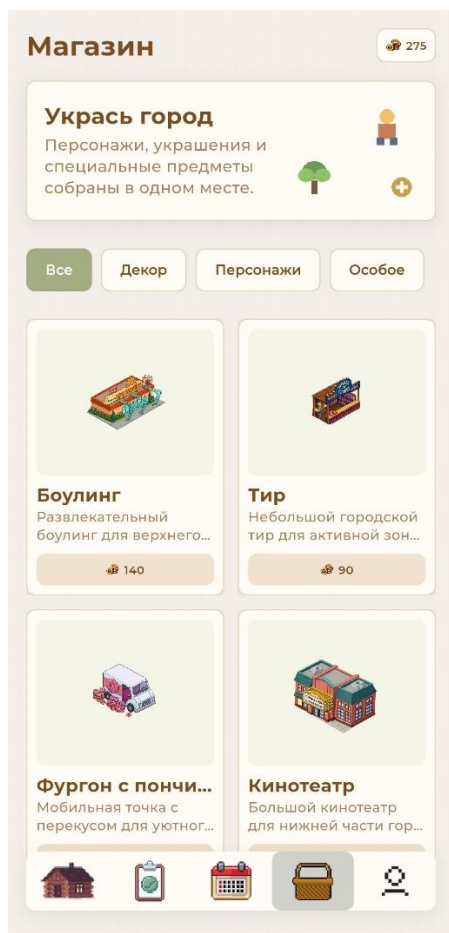


Рис. 17. Состояние магазина до покупки



Рис. 18. Отображение купленных декоративных объектов на карте

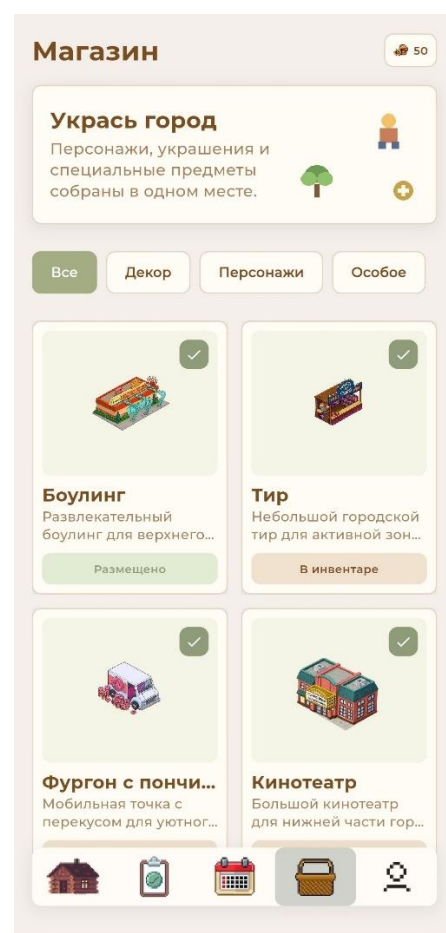


Рис. 19. Состояние магазина после покупки

6.5.9. Проверка отказа покупки при недостатке Coins

Исходные условия: баланс пользователя меньше стоимости выбранного товара.

Порядок проведения проверки:

1. Открыть магазин.
2. Выбрать товар, стоимость которого больше текущего баланса.
3. Попытаться выполнить покупку.
4. Проверить баланс пользователя и состояние товара.

Ожидаемый результат: покупка не выполняется, баланс Coins не изменяется, товар не появляется на карте города.

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.06.05-01 51 01-1				
Инв. № подл.	Подп. и дата	Взам. Инв. №	Инв. № дубл.	Подп. и дата



Рис. 20. Отказ покупки при недостаточном количестве Coins

6.5.10. Проверка календаря, достижений и профиля

Исходные условия: у пользователя есть созданные или завершенные задачи.

Порядок проведения проверки:

- 1. Открыть календарь активности.
- 2. Проверить отображение задач по датам.
- 3. Открыть экран достижений.
- 4. Проверить список достижений и статус открытых достижений.
- 5. Открыть профиль.
- 6. Сравнить количество открытых достижений в профиле с экраном достижений.
- 7. Проверить отображение уровня, XP, Coins, streak и количества выполненных задач.

Ожидаемый результат: календарь, достижения и профиль отображают согласованные данные пользователя.

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.06.05-01 51 01-1				
Инв. № подл.	Подп. и дата	Взам. Инв. №	Инв. № дубл.	Подп. и дата



Рис. 21. Календарь активности с задачами по датам

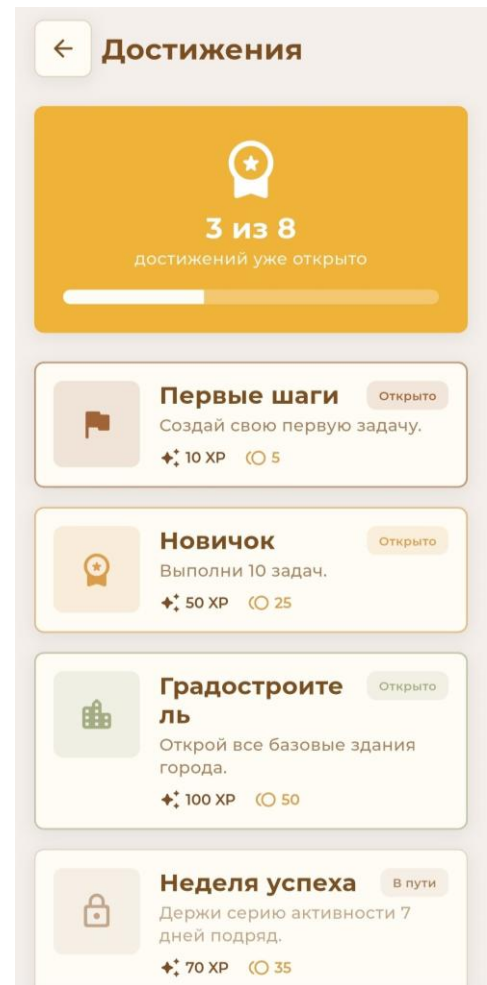


Рис. 22. Экран достижений с открытыми достижениями

6.5.11. Проверка поведения при недоступности серверной части

Порядок проведения проверки:

1. Остановить backend или отключить доступ к локальному API.
2. Открыть раздел задач, профиль или магазин.
3. Выполнить действие, требующее обращения к серверной части.
4. Проверить реакцию интерфейса.
5. Запустить backend повторно и обновить экран.

Ожидаемый результат: приложение не завершается аварийно. При недоступности сервера отображается состояние ошибки.

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.06.05-01 51 01-1				
Инв. № подл.	Подп. и дата	Взам. Инв. №	Инв. № дубл.	Подп. и дата

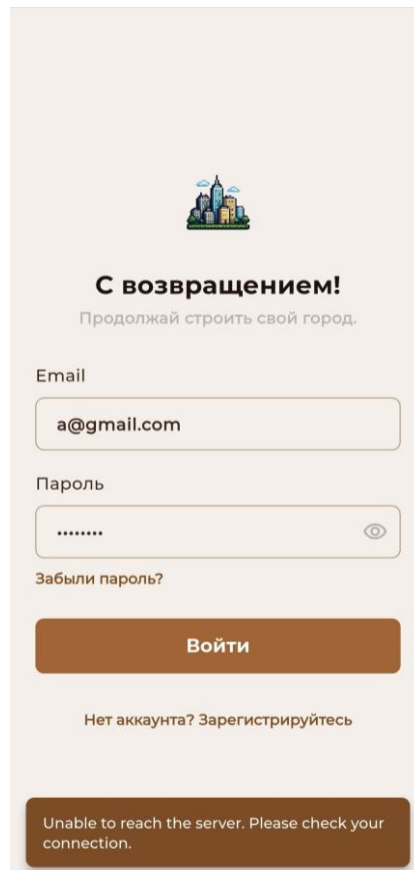


Рис. 23. Ошибка при недоступном бэкенде

6.6. Основные тестовые сценарии

Основные сценарии испытаний:

1. Регистрация нового пользователя.
2. Вход существующего пользователя.
3. Создание задачи с корректными данными.
4. Проверка запрета создания задачи без названия.
5. Просмотр списка задач и карточки задачи.
6. Поиск и фильтрация задач.
7. Разбиение задачи на подзадачи.
8. Завершение задачи и начисление XP и Coins.
9. Проверка запрета повторного начисления наград.
10. Проверка обновления города после выполнения задач.
11. Покупка объекта магазина при достаточном балансе.
12. Проверка отказа покупки при недостаточном балансе.
13. Проверка отображения достижений.

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.06.05-01 51 01-1				
Инв. № подл.	Подп. и дата	Взам. Инв. №	Инв. № дубл.	Подп. и дата

- 14. Проверка календаря активности.
- 15. Проверка профиля пользователя.
- 16. Проверка Swagger UI.
- 17. Проверка запуска через Docker Compose.

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.06.05-01 51 01-1				
Инв. № подл.	Подп. и дата	Взам. Инв. №	Инв. № дубл.	Подп. и дата

7. ОФОРМЛЕНИЕ РЕЗУЛЬТАТОВ ИСПЫТАНИЙ

По итогам испытаний фиксируется соответствие фактического поведения программы ожидаемому результату. При обнаружении ошибки указываются условия ее воспроизведения, ожидаемый результат и фактическое поведение программы.

Программа считается прошедшей испытания, если основные функции, указанные в настоящем документе, выполняются корректно, критические ошибки отсутствуют, а серверная и клиентская части успешно запускаются в предусмотренных режимах.

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.06.05-01 51 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 51 01-1				
Инв. № подл.	Подп. и дата	Взам. Инв. №	Инв. № дубл.	Подп. и дата

ЛИСТ РЕГИСТРАЦИИ ИЗМЕНЕНИЙ

[illegible]