|  |  |  |
| --- | --- | --- |
|  |  |  |
| МИНОБРНАУКИ РОССИИ | | |
| Федеральное государственное бюджетное образовательное учреждение  высшего образования  **«МИРЭА – Российский технологический университет»**  **РТУ МИРЭА** | | |

Институт Информационных технологий

Кафедра Математического обеспечения и стандартизации информационных технологий

**Отчет по практическим работам №1-4**

по дисциплине «Системная и программная инженерия»

|  |  |
| --- | --- |
| **Выполнили:**  Студенты группыИКБО-20-22 | Шумахер М.Е.  Муллаянов А.И.  Исаков Л.А.  Туганов Н.А.  Фридрих А.В. |
| **Проверил:**  Преподаватель | Михайлова Е.К. |

2025 г.

СОДЕРЖАНИЕ

[Практическая работа №1 3](#_Toc190806878)

[Практическая работа №2 4](#_Toc190806879)

[Практическая работа №3 6](#_Toc190806880)

[Практическая работа №4 8](#_Toc190806881)

**Практическая работа №1**

В нашей команде участвуют 5 человек: Шумахер Марк, Муллаянов Алим, Исаков Леонид, Туганов Никита, Фридрих Александр.

Руководителем группы и менеджером проекта выступает ХХХ.

Back-end разработчиком является Туганов Никита.

Front-end разработчиком и тестировщиком является Исаков Леонид.

Аналитиком является Фридрих Александр.

Шумахер Марк и Муллаянов Алим – технические писатели.

Тема проекта: веб-приложение «Магазин игрушек».

**Практическая работа №2**

**1 Назначение системы**

Назначение разрабатываемого веб-приложения "Магазин игрушек" заключается в цифровизации процесса продажи игрушек, предоставлении пользователям удобного и доступного инструмента для выбора и покупки товаров онлайн. Это позволяет автоматизировать процесс взаимодействия между продавцом и покупателем, упростить поиск и заказ товаров, а также обеспечить комфортный пользовательский опыт.

Основная цель приложения – создать платформу, которая объединяет широкий ассортимент игрушек, начиная от развивающих игр для детей и заканчивая коллекционными моделями для взрослых. Приложение решает задачи, связанные с автоматизацией бизнес-процессов, таких как управление каталогом товаров, обработка заказов, учет продаж и взаимодействие с клиентами.

**2 User story**

Ниже написаны необходимые для разработки пользовательские истории.

Таблица 1 – Перечень UserStory

|  |  |  |
| --- | --- | --- |
| Кто? | Что хочет? | С какой целью? |
| Пользователь | Зарегистрироваться в системе | Получить доступ к личному кабинету, сохранять данные и отслеживать заказы. |
| Покупатель | Формировать корзину товаров | Собрать все выбранные товары в одном месте для удобного оформления заказа. |
| Покупатель | Оплатить корзину товаров | Завершить покупку и получить подтверждение заказа. |
| Покупатель | Просматривать историю заказов | Отслеживать свои предыдущие покупки и статус заказов. |
| Покупатель | Добавлять товары в избранное | Сохранять понравившиеся товары для быстрого доступа в будущем. |

Продолжение таблицы 1

|  |  |  |
| --- | --- | --- |
| Покупатель | Фильтровать товары по категориям | Быстро находить нужные товары, сужая выбор по параметрам (цена, возраст, тип и т.д.). |
| Покупатель | Оценивать товары | Оставить отзыв и рейтинг, чтобы помочь другим покупателям с выбором. |
| Администратор веб-магазина | Просматривать аккаунты клиентов | Контролировать активность пользователей и анализировать данные для улучшения сервиса. |
| Администратор веб-магазина | Блокировать аккаунты клиентов | Пресекать нарушения правил использования платформы. |
| Администратор веб-магазина | Размещать новые товары | Обновлять ассортимент магазина, добавлять новые позиции для продажи. |

**3 Функциональные требования**

Ниже написаны функциальные требовая в соответствии с пользовательскими историями.

Таблица 2 – Перечень функциональных требований

|  |  |  |  |
| --- | --- | --- | --- |
| Кто? | Что хочет? | С какой целью? | Функциональное требование |
| Пользователь | Зарегистрироваться в системе | Получить доступ к личному кабинету, сохранять данные. | 1. Пользователь должен иметь возможность зарегистрироваться, указав email, пароль и дополнительные данные (имя, фамилия, телефон).  2. После регистрации пользователь должен получить доступ к личному кабинету. |

Продолжение таблицы 2

|  |  |  |  |
| --- | --- | --- | --- |
| Пользователь | Формировать корзину товаров | Собрать все выбранные товары в одном месте для удобного оформления заказа. | 1. Покупатель должен иметь возможность добавлять товары в корзину с указанием количества.  2. Корзина должна отображать общую стоимость товаров и возможность редактирования (удаление, изменение количества). |
| Пользователь | Оплатить корзину товаров | Завершить покупку и получить подтверждение заказа. | 1. Покупатель должен иметь возможность выбрать способ оплаты (карта, электронный кошелек).  2. После успешной оплаты покупатель может просмотреть заказ в личном кабинете. |
| Пользователь | Просматривать историю заказов | Отслеживать свои предыдущие покупки и статус заказов. | В личном кабинете покупатель должен видеть список всех заказов с деталями (дата, статус, состав заказа). |
| Пользователь | Добавлять товары в избранное | Сохранять понравившиеся товары для быстрого доступа в будущем. | Покупатель должен иметь возможность добавлять неограниченное количество товаров в избранное. |
| Пользователь | Фильтровать товары по категориям | Быстро находить нужные товары. | 1. Покупатель должен иметь возможность фильтровать товары по категориям. |

Продолжение таблицы 2

|  |  |  |  |
| --- | --- | --- | --- |
| Пользователь | Оценивать товары | Оставить оценку, чтобы помочь другим покупателям с выбором. | 1. Покупатель должен иметь возможность ставить оценку товару после покупки.  2. Оценки должны быть видны другим пользователям на странице товара. |
| Администратор веб-магазина | Просматривать аккаунты клиентов | Контролировать активность пользователей и анализировать данные для улучшения сервиса. | 1. Администратор должен иметь доступ к списку всех зарегистрированных пользователей с возможностью просмотра их данных и активности. |
| Администратор веб-магазина | Блокировать аккаунты клиентов | Пресекать нарушения правил использования платформы. | 1. Администратор должен иметь возможность блокировать аккаунты пользователей с указанием причины. |
| Администратор веб-магазина | Размещать новые товары | Обновлять ассортимент магазина, добавлять новые позиции для продажи. | 1. Администратор должен иметь возможность добавлять новые товары через администраторскую панель, указывая название, описание, цену, фото и категорию.  2. Товары должны автоматически появляться в каталоге после добавления. |

**4 Первоначальный план реализации проекта**

1. Определение целей и задач проекта

Цель: Создать клиент-серверное приложение для управления онлайн-магазином игрушек.

Основные задачи:

* Реализация функционала для клиентов (просмотр товаров, добавление в корзину, оформление заказов),
* Реализация функционала для администраторов (управление товарами, заказами, пользователями),
* Обеспечение безопасности данных (авторизация, регистрация, защита персональных данных).

2. Анализ требований. Функциональные требования:

Клиентская часть:

* Просмотр каталога товаров,
* Фильтрация и поиск товаров,
* Добавление товаров в корзину,
* Оплата корзины онлайн,
* Избранное,
* Отзывы,
* Оформление заказа,
* Личный кабинет пользователя (история заказов, данные профиля).

Серверная часть:

* Управление товарами (добавление, редактирование, удаление),
* Управление заказами (просмотр, изменение статуса),
* Управление пользователями (регистрация, авторизация, роли).

Нефункциональные требования:

* Высокая производительность и отказоустойчивость,
* Безопасность данных (шифрование, защита от атак),
* Масштабируемость.

3. Выбор технологий.

Клиентская часть:

* Веб-интерфейс: HTML, CSS, JavaScript (React.js или Vue.js);
* Мобильное приложение (опционально): React Native или Flutter.

Серверная часть:

* Язык программирования: Python (Django/Flask), Node.js, Java (Spring), или C# (.NET);
* База данных: PostgreSQL, MySQL, или MongoDB;
* API: RESTful API или GraphQL.

Инфраструктура:

* Хостинг: AWS, Google Cloud, или Azure,
* Контейнеризация: Docker,
* Оркестрация: Kubernetes (опционально).

4. Проектирование архитектуры.

Клиент-серверная архитектура:

* Клиент (веб-приложение, мобильное приложение),
* Сервер (API, база данных, бизнес-логика).

Схема базы данных:

Таблицы: Пользователи, Товары, Заказы, Категории, Корзина и т.д.

API Endpoints:

* GET /products — получение списка товаров,
* POST /order — создание заказа,
* PUT /products/{id} — обновление товара,
* DELETE /products/{id} — удаление товара.

5. Разработка

Этапы разработки:

1. Создание базовой структуры проекта (клиент, сервер, база данных).
2. Разработка клиентского интерфейса (каталог, корзина, оформление заказа).
3. Интеграция авторизации и регистрации.
4. Добавление суперпользователя
5. Тестирование и отладка.
6. Оптимизация и масштабирование.

6. Тестирование:

* Модульное тестирование (Unit Testing),
* Интеграционное тестирование (Integration Testing),
* Нагрузочное тестирование (Load Testing),
* Тестирование безопасности (Security Testing).

7. Деплой и запуск:

Настройка серверов и развертывание приложения.

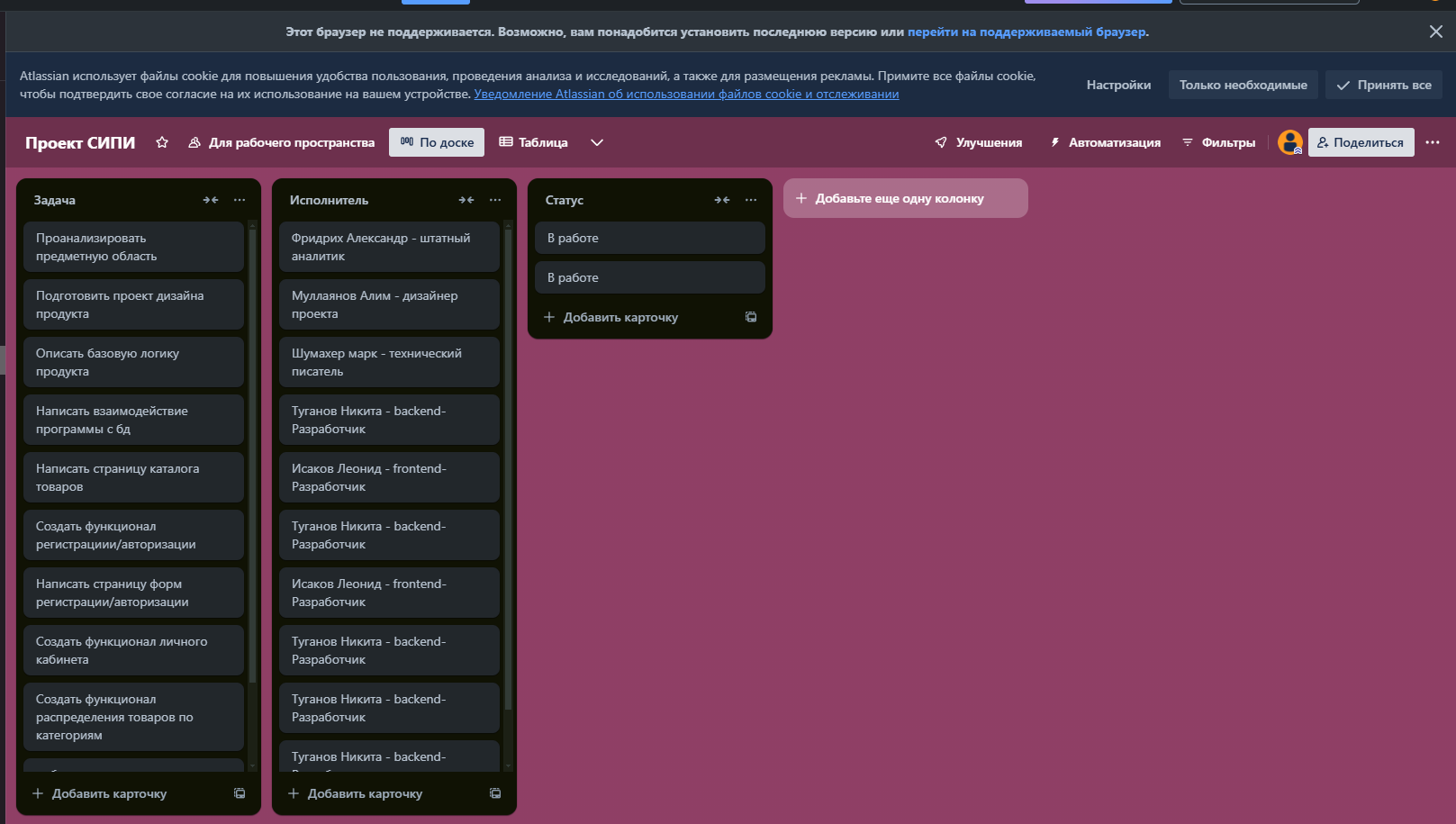


Рисунок 1 – первоначальный план реализации проекта

**Практическая работа №3**

**1 Портрет пользователя**

Имя: Анастасия, 32 года

Пол: Женский

Место проживания: Москва

Семейное положение: замужем, есть дети

Сфера занятости: Домохозяйка, занимается воспитанием детей

Уровень дохода: Ограниченный бюджет (основной доход семьи обеспечивает муж, Анастасия получает около 20 000 рублей в месяц на личные расходы и нужды детей)

Связанные с ним проблемы: ограниченный бюджет на покупку игрушек заставляет ее искать выгодные предложения и акции. У нее также не хватает времени на походы по магазинам из-за занятости с детьми и домашними делами. Она стремится выбрать качественные и безопасные игрушки, которые соответствуют возрасту и интересам ее детей, но часто испытывает трудности в поиске подходящих вариантов. Кроме того, дети часто просят конкретные игрушки, например, из популярных мультфильмов, что добавляет сложностей в процесс выбора.

Потребности и желания: хочет найти удобный и быстрый способ поиска игрушек, используя фильтры по категориям, такие как возраст, тип и цена. Она нуждается в возможности сравнить цены и выбрать оптимальный вариант, а также получить доступ к акциям и скидкам, чтобы сэкономить. Ей важно читать отзывы других родителей о товарах, чтобы убедиться в их качестве и безопасности. Она также желает упростить процесс оформления заказа и получить удобную доставку на дом. Сохранение понравившихся товаров в избранное для будущих покупок также является ее потребностью.

Фобии: боится приобрести некачественный товар, который может быть опасен для ее детей. Она также опасается, что заказ придет не вовремя, особенно если игрушка нужна к важному событию, например, к дню рождения ребенка.

**2 Use case диаграмма**

Ниже представлена Use case диаграмма.

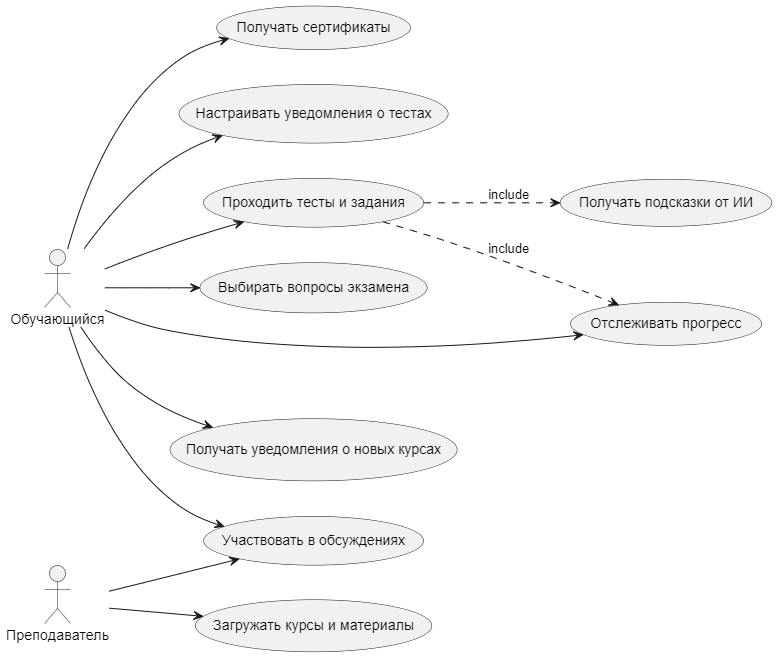


Рисунок 1 – Use case диаграмма системы

**3 Диаграмма последовательности**

Ниже представлена диаграмма последовательности взаимодействия пользователя с системой.

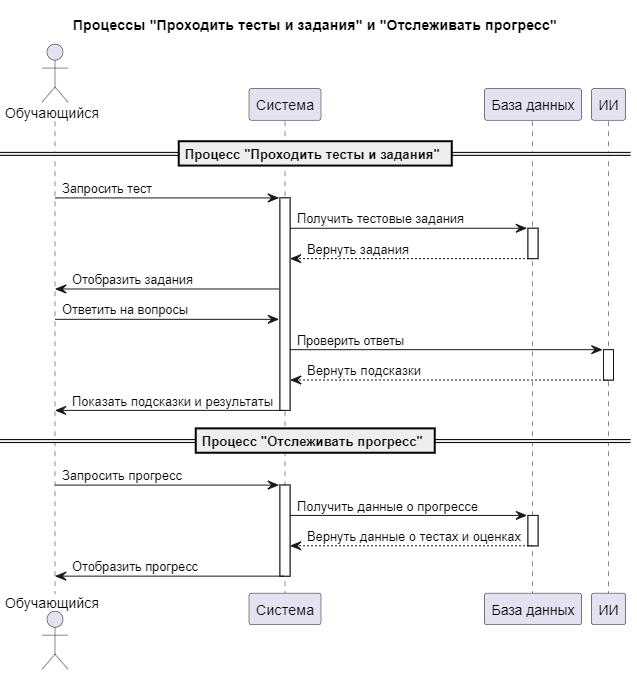


Рисунок 2 – Диаграмма последовательности

**Практическая работа №4**

**1 Нефункциональные требования**

Ниже представлена таблица с нефункциональными требованиями.

Таблица 3 – Перечень нефункциональных требований

|  |  |
| --- | --- |
| Тип требования | Содержание требования |
| Технические ограничения | Веб-приложение должно быть кроссбраузерным и одинаково функционировать в всех популярных браузерах: Google Chrome, Mozilla Firefox, Opera, Yandex.Браузер, Microsoft Edge, Safari. |
| Приложение должно быть адаптировано под использование на мобильных устройствах. |
| Отсутствие рекламы |
| Производительность | Загрузка главной страницы сайта должна занимать не более 3 секунд. |

Продолжение таблицы 3

|  |  |
| --- | --- |
|  | Открытие страницы товара должно происходить за 3 секунды (максимум). |
| Обработка добавления товара в корзину или избранное не должна превышать 1 секунды. |
| Масштабируемость | Система должна поддерживать одновременную работу минимум 1000 пользователей. |
| База данных должна быть оптимизирована для быстрого поиска и фильтрации товаров по категориям. |
| Возможность масштабирования системы для поддержки до 10 000 пользователей при увеличении нагрузки. |
| Надежность | Время безотказной работы (uptime) должно быть не менее 99,5% в год. |
| Приложение должно корректно восстанавливаться после сбоя при повторном запуске, сохраняя данные о корзине и избранном. |
| Регулярное резервное копирование данных для предотвращения потери информации. |
| Безопасность | Данные пользователей юдолжны передаваться по защищенному соединению (HTTPS, TLS 1.2+). |
| Использование протокола OAuth 2.0 для авторизации. |
| Пароли пользователей должны храниться в зашифрованном виде с использованием алгоритма bcrypt. |
| Локализация | Приложение должно поддерживать русский язык, при необходимости английский. |
| Формат даты и времени должен соответствовать региональным стандартам пользователей (например, DD.MM.YYYY для России). |
| Локализация текстовых элементов интерфейса, включая кнопки, меню и уведомления. |

**2 Матрица требований**

Ниже представлена таблица с матрицей требований.

Таблица 4 – Матрица требований

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **№** | **Требование** | **Суть** | **Автор** | **Ссылки** | **Критерий проверки** |
|  |  | | | | |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  | | | | |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  | | | | |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  | | | | |
|  |  |  |  |  |  |
|  |  |  |  |  |  |