

«УНИВЕРСИТЕТ ИТМО»

Кафедра вычислительной техники

КУРСОВАЯ РАБОТА

по дисциплине:

Программирование интернет-приложений

Выполнили:

Калугина Марина Максимовна

Каюков Иван Алексеевич

Группа Р3202

Санкт-Петербург

2018 г.

Содержание

1 О проекте	2
1.1 Цель создания	2
1.2 Целевая аудитория	2
1.3 Описание прецедентов	3
1.4 UML-диаграмма прецедентов	9
2 Требования к аппаратно-программному обеспечению	9
2.1 Требования к серверному обеспечению	9
2.2 Требования к клиентскому обеспечению	10
3 Требования к архитектуре системы	10
3.1 Уровень back-end	10
3.2 Уровень front-end	10
3.3 Telegram-бот	11
4 Требования к надежности и безопасности системы	11
5 Архитектура системы	11
6 Функциональные требования	12
6.1 Для неавторизированных пользователей	12
6.2 Для авторизированных пользователей	13
6.3 Для сотрудников	13
6.4 Для администраторов	13
7 Прототипы интерфейсов	14

1 О проекте

Предметом разработки является *RoboGit* – платформа для хостинга робототехнических проектов, который основан на системе контроля версий *Git* с поддержкой *markdown* и *RoboStore* – интернет-магазин с товарами для робототехники на домене *.onion*, поддерживающий оплату криптовалютой. Платформа позволяет пользователям формировать заказ на покупку необходимого для текущего проекта комплекта деталей и совершать покупку необходимого в один клик.

1.1 Цель создания

- Предоставление пользователям возможности размещения робототехнических проектов и организации пользовательских проектов с помощью системы контроля версий *Git*.
- Поддержание анонимности пользователей.
- Организация интернет-магазина с товарами для робототехники.
- Предоставление пользователям возможности упрощенно совершать покупку необходимых для проекта товаров.

1.2 Целевая аудитория

Плохие парни.

1.3 Описание прецедентов

Регистрация и авторизация

Цель: Завести аккаунт в системе, создать пользователь

Сценарий:

1. Незарегистрированный/неавторизированный пользователь имеет доступ ко всем страницам платформы, кроме своего профиля и создания нового репозитория.
2. При попытке перейти к себе в профиль (главную страницу *RoboGit*) или при попытке создания нового репозитория открывается форма авторизации с ссылкой на форму регистрации.
3. Пользователь автоавторизуется или переходит на страницу регистрации, заполняет поля и отправляет форму.
4. Система перенаправляет на страницу личного профиля пользователя.

Расширения:

- Ошибка при авторизации (неверный логин/пароль) или ошибка при регистрации (адрес электронной почты уже используется, пароль слишком короткий) предотвращает переход на страницу личного профиля.

Создание нового репозитория

Цель: Создать новый репозиторий

Сценарий:

1. Авторизованный пользователь переходит на страницу создания нового репозитория.
2. Открывается форма для ввода имени репозитория и его описания.
3. После успешной отправки формы появляется страница, в которой описывается как создать репозиторий с командной строки или импортировать код.

Расширения:

- Ошибка при создании репозитория (репозиторий уже с таким именем уже существует в данном аккаунте) предотвращает создание репозитория.

Работа с существующим репозиторием

Цель: Держать актуальные версии проектов в *RoboGit*.

Сценарий:

1. Работа с существующим репозиторием ведется через командную строку пользователя персонального компьютера.

Просмотр пользовательских репозиториев

Цель: Просмотр пользователями репозиториев других пользователей, знакомство с кодом и описаниями.

Сценарий:

1. Пользователь открывает страницу репозитория.
2. Платформа предоставляет обзор всех файлов, находящихся в репозитории.
3. Файлы, написанные при помощи языка разметки *Markdown* и имеющие расширение *.md*, представляются на платформе в виде страницы-публикации.

Расширения:

- В файлах с расширением *.md* можно указывать товары из *RoboStore* для добавления из корзины пользователей.

Оценка пользовательских репозиториев

Цель: Поощрение интересных проектов, создание топа лучших проектов *RoboGit*.

Сценарий:

1. Пользователь отмечает понравившиеся репозитории, нажимая кнопку *stars* на странице репозитория.
2. Система сохраняет информацию о количестве *stars* для текущего репозитория.

Расширения:

- Параметр *stars* влияет на положение репозитория в топе.
- В пользовательских профилях для каждого репозитория отображается количество *stars*.

Формирование заказа на покупку из пользовательских проектов

Цель: Добавление товара из проекта в корзину в один клик для дальнейшей покупки.

Сценарий:

1. Пользователь открывает файл с расширением *.md*, в котором есть товар из *RoboStore*.
2. При нажатии на наименование товара, он отправляется в пользовательскую корзину.

Просмотр профиля пользователей

Цель: Просмотр пользовательских профилей, знакомство с репозиториями пользователя.

Сценарий:

1. Пользователь открывает страницу пользовательского профиля.
2. Платформа предоставляет обзор всех репозиториев пользователя.

Просмотр главной страницы магазина

Цель: Ознакомить пользователей с ассортиментом магазина.

Сценарий:

1. Главная страница платформы содержит краткое описание товара.
2. Пользователь имеет возможность выбрать категорию товара, при выборе категории на странице остаются товары только выбранной категории.
3. Платформа позволяет отфильтровать товары. Фильтры для каждой категории товаров индивидуальны.

Просмотр страницы товара

Цель: Позволяет пользователем подробнее ознакомиться с товаром.

Сценарий:

1. С краткого описания товара на главной странице *RoboStore* пользователь может перейти к просмотру страницы товара, нажав на кнопку "Подробнее".
2. После нажатия кнопки пользователь переходит на страницу с подробным описанием товара (характеристики, описание).

Выбор фильтров для товаров

Цель: Позволить пользователям фильтровать товар по характеристикам, зависящим от категории товара.

Сценарий:

1. После выбора категории товаров, пользователю становится доступен выбор категорий для фильтров товаров.
2. Пользователь может установить необходимые ему фильтры и отправить форму.
3. После отправки формы платформа оставит товары, удовлетворяющие фильтру.

Добавление товаров в корзину

Цель: Формирование пользовательской корзины для дальнейшего оформления заказа.

Сценарий:

1. С краткого описания товара на главной странице *RoboStore* пользователь может перейти к просмотру страницы товара, нажав на кнопку "В корзину".
2. После этого платформа отправляет товары пользовательскую корзину.

Просмотр корзины

Цель: Ознакомить пользователя с корзиной для дальнейшего формирования заказа.

Сценарий:

1. Пользователь открывает страницу с личной корзиной.
2. В один клик пользователь может изменить количество товаров в заказе, удалить товар из корзины или начать оформлять заказ.

Расширения:

- Если корзина пуста, платформа предлагает перейти на главную страницу *RoboStore*.

Формирование заказа

Цель: Сформировать заказ для покупки товара.

Сценарий:

1. Пользователь нажимает кнопку формирования товара.
2. Открывается страница, которая предоставляет пользователю ссылку для связи с телеграм-ботом для дальнейшего формирования заказа.

Просмотр лучших проектов

Цель: Ознакомление с лучшими проектами *RoboGit*.

Сценарий:

1. Система создает топ лучших репозиториев исходя из количества *stars* в пользовательских проектах.
2. Пользователь может посетить страницу топа лучших репозиториев.

Добавление товара

Цель: Поддержание полной базы товаров *RoboStore*.

Сценарий:

1. Сотрудник *Robostore* нажимает кнопку добавление товара в *Robostore*.
2. Сотрудник заполняет форму с необходимой информацией о товаре.
3. Сотрудник отправляет форму и публикует товар.

Расширения:

- Если сотрудник не заполняет обязательные поля, система предотвращает публикацию товара в *Robostore*.

Обновление информации о товаре

Цель: Поддержание актуальной информации о товарах.

Сценарий:

1. Сотрудник *Robostore* нажимает кнопку изменения товара в *Robostore*.
2. Сотрудник получает форму с заполненной ранее информацией о товаре.
3. Сотрудник изменяет необходимые поля формы.
4. Сотрудник отправляет форму и система обновляет информацию о товаре.

Расширения:

- Если сотрудник оставляет пустыми обязательные поля, система предотвращает обновление информации о товаре в *Robostore*.

Удаление товара

Цель: Удаление неактуальной информации.

Сценарий:

1. Сотрудник *Robostore* нажимает кнопку удаления товара в *Robostore*.
2. Система удаляет товар.

Изменение прав пользователей

Цель: Поддержание актуальной базы сотрудников и администраторов.

Сценарий:

1. Администратор может поменять права пользователя на сотрудника, обычного пользователя или администратора.
2. Система дает необходимые права пользователю.

1.4 UML-диаграмма прецедентов

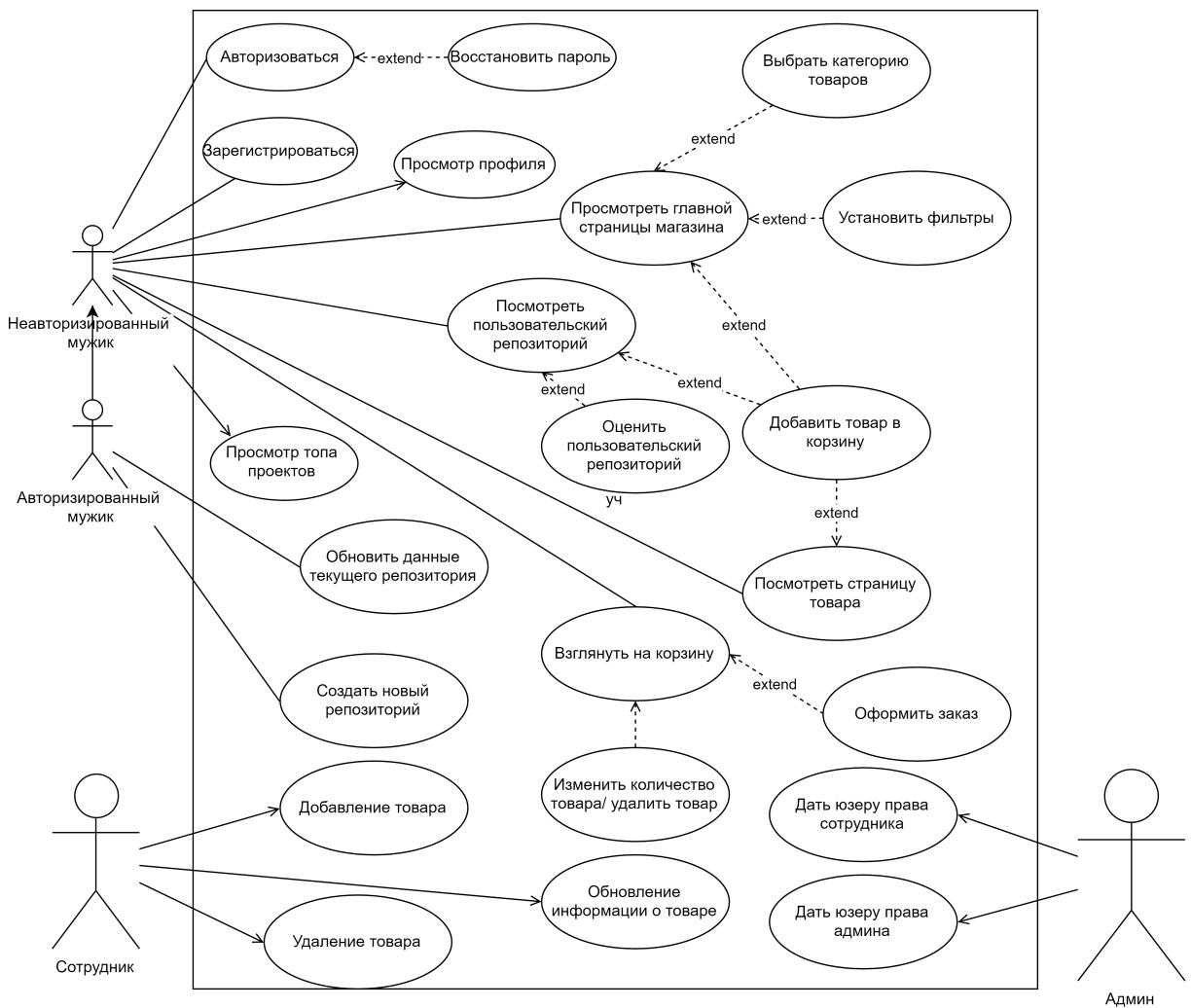


Рис. 1: UML-диаграмма прецедентов

2 Требования к аппаратно-программному обеспечению

2.1 Требования к серверному обеспечению

Система, которая обеспечивает выполнение программных продуктов сервера приложений и хранение данных платформы, должна отвечать следующим требованиям:

1. Наличие операционной системы *Linux*
2. Наличие сервера баз данных *PostgreSQL* версии 9.6 или выше
3. Наличие сервера приложений *GlassFish* версии 4.1.2 или выше
4. Наличие на сервере *JDK 8*

2.2 Требования к клиентскому обеспечению

Браузерный интерфейс разрабатывается с учетом следующих требований к программному обеспечению на стороне пользователя:

1. Веб-браузер *Tor* версии 8.01 или выше
2. Включенный интерпретатор сценариев *JavaScript*.
3. Отсутствие запрета веб-страницам платформы доступа к внешним ресурсам, а именно изображениям, шрифтам, таблицам стилей *CSS* и сценариям *JavaScript*, в том числе блокировщиками рекламы.

3 Требования к архитектуре системы

3.1 Уровень back-end

1. Уровень back-end основывается на *Spring* версии 5 или выше.
2. Взаимодействие между уровнями back-end и front-end организуется посредством *REST API*.
3. Для доступа к БД используется *Spring Data*.
4. Серверное приложение должно отправлять пользователям еженедельное новостное сообщение электронной почты, используя *JavaMail API*.
5. Серверное приложение должно логировать все вызовы методов на уровне бизнес-логики системы с использованием технологии *Spring AOP* и *AspectJ*.
6. Ролевое разграничение доступа к внутренним разделам системы организовано с помощью технологии *Spring Security*.
7. Доступ пользователей в систему осуществляется через систему "единого входа" (Single Sign On). В качестве провайдера SSO использована система *ForgeRock OpenAM*, развернутая на отдельном экземпляре (домене) сервера приложений.

3.2 Уровень front-end

1. Уровень front-end строится на *ReactJS + Redux* с использованием *ES6* и *JSX* и набора компонентов *Belle*.
2. Веб-интерфейс адаптирован для отображения в трех режимах:
 - *Десктопном*: ширина экрана больше 1105px
 - *Планшетном*: ширина экрана больше 687px и меньше 1105px
 - *Мобильном*: ширина экрана меньше 687px

3.3 Telegram-бот

Для сохранения анонимности пользователей оформление заказа происходит через телеграм-бота.

1. Telegram-бот отправляет инструкции и ссылку для оплаты заказа.
2. Telegram-бот позволяет определить время и место доставки заказа.

4 Требования к надежности и безопасности системы

1. Серверное приложение должно содержать механизмы защиты от уязвимостей, входящих в список OWASP TOP-10.
2. Пароли пользователей должны содержать не менее восьми символов и храниться как криптографический хэш.

5 Архитектура системы

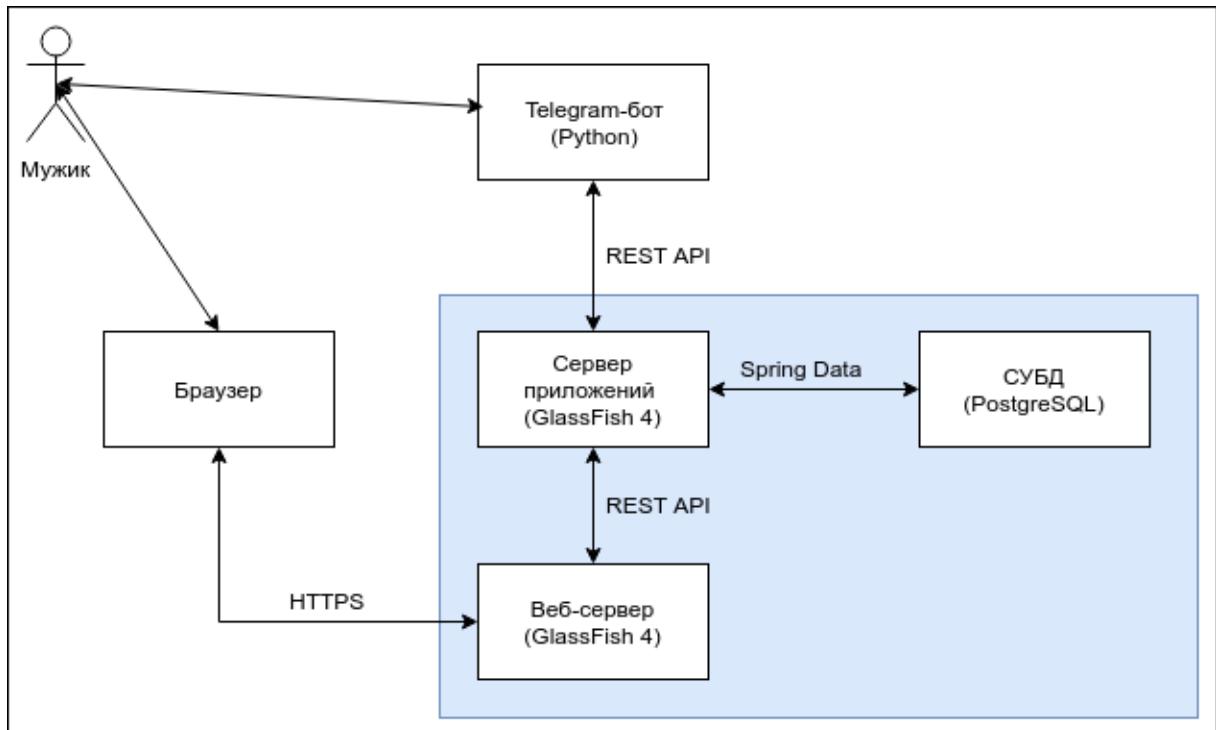


Рис. 2: Архитектура системы

6 Функциональные требования

6.1 Для неавторизированных пользователей

1. Система должна предоставлять пользователю возможность зарегистрироваться в системе.
2. Система должна предоставлять пользователям возможность авторизоваться с помощью логина и пароля.
3. Система должна предоставлять пользователям возможность восстановить пароль.
4. Система должна предоставлять пользователям возможность просмотра пользовательских профилей *RoboGit*.
5. Система должна предоставлять пользователям возможность просмора пользовательских репозиториев.
6. Система должна предоставлять возможность пользователям просматривать топ-проектов - проекты, имеющие наибольшее количество *stars*.
7. Система должна предоставлять возможность добавления товара в корзину из пользовательских проектов, выбрав и кликнув на необходимый товар в *markdown* файле.
8. Система должна предоставлять возможность просмотра главной страницы интернет-магазина *RoboStore*, в которой расположены все товары.
9. Система должна позволять выбор категории товаров для просмотра.
10. Система должна позволять пользователю выбирать фильтры для товаров. Возможности фильтра зависят от конкретной категории товара.
11. Система должна предоставлять возможность пользователю просматривать страницы товаров с более подробным описанием.
12. С главной страницы товара или со страницы товара система должна позволять добавлять товар в корзину.
13. Система должна предоставлять возможность просмотра корзины.
14. В пользовательской корзине должно быть возможно изменить количество товара и/или удалить товар.
15. С пользовательской корзины система должна предоставлять возможность оформления заказа, после чего пользователь получает номер своего заказа и ссылку на телеграм-бота для дальнейшего оформления заказа.
16. Система должна предоставлять возможность экспортовать корзину как внешнюю ссылку, нажав на которую, пользователь заполнит свою корзину товарами из экспортированной корзины.

6.2 Для авторизованных пользователей

1. Система должна позволять авторизованному пользователю создавать новый репозиторий.
2. Система должна позволять авторизированному пользователю обновлять данные текущего репозитория.
3. Система должна позволять авторизированному пользователю оценивать пользовательские репозитории, нажимая на кнопку *stars* в пользовательском проекте.

6.3 Для сотрудников

1. Система должна позволять сотруднику добавлять товар в *RoboStore*.
2. Система должна позволять сотруднику обновлять информацию о товаре в *Robostore*.
3. Система должна позволять сотруднику удалять товар в *RoboStore*.

6.4 Для администраторов

1. Система должна позволять администратору сделать пользователя сотрудником/администратором (назначить привилегии).
2. Система должна позволять администратору сделать другого администратора/сотрудника обычным пользователем (отнять привилегии).

7 Прототипы интерфейсов

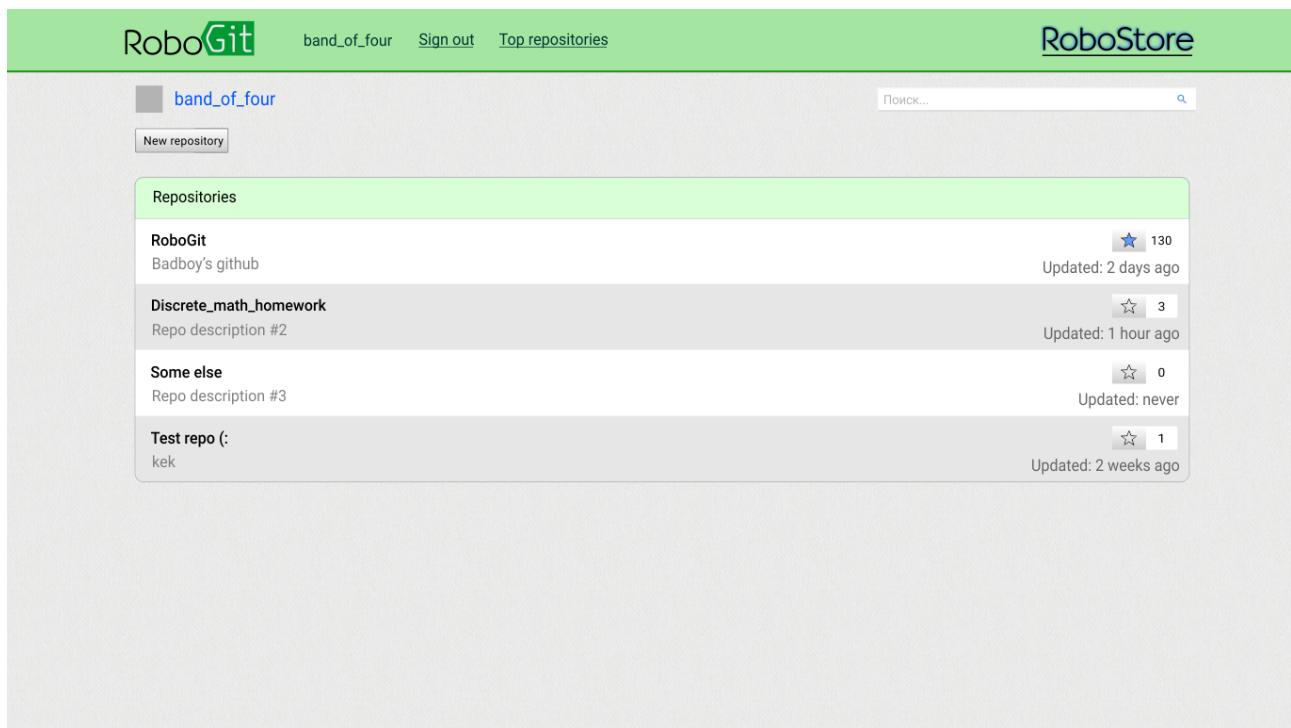


Рис. 3: Профиль пользователя RoboGit

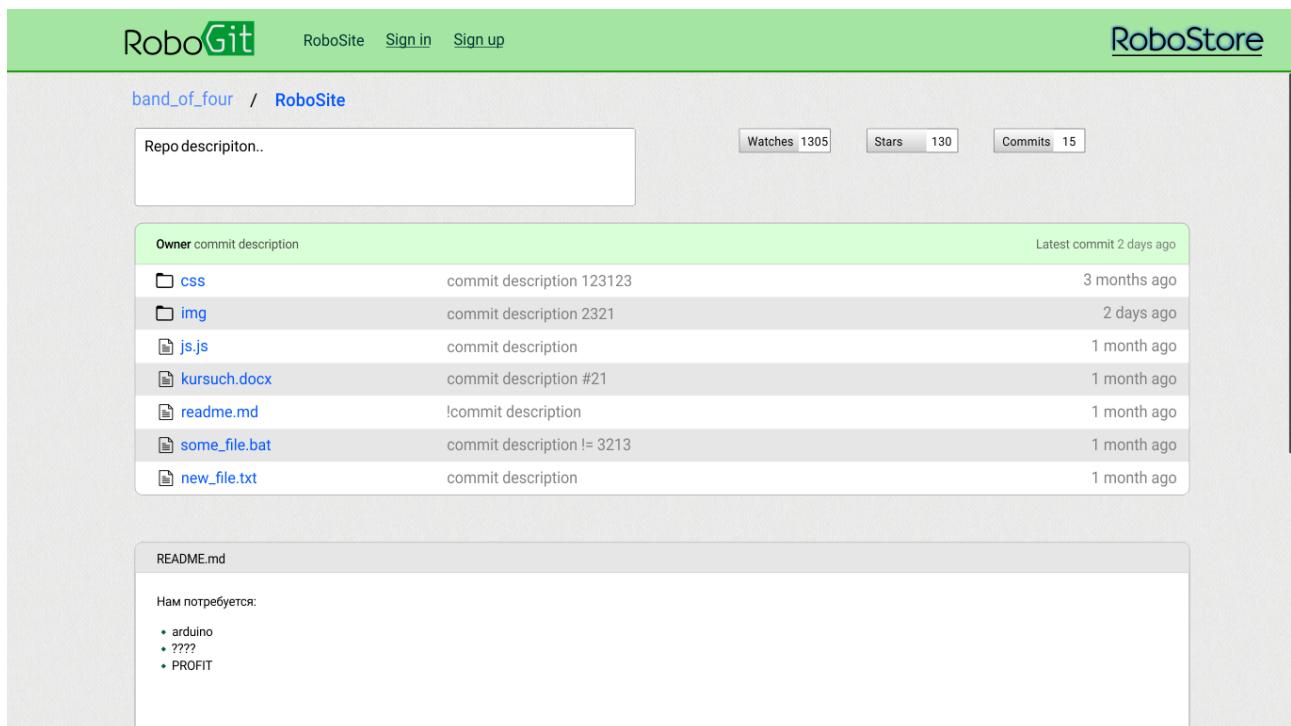


Рис. 4: Страница репозитория RoboGit

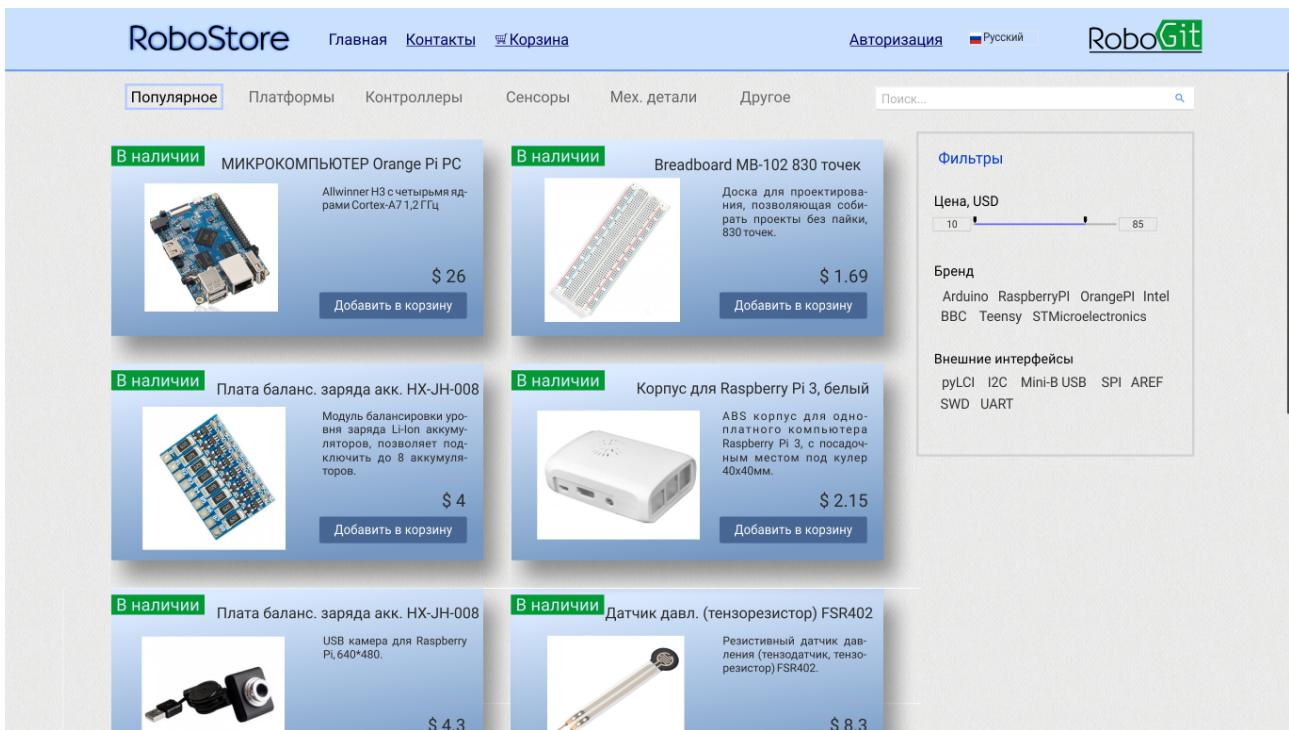


Рис. 5: Главная страница RoboStore

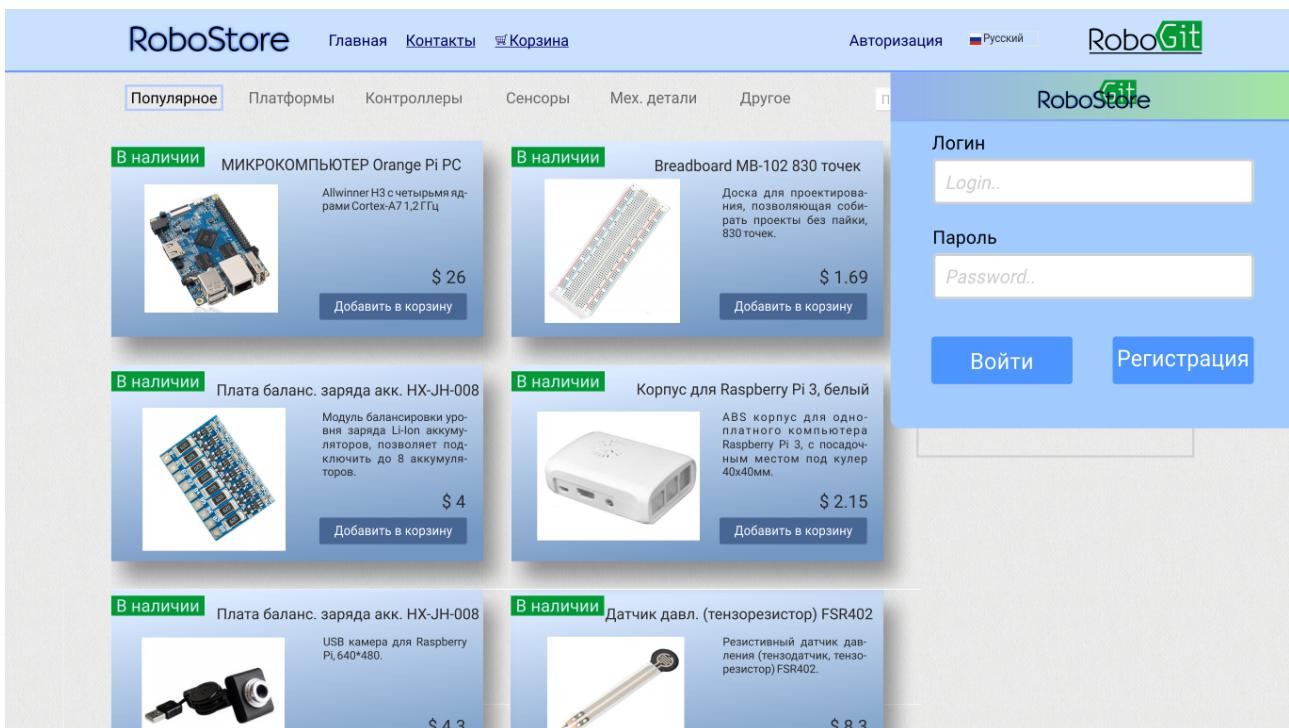


Рис. 6: Форма авторизации



Рис. 7: Страница товара RoboStore

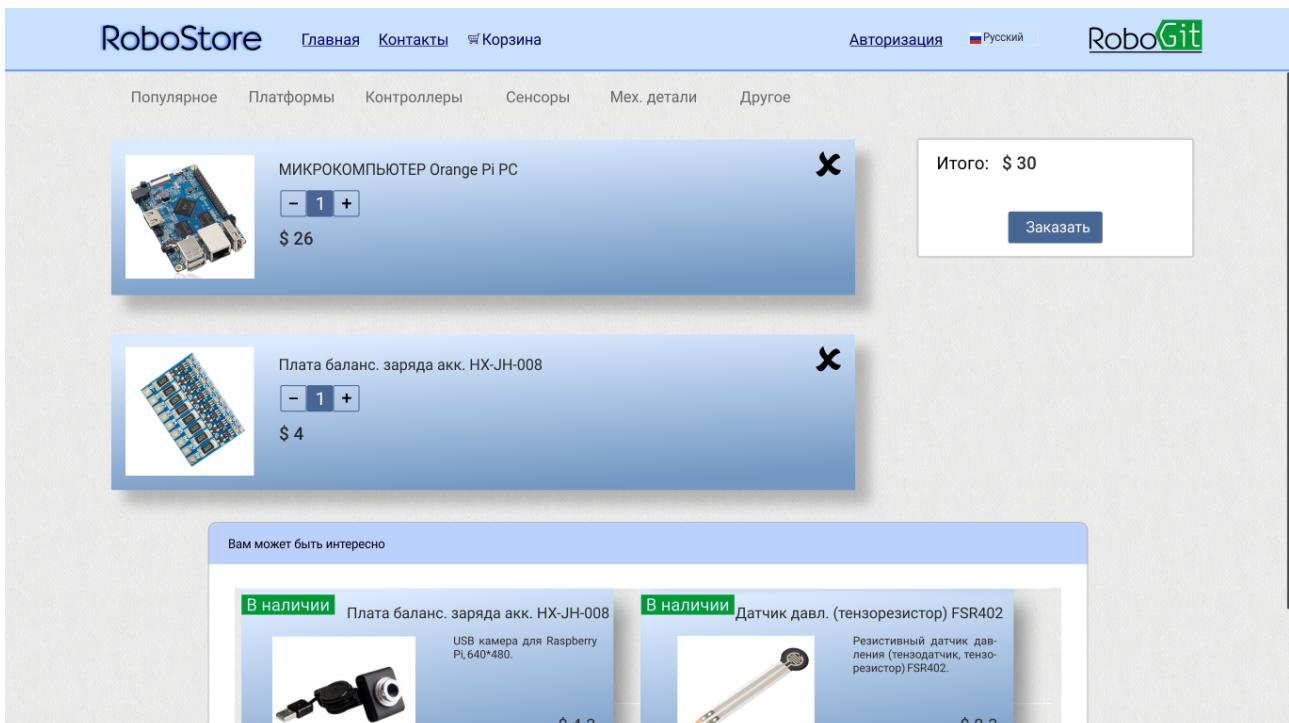
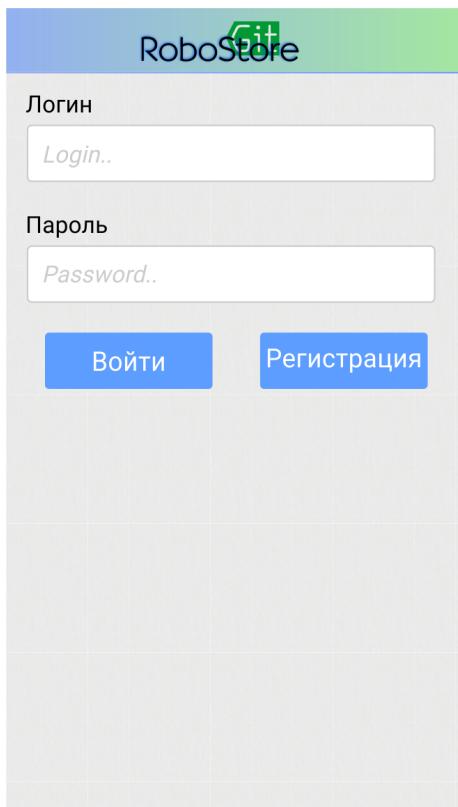


Рис. 8: Корзина RoboStore

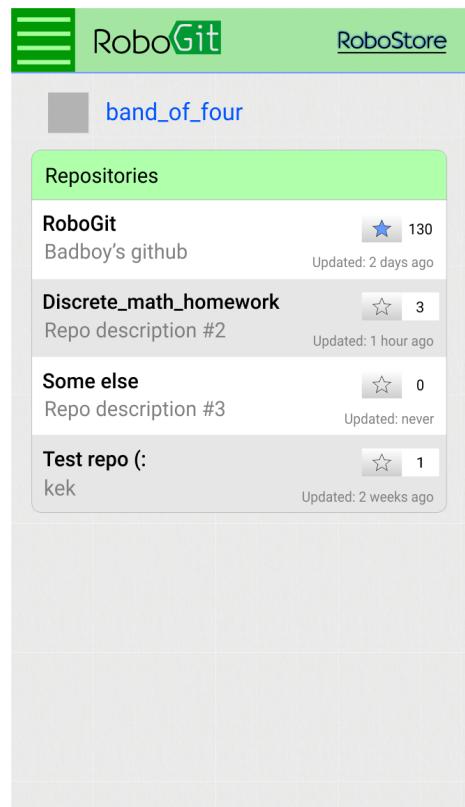


RoboStore

Логин

Пароль

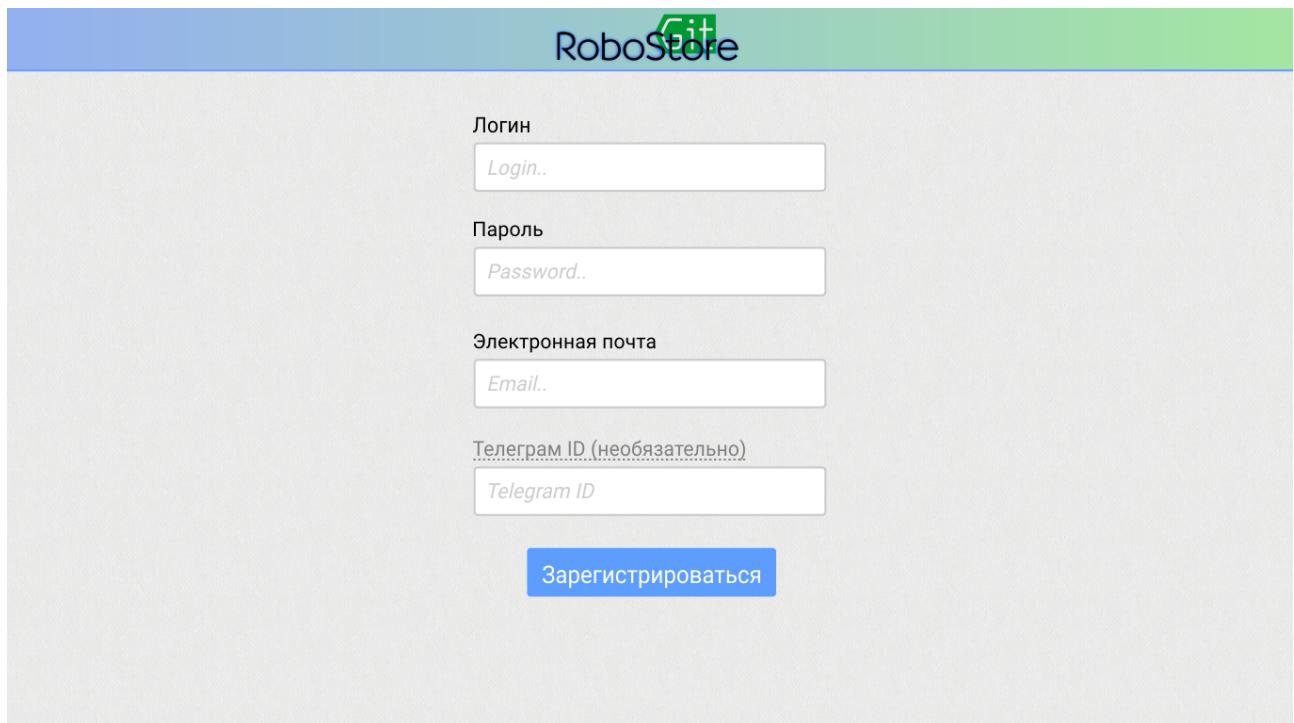
[Войти](#) [Регистрация](#)



(a) Форма авторизации

(b) Профиль RoboGit

Рис. 10: Макеты для смартфона №1



RoboStore

Логин

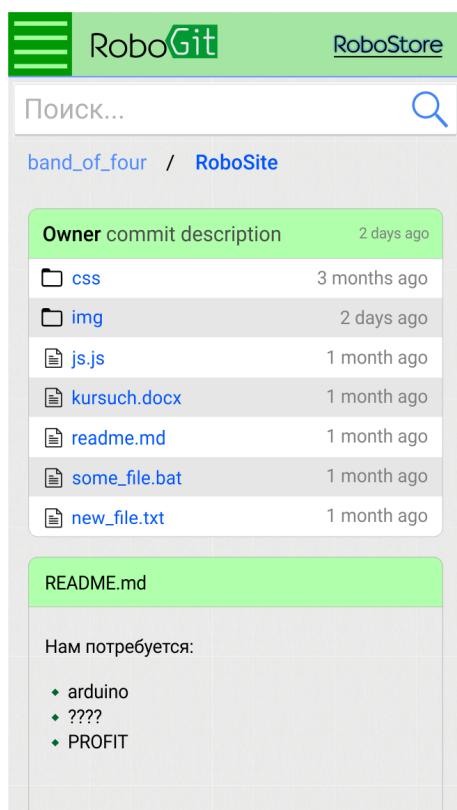
Пароль

Электронная почта

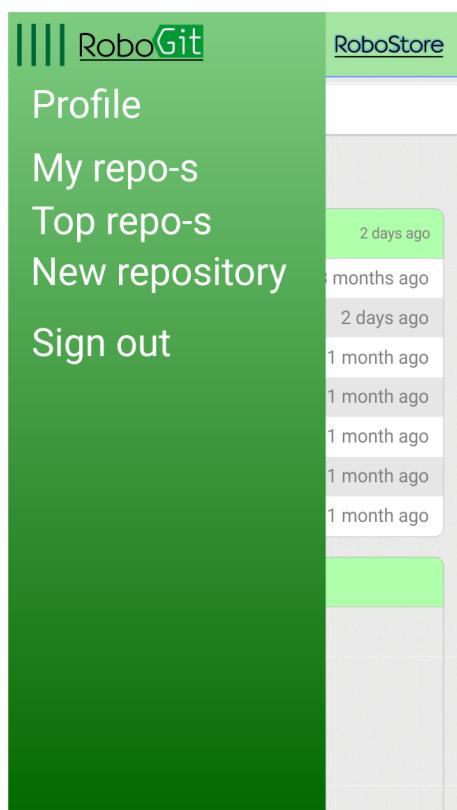
Телеграм ID (необязательно)

[Зарегистрироваться](#)

Рис. 9: Форма регистрации

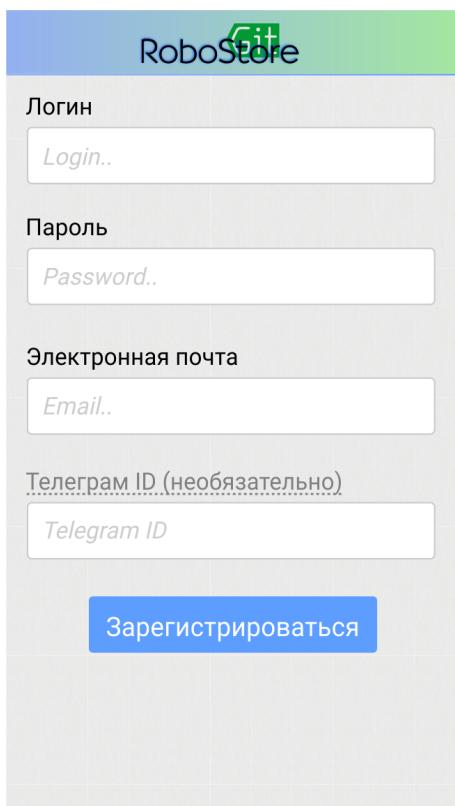


(a) Репозиторий RoboGit



(b) Боковое меню RoboGit

Рис. 11: Макеты для смартфона №2



RoboStore

Логин
Login..

Пароль
Password..

Электронная почта
Email..

Телеграм ID (необязательно)
Telegram ID

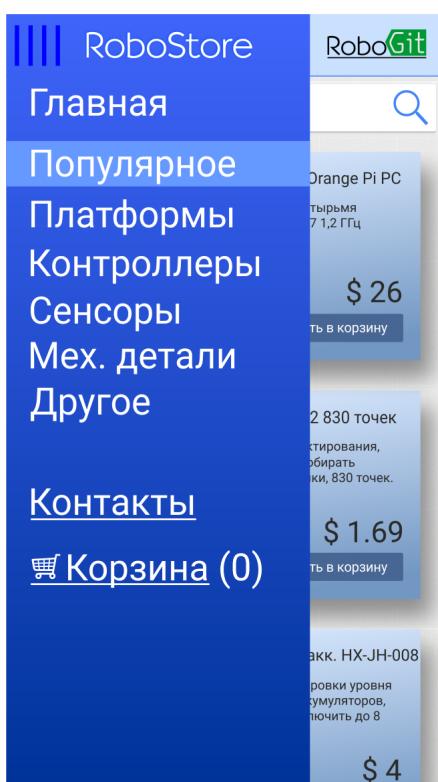
Зарегистрироваться



(a) Форма регистрации

(b) Главная страница RoboStore

Рис. 12: Макеты для смартфона №3



(a) Боковое меню RoboStore



(b) Страница товара RoboStore

Рис. 13: Макеты для смартфона №4



(a) Корзина RoboStore

Рис. 14: Макеты для смартфона №5