



МИНОБРНАУКИ РОССИИ

Федеральное государственное бюджетное образовательное учреждение высшего образования
«МИРЭА – Российский технологический университет»

РТУ МИРЭА

Институт перспективных технологий и индустриального программирования
Кафедра индустриального программирования

КУРСОВОЙ ПРОЕКТ (РАБОТА)

по дисциплине

Технологии индустриального программирования

Тема курсового проекта (работы) Разработка прототипа сервиса для учета коллекции Lego

Студент группы Муждабаев Сергей Георгиевич, ЭФМО-02-24 _____
(Ф.И.О., учебная группа) (подпись студента)

Руководитель
курсового
проекта
(работы) Адышкин Сергей Сергеевич,
преподаватель, доцент _____
(Ф.И.О., должность, звание, ученая степень) (подпись руководителя)

Курсовой
проект (работа)
представлен(а)
к защите « 4 » июня 20 25 г.

Допущен(а) к
защите « 1 » июня 20 25 г.

Получена оценка:	
Подпись преподавателя:	
Комментарий по работе:	

ВВЕДЕНИЕ	3
1 ОБЗОР СУЩЕСТВУЮЩИХ РЕШЕНИЙ	5
2 ОПИСАНИЕ ПРОГРАММИРУЕМОЙ СРЕДЫ	9
2.1 Требования к программируемой среде	9
2.2 Описание действий пользователя для работы с системой	11
2.2.1 Импорт данных из внешнего API Rebrickable	11
2.2.2 Просмотр каталога LEGO	11
2.2.3 Добавление наборов в коллекцию	11
2.2.4 Удаление наборов из коллекции и редактирование описания.....	11
2.2.5 Получение статистики по коллекции	12
2.2.6 Поиск по каталогу и фильтрация	12
2.2.7 Учет недостающих элементов	12
2.2.8 Регистрация и авторизация	12
3 ПРАКТИЧЕСКАЯ ЧАСТЬ	13
3.1 Реализация требований к системе.....	13
3.2 Функциональное тестирование программного продукта	14
3.3 Инструкция по эксплуатации системы	17
Заключение	23
Список использованных источников.....	24

ВВЕДЕНИЕ

В последние годы наблюдается рост популярности конструктора LEGO как среди детей, так и среди их родителей [1]. Конструктор Датского производителя стал не просто игрушкой, но и объектом творчества, коллекционирования, и даже инвестирования. Это вызвало появление онлайн-сервисов для учёта и управления личными коллекциями наборов LEGO. Однако в условиях текущей геополитической ситуации, многие зарубежные сервисы по учёту коллекций стали недоступны для пользователей из России. Это существенно осложнило возможность вести цифровую инвентаризацию личных коллекций, обмениваться данными и поддерживать актуальное состояние наборов. В связи с этим актуальность разработки собственного решения, обеспечивающего импортозамещение недоступных платформ учёта LEGO, значительно возросла.

Пользователи нуждаются в надёжной, функциональной и независимой системе, способной аккумулировать актуальные данные о существующих наборах и деталях, структурировать их для последующего отображения и анализа.

Особенно востребованными становятся решения, построенные на микросервисной архитектуре, позволяющей гибко масштабировать и модернизировать систему под нужды рынка.

Объектом исследования является сервис учета коллекции LEGO с возможностью интеграции внешних сервисов для актуализации собственных баз данных о существующих наборах и деталях конструктора

Предмет исследования – методы разработки информационной системы на микросервисной архитектуре, поддерживающей интеграцию внешнего API, масштабируемость и ориентированной на удобство использования продукта.

Целью настоящей работы является разработка прототипа сервиса, предназначенного для получения, хранения и обработки информации о сериях и наборах LEGO с возможностью интеграции с внешними сервисами для получения актуальных данных.

Для достижения цели необходимо решить следующие задачи:

- провести анализ существующих решений;
- сформировать архитектуру системы с учётом ключевых функциональных требований;
- реализовать микросервисы для получения данных из внешних систем по API, работы с локальной базой данных, авторизации пользователей, пользовательских действий;
- разработать интуитивно-понятный интерфейс системы;
- провести тестирование и оценить эффективность работы.

Таким образом, данная работа затрагивает практические вопросы импортозамещения зарубежных цифровых решений и демонстрирует возможности построения современной, автономной системы учёта коллекций LEGO на базе микросервисной архитектуры.

1 ОБЗОР СУЩЕСТВУЮЩИХ РЕШЕНИЙ

На сегодняшний день существует несколько популярных онлайн-сервисов, предназначенных для ведения учёта коллекций LEGO. Большинство из них предоставляют широкий функционал для энтузиастов и коллекционеров: отслеживание собранных наборов, детализация по частям, планирование сборок и просмотр статистики. Рассмотрим наиболее популярные сервисы в области учета и коллекционирования LEGO.

Brickset [2] — один из старейших и наиболее полных онлайн-каталогов наборов LEGO. Платформа позволяет зарегистрированным пользователям вести собственную коллекцию, отслеживать редкие или недостающие наборы, оставлять отзывы и просматривать аналитику по своей коллекции. Интеграция с внешними базами данных и регулярные обновления обеспечивают точность информации.

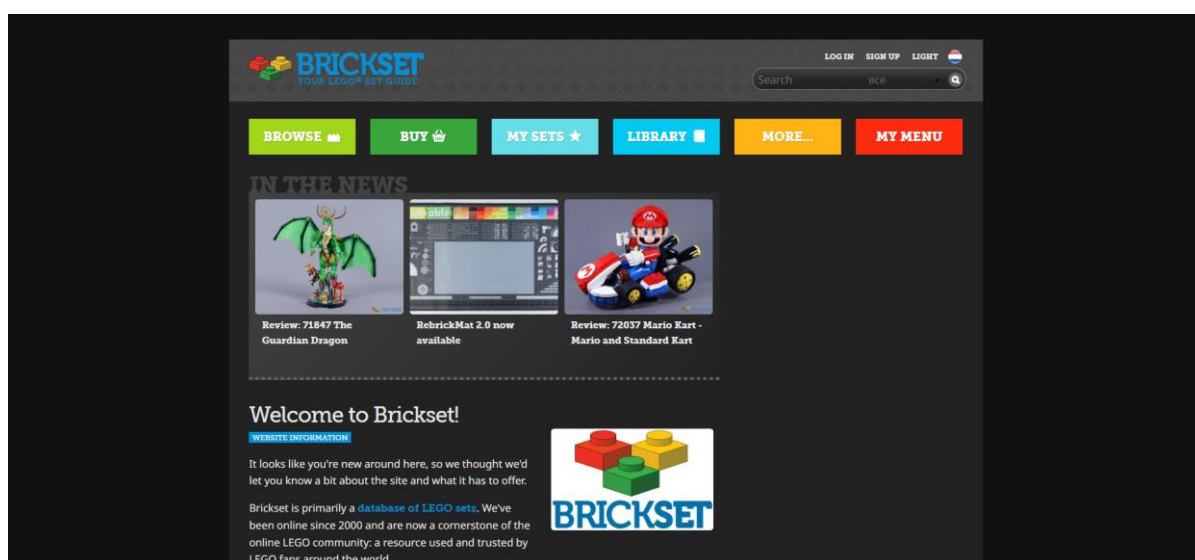


Рисунок 1.1 – Главная страница сайта «Brickset»

Rebrickable [3] — предлагает уникальный подход к работе с коллекцией: система анализирует детали, доступные у пользователя, и подсказывает, какие наборы он может собрать. Сайт также предлагает API, что делает его удобным для разработки собственных решений и автоматизации.

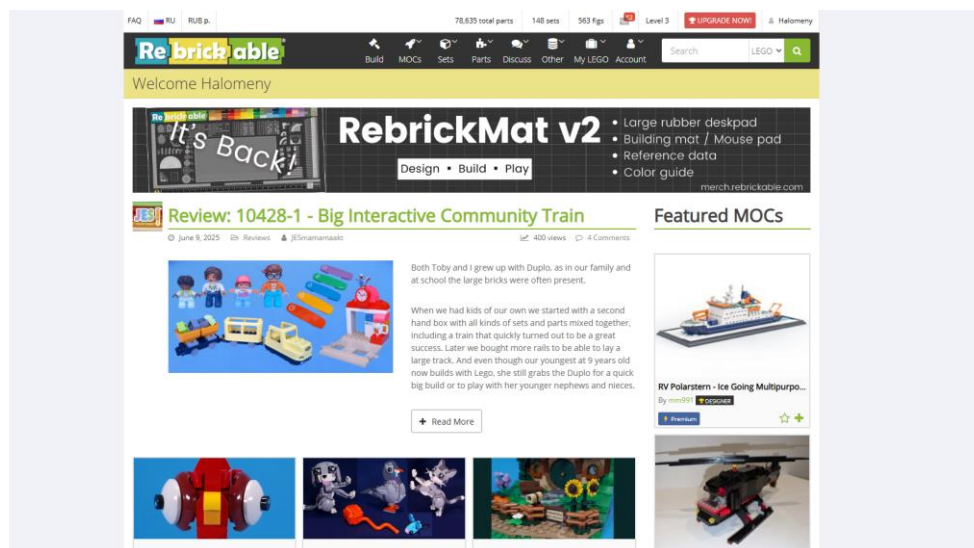


Рисунок 1.2 – Главная страница сайта «Rebrickable»

BrickLink [4] — крупнейшая торговая площадка LEGO, на которой также доступен модуль учета коллекции. На данный момент данный сервис принадлежит компании LEGO. Интеграция с программой моделирования BrickLink Studio позволяет вести учёт деталей, собирать виртуальные модели и заказывать недостающие элементы.

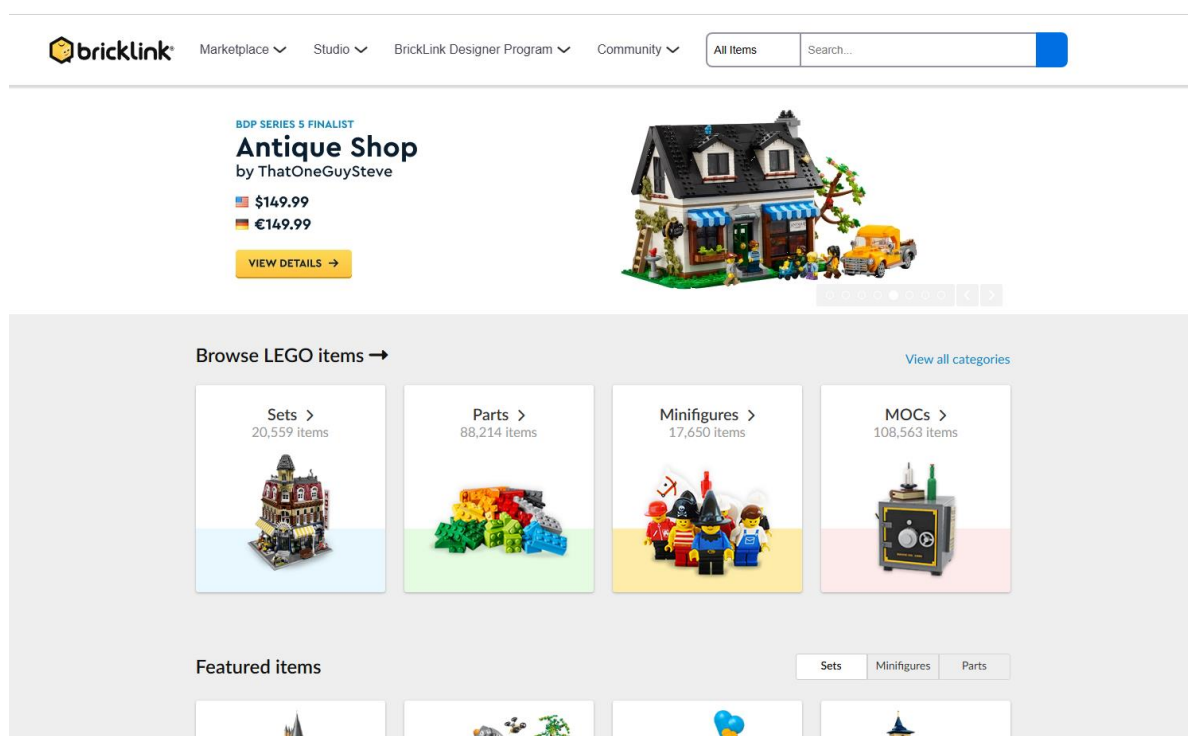


Рисунок 1.3 – Главная страница сайта «Bricklink»

LEGO Life — официальное мобильное приложение LEGO от производителя также предоставляют функции управления цифровой коллекцией. Однако их функциональность ограничена и преимущественно ориентирована на детей.

Сравним рассмотренные системы по основным функциям, которые предоставляются продуктами, а также по доступности на территории России в таблице 1.1.

Таблица 1.1 – Сравнительный анализ существующих на рынке решений

Критерий	Brickset	Rebrickable	Bricklink	Lego Life
Учет наборов	+	+	+	Ограниченный (добавление наборов происходит путем сканирования QR кодов на коробках)
Учет деталей	Ограниченный (исключительно через самоделки)	+	+	-
Импорт/экспорт данных	Ограниченный (Excel)	+ (Excel, файл Bricklink)	+	-
Поиск по базе наборов	+	+	+	Ограниченный
API для разработчиков	-	+	-	-
Список пожеланий (wishlist)	+	+	+	+
Мобильное приложения	-	-	+	+
Доступность в РФ	-	Ограниченная	-	-
Учет недостающих деталей в разрезе наборов	-	-	-	-
Возможность заказа деталей/наборов напрямую из сервиса	-	-	+	-
Ведение статистики по коллекции	+	+	Ограниченная	-
Наличие русского языка	-	-	-	-
Интерфейс	Устаревший, перегруженный	Устаревший, перегруженный	Современный	Геймифицированный

Рассмотренные решения по учёту коллекций LEGO обладают как сильными, так и слабыми сторонами. Brickset — один из самых популярных сервисов, позволяющий вести учёт наборов, получать сводную статистику и взаимодействовать с сообществом, однако он не поддерживает учёт деталей и в настоящее время может быть недоступен из России. Rebrickable предоставляет наиболее полный функционал: пользователь может отслеживать как наборы, так и отдельные детали, строить модели из своей коллекции, пользоваться рекомендациями и получать подробную статистику. При этом интерфейс может быть перегружен для новичков, а доступ к сервису также ограничен у некоторых Российских провайдеров. Сервис BrickLink ориентирован на взаимодействие с торговой площадкой LEGO-компонентов, позволяет учитывать коллекцию, но практически не предоставляет аналитических инструментов и не предлагает открытого API. Официальное мобильное приложение LEGO Life от LEGO Group фокусируется на визуализации и пошаговом построении моделей, ориентировано на детей. Таким образом, несмотря на широкие возможности перечисленных сервисов, ни одно из решений не является универсальным и полностью доступным для российских пользователей. Это подчёркивает актуальность создания отечественного аналога, который обеспечивал бы свободный доступ, удобство использования, поддержку всех ключевых функций и мог бы развиваться независимо от зарубежной инфраструктуры.

2 ОПИСАНИЕ ПРОГРАММИРУЕМОЙ СРЕДЫ

2.1 Требования к программируемой среде

Разработка программного обеспечения требует чёткого определения технических и функциональных требований к создаваемой системе. В условиях перехода к импортозамещающим решениям особое внимание уделяется использованию свободно распространяемых технологий и инструментов, доступных для российских разработчиков. В рамках настоящего проекта, посвящённого созданию микросервисного приложения для учёта коллекции LEGO, важным этапом становится формализация требований к программируемой среде, в которой будет функционировать данное решение.

Ниже представлены ключевые требования, определяющие архитектурные, технологические и пользовательские особенности разрабатываемой системы. Эти требования формируют основу для последующего проектирования, реализации и тестирования программного продукта, а также обеспечивают его стабильную и расширяемую работу.

1. Проект должен быть полностью реализован на языке Go. Это обеспечивает простоту сопровождения, высокую производительность, а также возможность реализации микросервисной архитектуры с низкими накладными расходами.
2. Система должна стабильно запускаться, обрабатывать запросы, взаимодействовать с внешними API (Rebrickable) и базой данных PostgreSQL.
3. Проект должен использовать ООП-подход в проектировании бизнес-логики: реализация структур, инкапсуляция методов, декомпозиция ответственности между компонентами.
4. Должен быть реализован пользовательский интерфейс и REST API с документацией Swagger, через который пользователь может выполнять

операции учёта коллекции LEGO: загрузка данных, просмотр, фильтрация и удаление.

5. Программа должна быть разделена на независимые микросервисы: сервис для получения данных из Rebrickable, сервис для работы с базой данных и внутренней логикой, сервис для аутентификации и взаимодействия с пользователем. Это улучшает масштабируемость, отказоустойчивость и упрощает сопровождение системы.
6. Должен быть реализован RESTful API, через который возможно получать, сохранять и удалять информацию о коллекции. API должен быть документирован и следовать общепринятым стандартам.
7. Приложение должно использовать СУБД PostgreSQL как хранилище данных о наборах и сериях. Структура базы данных должна автоматически формироваться через миграции (GORM AutoMigrate) и обеспечивать целостность внешних ключей.
8. Система должна автоматически загружать данные из открытого Rebrickable API, фильтровать и трансформировать их в формат, пригодный для сохранения в локальной базе данных. Импорт должен учитывать иерархию (серии, подсерии).
9. Архитектура системы должна допускать простое добавление новых возможностей: учёт деталей, пользовательских МОС-моделей, статистики по наборам и визуализации данных. Это достигается благодаря модульности и соблюдению принципов SOLID.

Сформулированные требования к программируемой среде обеспечивают комплексный подход к созданию надёжного и масштабируемого микросервисного решения. Они учитывают не только выбор оптимальных технологий, но и определяют ключевые принципы архитектурной организации системы — микросервисную структуру, REST API и автоматические миграции данных. Таким образом, данный набор требований служит прочной основой для эффективной реализации программного продукта, ориентированного на импортозамещение в сфере учёта коллекций LEGO

2.2 Описание действий пользователя для работы с системой

2.2.1 Импорт данных из внешнего API Rebrickable

Администратор запускает процедуру импорта данных через специальную функцию «Импорт данных». Эта операция инициирует микросервис, который обращается к API Rebrickable, получает актуальный список серий, наборов и деталей LEGO, а затем передаёт эти данные в сервис хранения для записи в локальную базу данных. Таким образом обеспечивается регулярное обновление информации о доступных сериях и наборах LEGO. В дальнейшем будет настроено расписание для автоматического обновления данных.

2.2.2 Просмотр каталога LEGO

Пользователь может просматривать весь список серий, наборов и деталей LEGO. Для каждого набора отображаются основные сведения: название, описание, серия, артикул, состав набора (детали и минифигурки). Просмотр доступен как списком, так и с возможностью открытия отдельной страницы для каждого набора. Для просмотра каталога пользователь может быть не авторизован.

2.2.3 Добавление наборов в коллекцию

Авторизованный пользователь может добавить набор в личную коллекцию, создав для этого список. Добавление набора возможно как из страницы списка (по кнопке «добавить» через поиск по артикулу), так и при просмотре общего каталога. Список может быть как публичным (доступным для других пользователей сервиса) или приватным (доступным исключительно создателю списка). Публичными списками можно делиться по кнопке «поделиться списком». Данный функционал дает возможность как создавать списки желаемого, так и демонстрировать свою коллекцию в собственных интересах.

2.2.4 Удаление наборов из коллекции и редактирование описания

Авторизованный пользователь может удалять ранее добавленные в список наборы по соответствующей кнопке «Удалить». Также предусмотрена возможность добавления собственных заметок к наборам из списков, которая позволяет сохранять текстовую информацию в разрезе наборов пользователя

2.2.5 Получение статистики по коллекции

Система предоставляет пользователю статистические отчёты по его коллекции: количество наборов, количество минифигурок распределение по сериям, количество имеющихся и отсутствующих (которые пользователь пометил вручную) деталей, распределение наборов по годам выпуска. Эта функция помогает оценить масштабы и особенности личной коллекции.

2.2.6 Поиск по каталогу и фильтрация

Для удобства навигации в коллекции и по каталогу реализован поиск по ключевым параметрам — названию набора, артикулу, серии, году выпуска. Фильтры позволяют выделять наборы по заданным критериям, что облегчает навигацию среди большого количества элементов.

2.2.7 Учет недостающих элементов

Авторизованный пользователь имеет возможность отмечать отсутствующие детали в разрезе добавленных в пользовательский список наборов. Для этого ему необходимо из списка перейти на страницу набора, на которой отметить недостающие элементы в списке деталей.

2.2.8 Регистрация и авторизация

Незарегистрированный пользователь может пройти процесс регистрации в сервисе. Для этого пользователь вводит логин, который является уникальным среди всех пользователей сервиса, электронную почту и пароль. Для авторизации пользователь использует логин или электронную почту в связке с паролем.

3 ПРАКТИЧЕСКАЯ ЧАСТЬ

3.1 Реализация требований к системе

Реализация сервиса велась в соответствии с ранее сформулированными требованиями в разделе 2.1. В таблице 3.1. приведено описание реализации каждого ключевого требования.

Таблица 3.1 – реализация требований

Требование	Реализация
Проект должен быть полностью реализован на языке Go	Вся серверная часть разработана на языке Go
Система должна взаимодействовать с внешними API (Rebrickable) и базой данных PostgreSQL	Для взаимодействия с API Rebrickable был разработан микросервис «set-service», для взаимодействия с базой данных разработан микросервис «data-service»
Проект должен использовать ООП-подход в проектировании бизнес-логики	При разработке учтена специфика реализации ООП подхода при использовании языка программирования Golang, а именно: для организации данных применялись структуры, использование которых прописывалось в методах
Программа должна быть разделена на независимые микросервисы	В результате разработки было создано 4 микросервиса: data-service (взаимодействие с базой данных), set-service (взаимодействие с API Rebrickable), user-service (Пользовательские действия), authentication-service (Регистрация, авторизация)
Должен быть реализован RESTful API, через который возможно получать, сохранять и удалять информацию о коллекции	Взаимодействие между микросервисами реализовано через RESTful API, документация разработана с использованием swagger
Структура базы данных должна автоматически формироваться через миграции	База данных создается с использованием автомиграций (GORM)
Система должна автоматически загружать данные из открытого Rebrickable API	Для загрузки данных по API Rebrickable в микросервисе «set-service» были реализованы эндпоинты: «/api/import/details», «/api/import/minifigs», «/api/import/series», «/api/import/sets»
Архитектура системы должна допускать простое добавление новых возможностей	При разработке была выбрана микросервисная архитектура, которая поддерживает бесшовное масштабирование

Таким образом, реализация приведенных требований позволила создать устойчивый, безопасный и простой в масштабировании сервис для учета коллекции Lego.

3.2 Функциональное тестирование программного продукта

Для проверки корректности работы серверной части был реализован и протестирован полный набор REST-эндпоинтов. Документирование и тестирование проводились с использованием встроенного интерфейса Swagger UI, интегрированного в каждый из микросервисов разрабатываемого продукта. Была выполнена загрузка данных через API Rebrickable, их сохранение в локальную базу, созданы тестовые пользователи и пользовательские списки. На рисунке 3.5 представлена структура проекта.

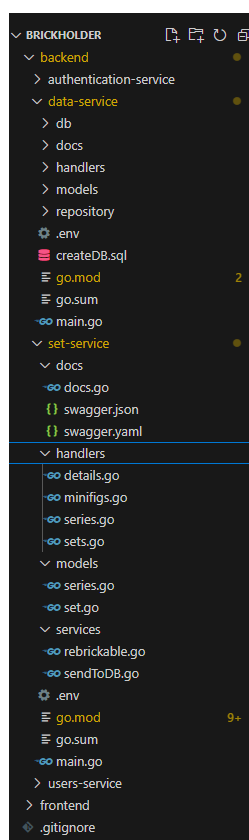


Рисунок 3.1 – Структура проекта

На рисунках 3.2 – 3.5 представлены основные эндпоинды и их функциональное назначение.

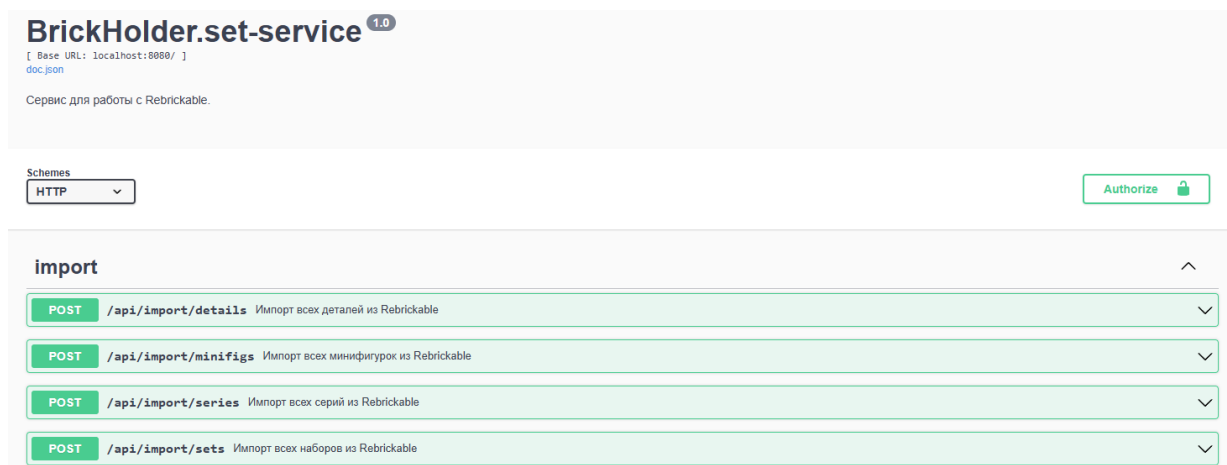


Рисунок 3.2 – Эндпоинты микросервиса «Set-service»

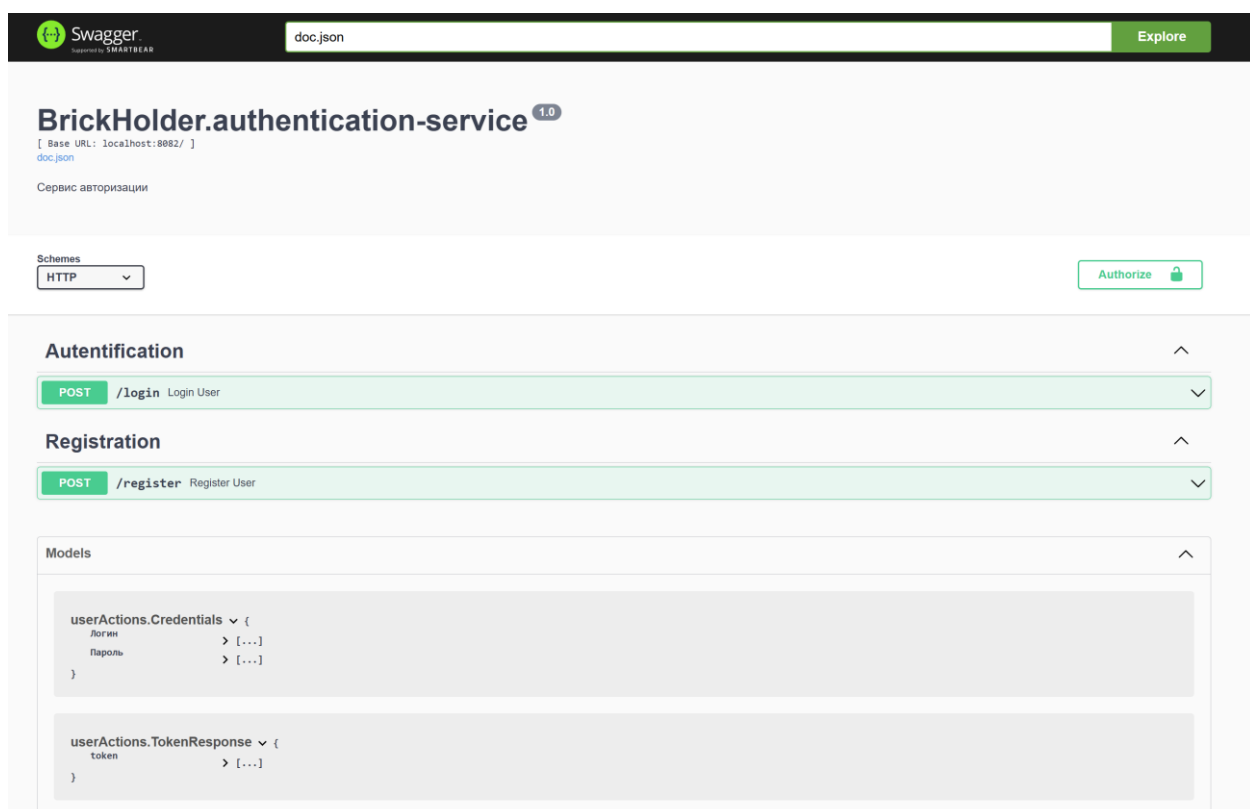


Рисунок 3.3 – Эндпоинты микросервиса «autentification-service»

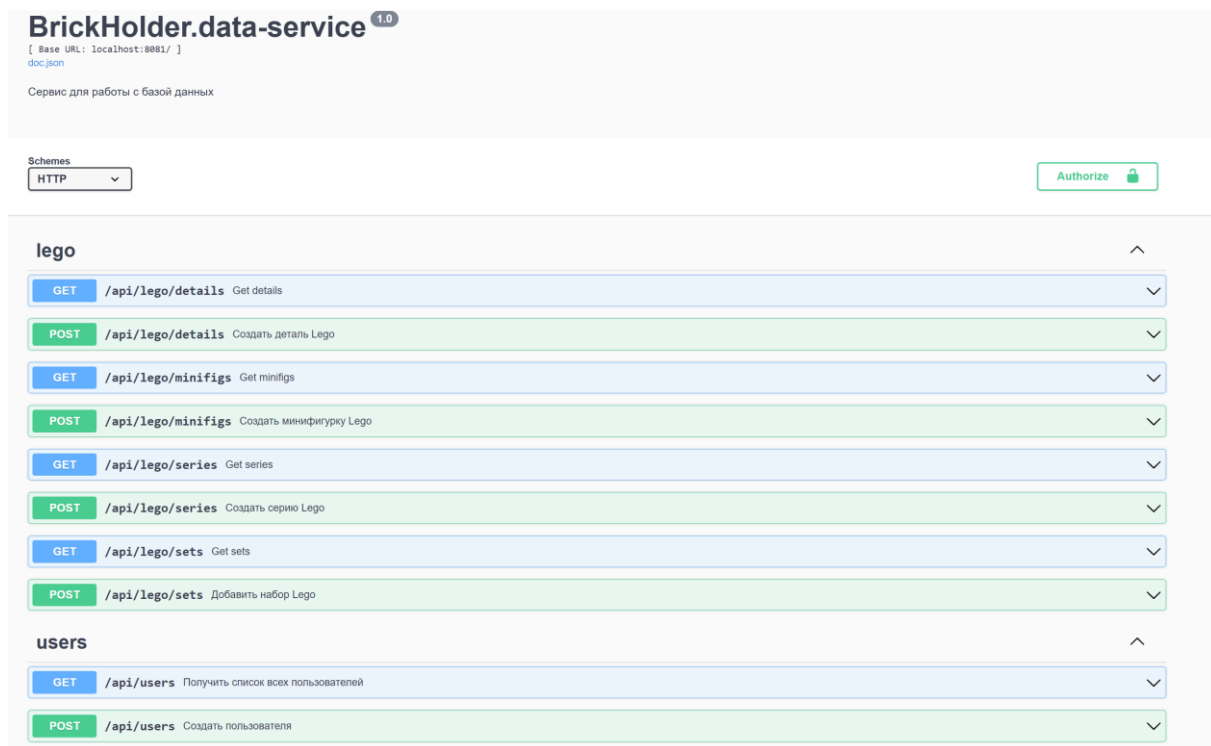


Рисунок 3.4 – Эндпоинты микросервиса «data-service»

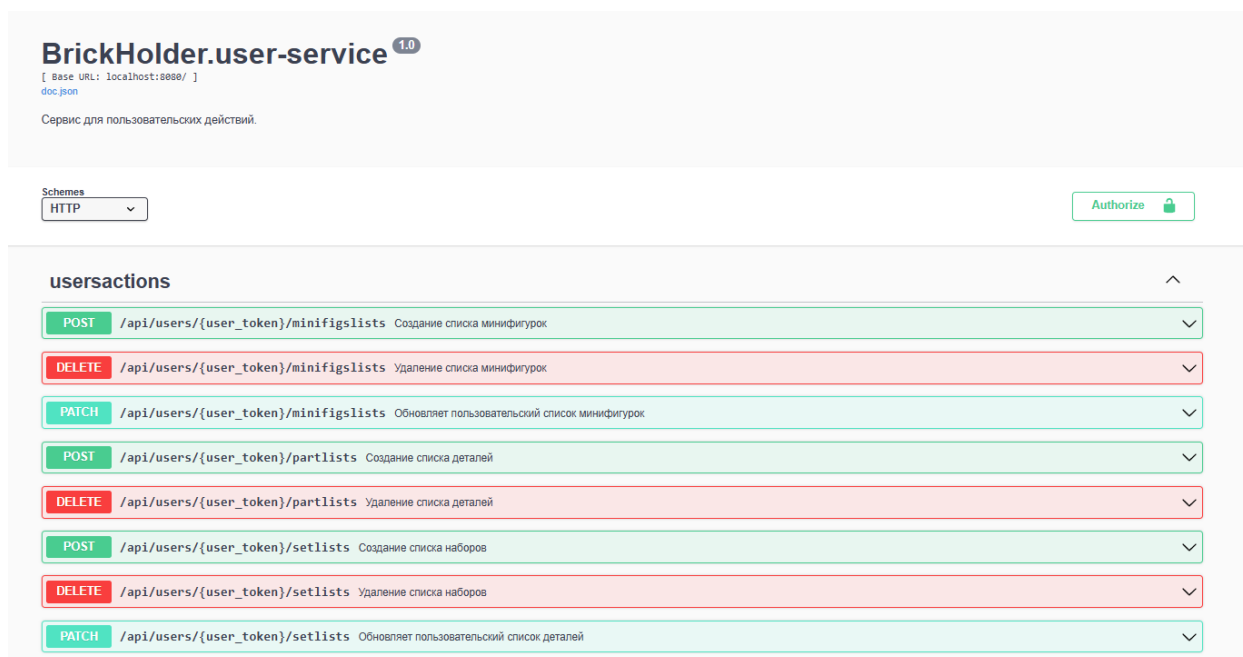


Рисунок 3.5 – Эндпоинты микросервиса «users-service»

В ходе функционального тестирования при помощи Swagger UI были проверены все основные маршруты системы: регистрация, авторизация, просмотр каталогов наборов, создание, редактирование и удаление пользовательских списков, добавление наборов в коллекцию.

3.3 Инструкция по эксплуатации системы

В рамках реализованного функционала доступны следующие функции:

- просмотр каталогов наборов, минифигурок и деталей Lego;
- поиск по артикулу деталей/наборов/минифигурок;
- регистрация и авторизация пользователей;
- создание, редактирование пользовательских списков наборов, фигурок, деталей;
- добавление элементов в пользовательские списки через поиск по артикулу.

В рамках данного раздела рассмотрены ключевые экранные формы разрабатываемого продукта.

При переходе на сайт проекта пользователь попадает на главную страницу, на которой находятся новости и ссылки на существующие разделы. Главная страница изображена на Рисунке 3.6.

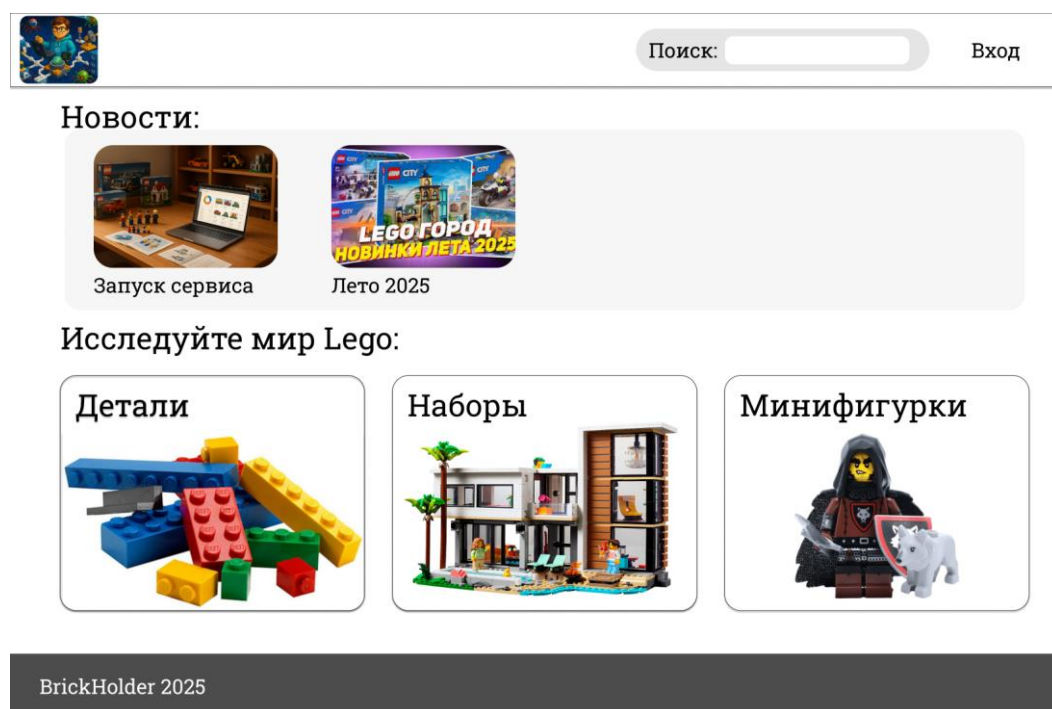
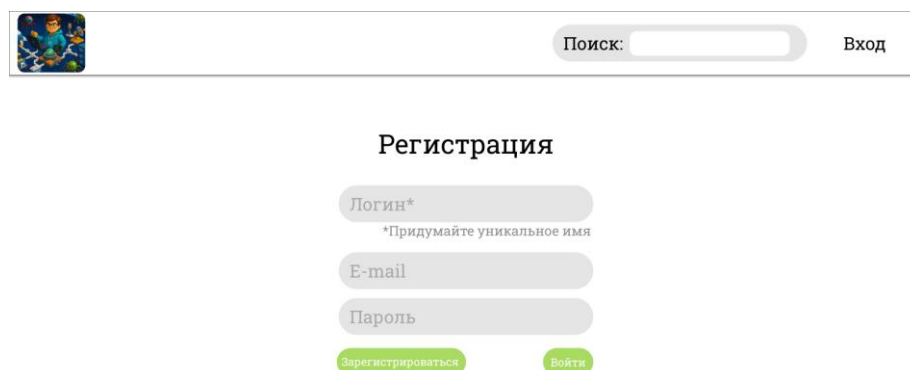


Рисунок 3.6 – Главная страница

При нажатии на кнопку «Вход» пользователя перенаправляют на страницу входа/регистрации. Для регистрации требуется придумать уникальный логин, ввести электронную почту и пароль. По кнопке «Зарегистрироваться» происходит

регистрация, если логин является уникальным. При вводе существующей связки логин-пароль и нажатии кнопки «Войти» происходит вход в систему. Страница регистрации изображена на Рисунке 3.7.



BrickHolder 2025

Рисунок 3.7 – Страница регистрации

При успешном входе в систему пользователя перенаправляет на главную страницу. Для авторизованных пользователей в шапке сайта появляются ссылки на коллекцию пользователя. Главная страница для авторизованного пользователя изображена на Рисунке 3.8.

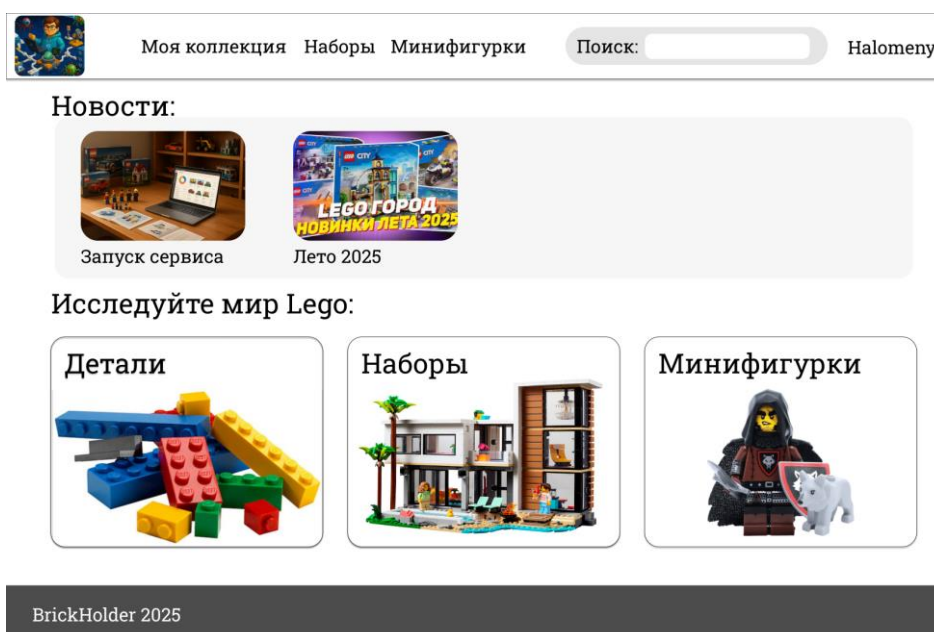


Рисунок 3.8 – Главная страница авторизованного пользователя

При переходе по ссылке «Моя коллекция» пользователя перенаправляет на страницу его коллекции. Страница коллекции пользователя изображена на Рисунке 3.9.

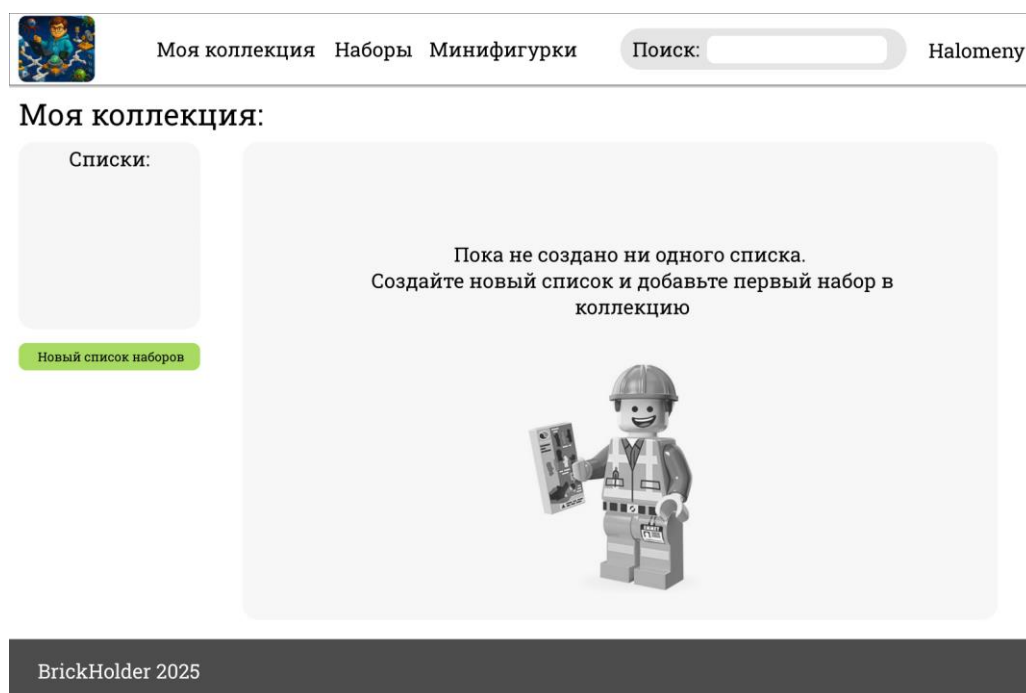


Рисунок 3.9 – Коллекция пользователя

Для создания нового списка необходимо нажать на кнопку «Новый список наборов», в результате чего откроется страница создания списка. Страница создания списка изображена на Рисунке 3.10.

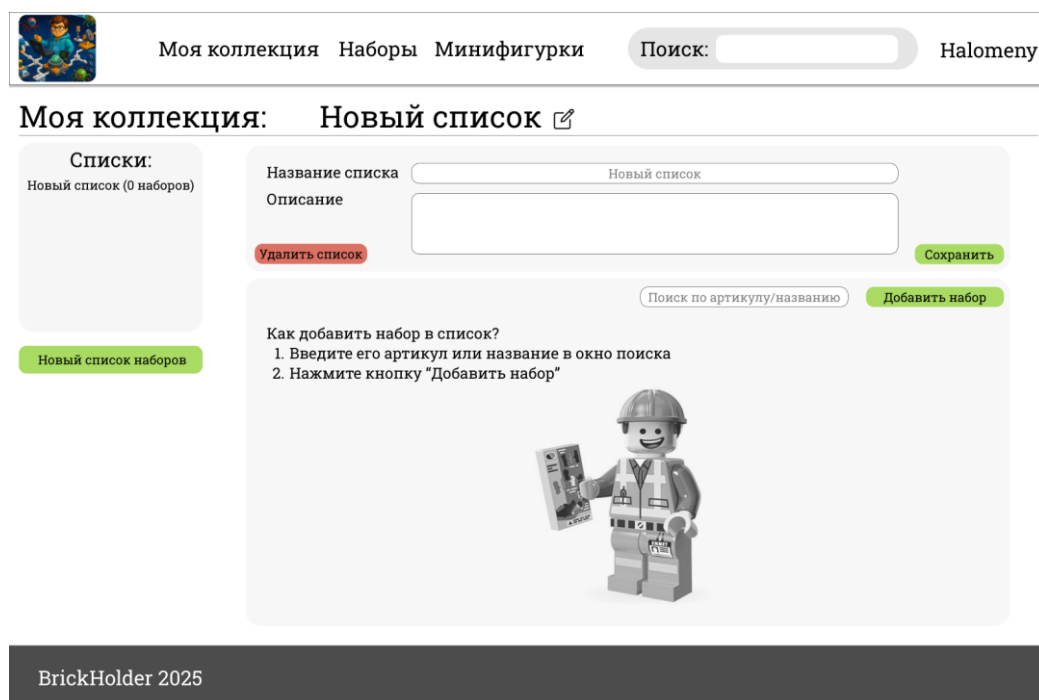


Рисунок 3.10 – Создание пользовательского списка наборов

При создании списка требуется ввести его название (по умолчанию указывается «Новый список»). Для сохранения изменений в названии или описании списка требуется нажать кнопку «Сохранить». Для добавления набора в список необходимо воспользоваться поиском по артикулу или названию набора, расположенном в правом верхнем углу раздела с наборами. При вводе артикула в строку появится выпадающий список с найденными совпадениями. Процесс добавления набора в список изображен на Рисунке 3.11.

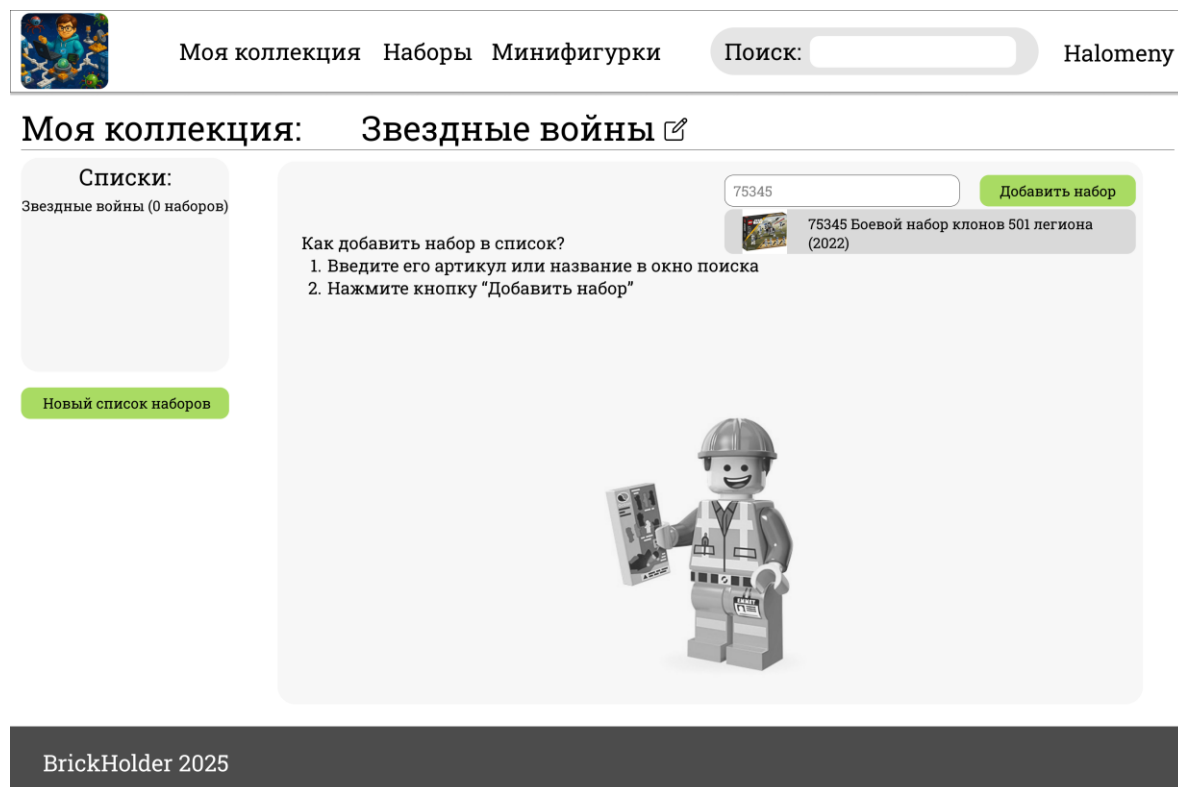
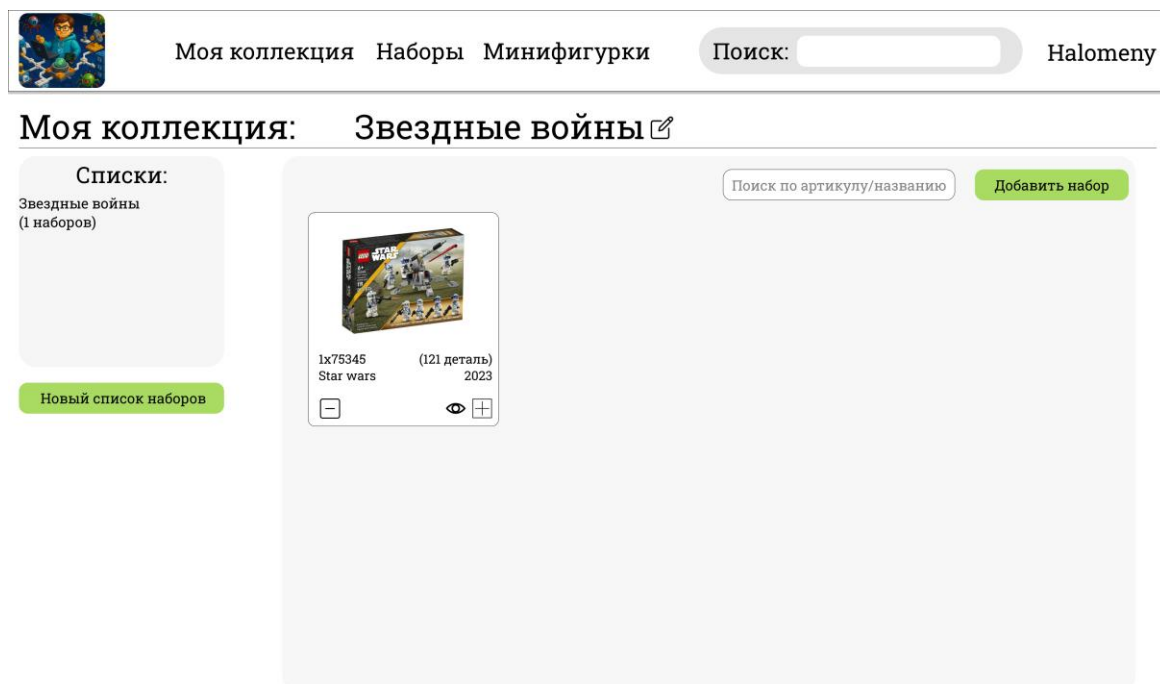


Рисунок 3.11 – Добавление набора в список

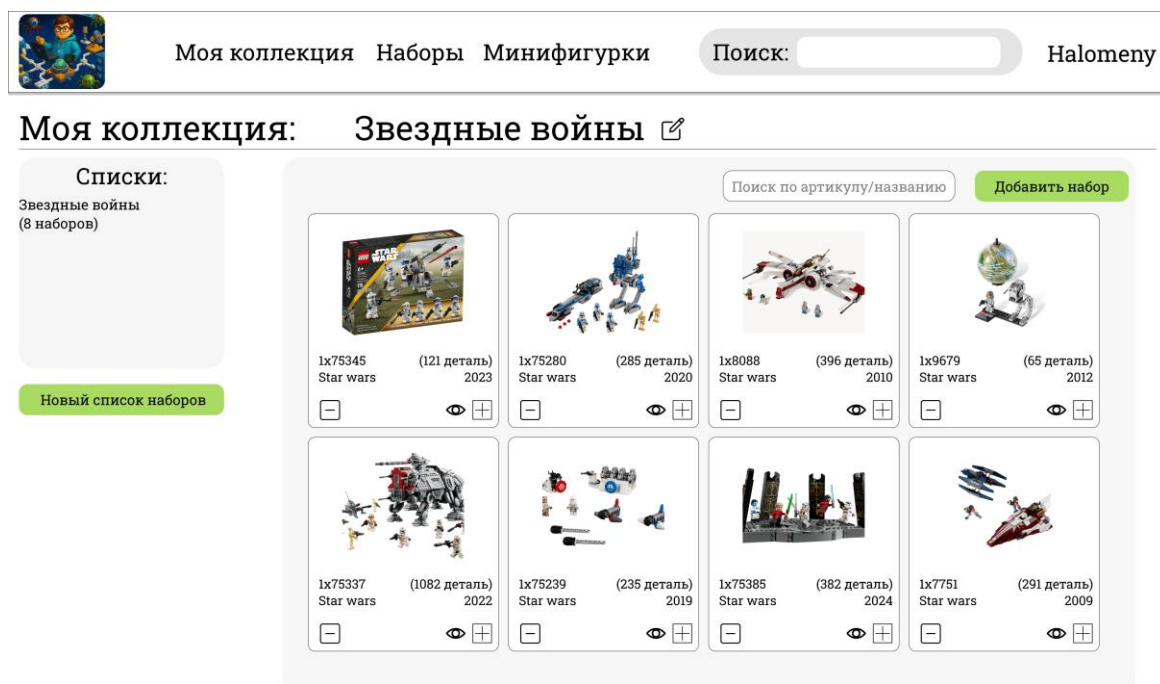
При нажатии на элемент выпадающего списка или на кнопку «Добавить набор» произойдет добавление набора в коллекцию. После чего в разделе с наборами появится элемент с информацией о наборе, а информационный текст пропадет. Заполнение раздела наборов изображено на Рисунке 3.12.



BrickHolder 2025

Рисунок 3.12 – Заполнение списка наборами


Пример заполненного списка изображен на Рисунке 3.13.



BrickHolder 2025

Рисунок 3.13 – Заполненный раздел с наборами


Для каждого набора в списке отображается следующая информация: количество наборов в коллекции, артикул, серия, количество деталей, год выпуска. При нажатии на иконку «Минус» происходит удаление набора из списка. Нажимая на кнопку «Плюс» имеется возможность увеличить количество наборов в коллекции. При нажатии на кнопку «Глаз» открывается страница набора с подробной информацией о нем. Страница набора представлена на Рисунке 3.14.


Моя коллекция
Наборы
Минифигурки


Поиск:

Вход

Лего набор 75345 - Боевой набор клонов 501 легиона




75345	
Название	Боевой набор клонов 501 легиона
Год выпуска	2023
Количество деталей	121
Количество минифигурок	4




2xfig-013396

Просмотр деталей (121)



1xfig-013395

Просмотр минифигурок (4)



1xfig-013397

Инструкция

Добавить набор в коллекцию

BrickHolder 2025

Рисунок 3.14 – Страница набора

На странице набора имеется возможность просмотреть его фотографии, узнать количество и просмотреть поштучно детали и минифигурки, входящие в набор, скачать электронную инструкцию по сборке, а также добавить набор в коллекцию.

В данном разделе был продемонстрирован основной функционал разрабатываемой системы, приведена подробная инструкция по созданию пользовательских списков, наполнения их наборами, а также просмотр информации по конкретному набору.

ЗАКЛЮЧЕНИЕ

В результате выполнения курсовой работы была спроектирована и реализована MVP-версия цифровой системы для учёта коллекции LEGO на основе открытого API Rebrickable. В рамках проекта был проведен анализ предметной области, исследование существующих решений-аналогов, разработаны требования к проектированию системы, создана микросервисная архитектура, реализованы основные компоненты системы, проведено тестирование и подготовка пользовательской документации.

На этапе анализа были рассмотрены наиболее популярные международные платформы для коллекционеров, такие как Brickset, Rebrickable и Bricklink. Это позволило выявить их ключевые преимущества и недостатки, а также определить требования к функционалу, соответствующему ожиданиям российских пользователей, учитывая факт ухода ряда зарубежных сервисов с российского рынка.

Несмотря на то, что проект реализован в формате MVP, его архитектура позволяет в дальнейшем масштабировать решение под более широкую аудиторию, включая не только коллекционеров, но и образовательные учреждения или клубы по интересам.

В процессе работы были успешно решены задачи интеграции с внешними API, построения микросервисного взаимодействия, управления связями между сущностями в базе данных, а также автоматизации миграций и генерации Swagger-документации. Работа над проектом позволила углубить знания в области проектирования микросервисов, взаимодействия с REST API, организации баз данных и Docker-оркестрации.

Таким образом, все цели курсовой работы были достигнуты. Разработанная система продемонстрировала функциональность, удобство использования и гибкость архитектурных решений. Полученный опыт станет основой для дальнейших профессиональных и исследовательских инициатив в области разработки цифровых продуктов для нишевых сообществ.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. WSJ: Продажи Lego растут, несмотря на общее падение мирового рынка игрушек [Электронный ресурс]. URL: <https://finance.mail.ru/2025-03-11/wsj-rodazhi-lego-rastut-nesmotrya-na-obschee-padenie-mirovogo-rynka-igrushek-65225047/> (Дата обращения 25.05.2025)
2. Brickset – база данных наборов Lego [Электронный ресурс]. URL: <https://brickset.com/> (Дата обращения 25.05.2025)
3. Rebrickable – сервис, помогающий создавать из имеющихся деталей уникальные самоделки [Электронный ресурс]. URL: <https://rebrickable.com/home/> (Дата обращения 25.05.2025)
4. Bricklink – наиболее популярный сервис по продаже наборов Lego [Электронный ресурс]. URL: <https://www.bricklink.com/v2/main.page> (Дата обращения 25.05.2025)
5. Документация GIN [Электронный ресурс]. - URL: <https://gin-gonic.com/ru/docs/> (Дата последнего обращения: 27.05.2025)
6. Статья о JWT [Электронный ресурс]. - URL: <https://habr.com/ru/articles/842056/> (Дата последнего обращения: 27.05.2025)
7. Документация PostgreSQL [Электронный ресурс]. - URL: <https://www.postgresql.org/docs/> (Дата последнего обращения: 27.05.2025)
8. Документация GORM [Электронный ресурс]. - URL: <https://gorm.io/docs/index.html> (Дата последнего обращения: 27.05.2025)
9. Реализация паттерна проектирования MVC с использованием фреймворка spring MVC / А. И. Тымкив, А. В. Федоренко, Ю. Г. Худасова, О. Г. Худасова // Системная трансформация - основа устойчивого инновационного развития : сборник статей Международной научно-практической конференции, Новосибирск, 20

декабря 2021 года. – Уфа: Общество с ограниченной ответственностью "Аэтерна", 2021. – С. 104-108. (Дата последнего обращения: 04.06.2021)

10. Документация Postman [Электронный ресурс]. - URL: <https://gorm.io/docs/index.html> (Дата последнего обращения: 04.06.2025)