**ВВЕДЕНИЕ**

В качестве темы проекта была выбрана разработка интернет-магазина брелоков. Цель проекта – реализовать основные функции для взаимодействия пользователя с веб-приложением для размещения своих брелков на продажу, а именно:

* Регистрация и авторизация пользователя, а также изменение информации авторизированного пользователя при необходимости;
* Возможность размещения брелка на продажу и изменение его информации при необходимости;
* Поиск по ключевым словам среди пользователей и брелков;
* Коммуникация пользователей и оценка брелков.

**ОСНОВНАЯ ЧАСТЬ**

* 1. **Список решаемых задач**

В ходе анализа предметной области проекта были сформулированы следующие функциональные задачи, которые должна решать система:

* Регистрация пользователя в системе и сохранение информации о нём в базе данных;
* Авторизация пользователя в системе и создание сессии Django и возможность выхода из системы;
* Создание авторизованным пользователем брелка, где указывается его название, описание, цена, а также при желании – 3D модель, которую можно просмотреть непосредственно на странице. После отправки формы информация заносится в базу данных;
* Отображение информации о брелке в виде отдельной страницы с указанием всех его характеристик (названия, описания, цены, автора, кнопки для отображения 3D модели);
* Коммуникация с автором посредством отправки сообщения на электронную почту непосредственно через интерфейс системы;
* Возможность оценить брелок в виде комментария на его странице;
* Создание личной страницы пользователя, где он может просмотреть свою информацию, размещённые на продажу брелки и оставленные комментарии;
* Поиск по брелкам или пользователям в системе;
* Отображение истории просмотра каждого пользователя;
* Валидация введённых данных полей ввода в системе для предотвращения возникновения противоречивой информации, нарушающих целостность данных.
  + 1. **Подключаемые модули и платформы**

Разработка ведётся на языке Python с использованием фреймворка Django. Разработка ведётся на ОС Windows в среде разработки JetBrains PyCharm. База данных реализована через СУБД MySQL и располагается на локальном сервере OpenServer. Основные модули, которые были применены при разработке:

* Django 4.1.7 – основной фреймворк, на котором ведётся разработка;
* Brelki – основной модуль приложения, в котором созданы views (функции для отрисовки шаблона), templates (html файлы шаблонов), models (модели базы даных), forms (формы для введения пользователем информации о себе или создаваемых им брелках)
* Os – для взаимодействия с файлами в директории проекта;
* Mysql-connector – для подключения к СУБД MySQL из программного кода;
* Decouple – для разнесения приватной информации по разным файлам.

**РЕАЛИЗУЕМАЯ АРХИТЕКТУРА**

Основной функционал приложения, получение данных из базы данных, и их последующая обработка описаны в вид-функциях в файле views. В каждой вид-функции данные получаются и отправляются в шаблоны виде модели Django. Каждая модель и её реализация в базе данных (используемые поля, типы данных полей, ограничения целостности) описана в файле models. Формы, которые используются для создания или манипулирования данными в приложении, описаны в файле forms. В нём описано то, какие ограничения нужно накладывать на пользовательские данные, чтобы в базе данных не возникло нарушений целостности. Ограничения накладываются в виде валидаторов Django, которые привязываются отдельно к каждому полю в форме. Валидаторы проверяют введённый результат и отправляют текст ошибок в шаблон, чтобы пользователь мог понять, в каком поле введены некорректные данные. Для того, чтобы системе понять, какую вид-функцию вызывать при переходе по конкретной ссылке, имеется файл urls, он связывает каждую ссылку с вид-функцией в файле views. Файл settings используется для задания настроек приложения (в конкретном примере были заданы настройки локализации отображения дат, пути хранилища пользовательских картинок и статичных файлов, таких как файлы css, js и статичных картинок, а также почтового протокола и т.д.).

Для визуализации функциональной части приложения была спроектирована диаграмма, представленная на Рисунке 1.

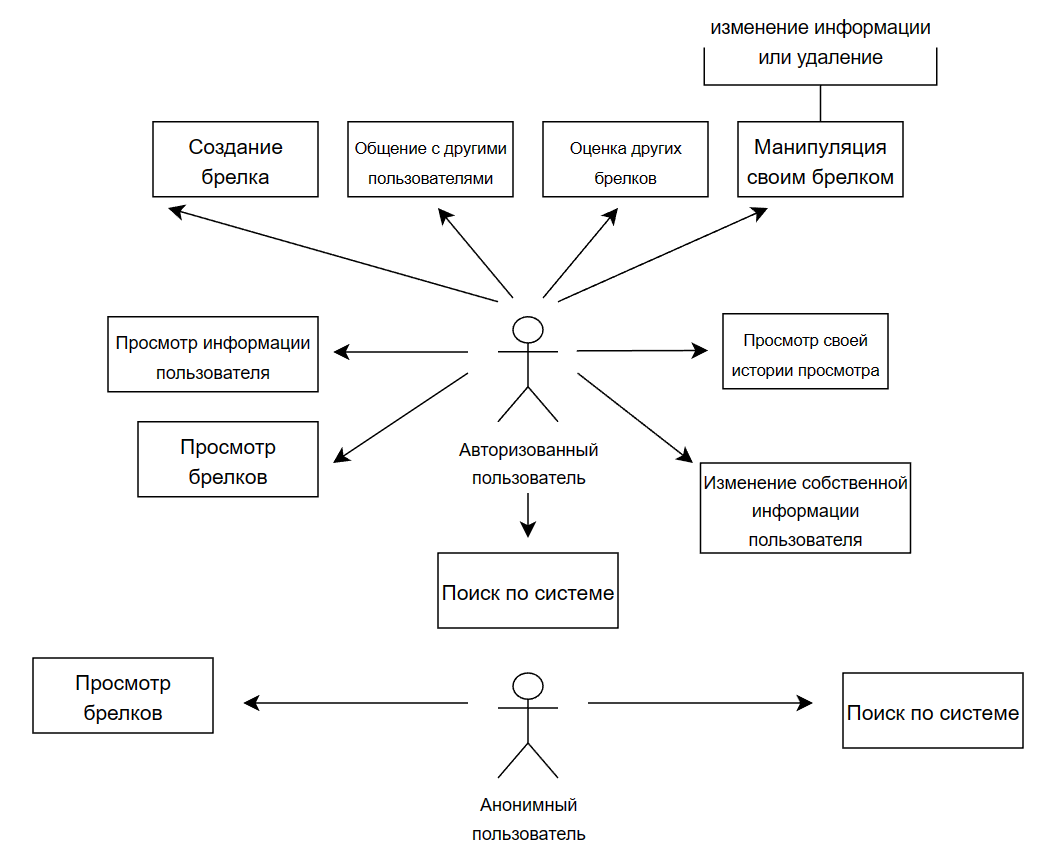


Рисунок 1. Диаграмма функций пользоваталей

Для анонимного пользователя (для того, кто не прошёл авторизацию) доступен просмотр брелков и поиск по системе. Для авторизовавшегося пользователя, помимо этих функций, также доступно создание своего собственного брелка в системе, изменение информации о нём или удаление, просмотр истории просмотров других брелков, коммуникация с другими пользователями, комментирование других брелков, а также изменение информации о своём аккаунте.

Для примера работы системы описан алгоритм отрисовки главной страницы, который представлен на Рисунке 1.

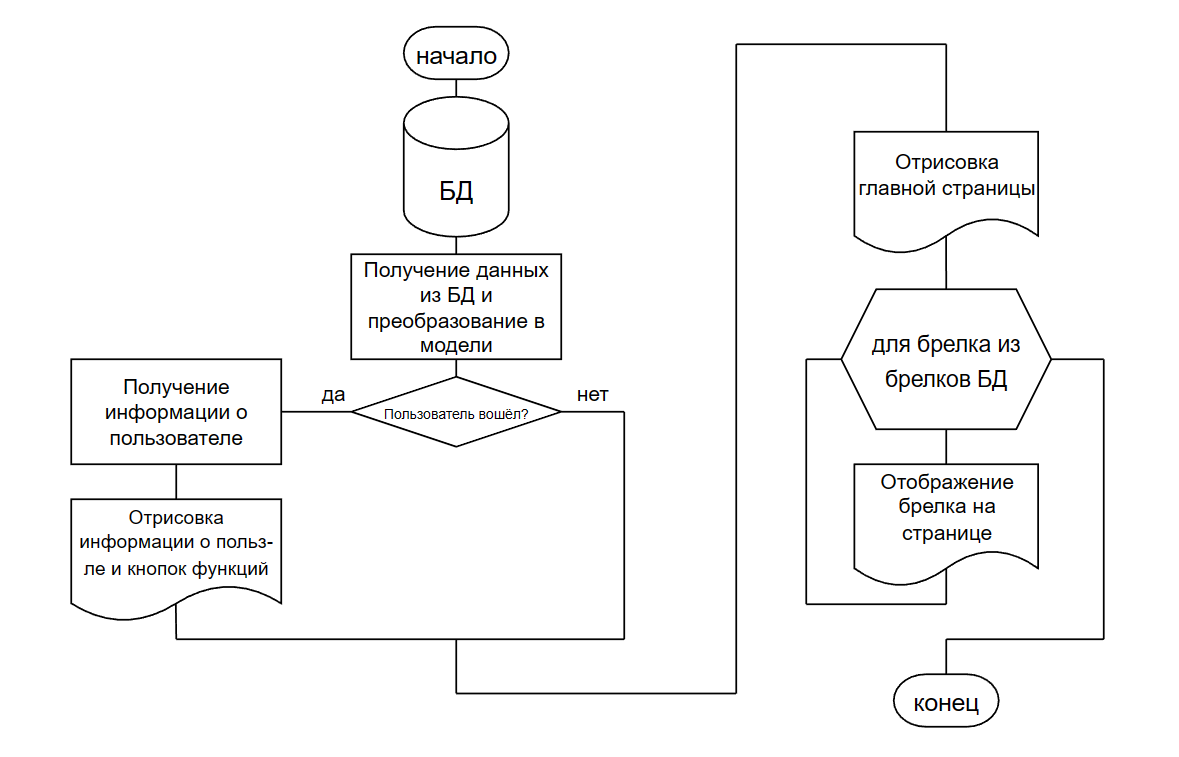


Рисунок 1. Алгоритм отрисовки главной страницы

Данные, которые будут использованы в работе приложения, получаются из базы данных и средствами фреймворка преобразуются в модели. Если пользователь вошёл в систему, то на главной странице также отобразится пользовательское меню. Примеры меню авторизованного пользователя и меню анонимного пользователя представлены на Рисунках 2 и 3.

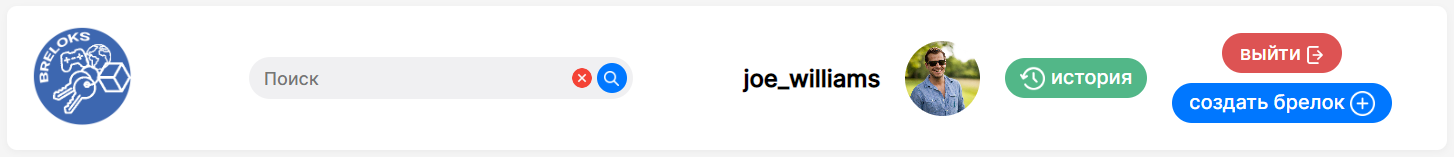


Рисунок 2. Меню авторизованного пользователя

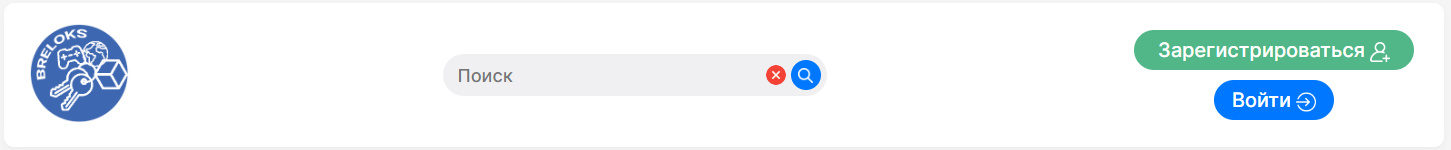


Рисунок 3. Меню анонимного пользователя

В любом случае, на главной странице по циклу отображаются все брелки, которые были созданы в системе, и любой пользователь может просмотреть информацию о них.

База данных реализована при помощи СУБД MySQL. Логическая схема базы данных представлена на Рисунке 4.

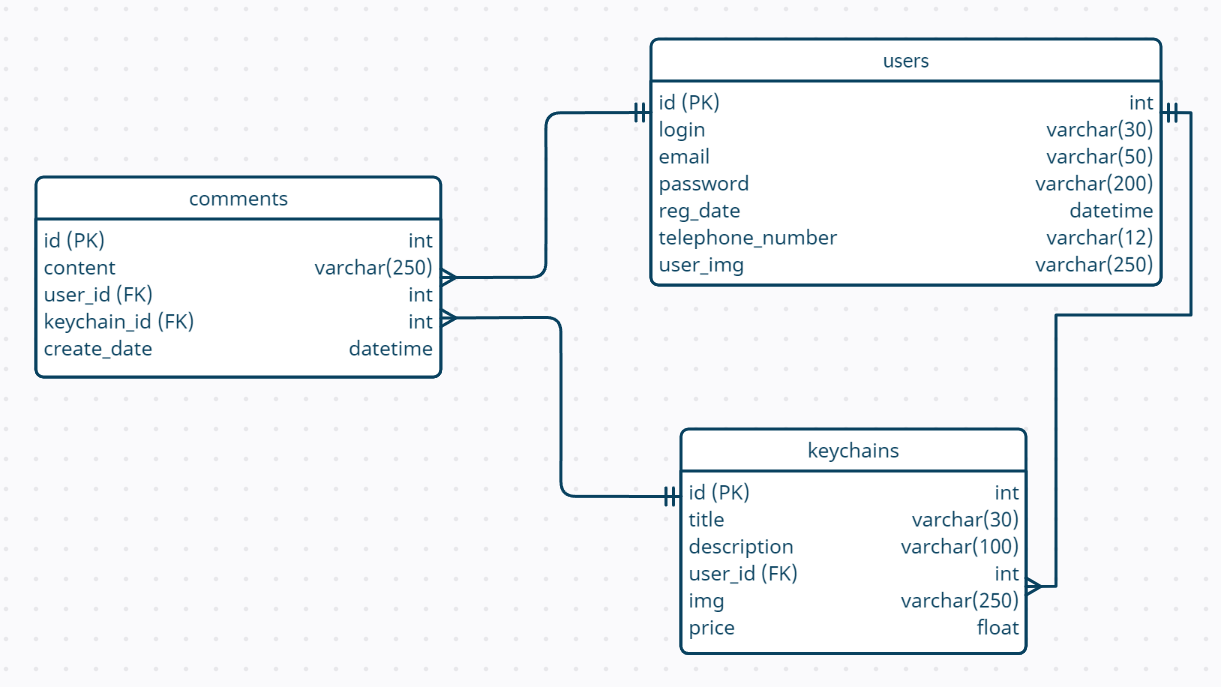


Рисунок 4. Логическая схема базы данных

Таблица users содержит информацию о пользователях (id – первичный ключ), таблица keychains содержит информацию о брелках (id – первичный ключ, user\_id – внешний ключ, пользователь, к которому привязан брелок), таблица comments содержит информацию о комментариях (id – первичный ключ, keychain\_id – внешний ключ, брелок, к которому привязан комментарий, user\_id – внешний ключ, пользователь, который оставил комментарий).

Между таблицами users и comments реализована связь 1:N (один-ко-многим), между comments и keychains – связь 1:N (один-ко-многим), между users и keychains – 1:N (один-ко-многим).

**ИНСТРУКЦИЯ ПОЛЬЗОВАТЕЛЯ**

Пользователю, не осуществившему авторизацию в системе, будет видна страница, представленная на Рисунке 5.

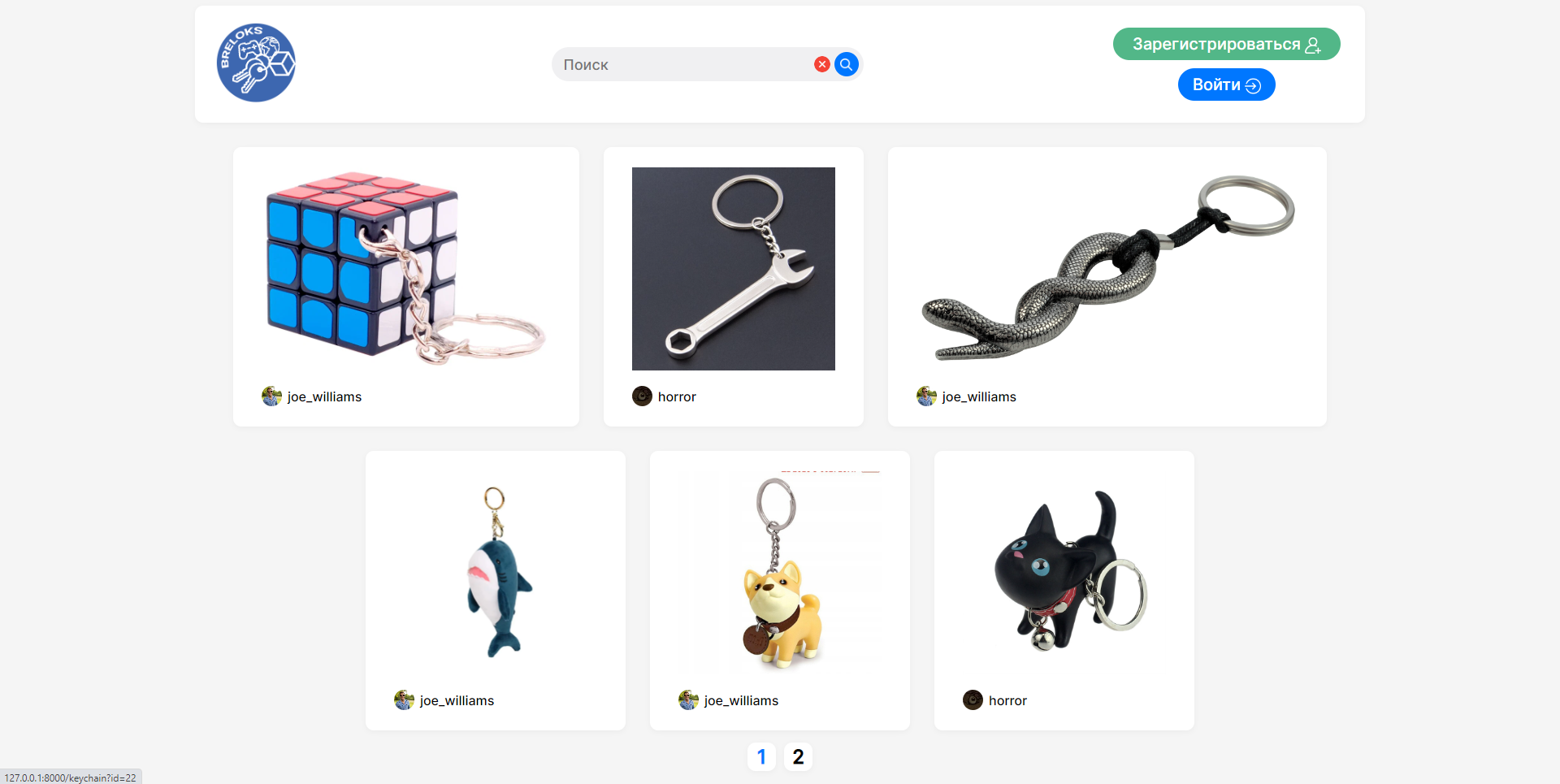


Рисунок 5. Главная страница для анонимного пользователя

Он может зайти на страницы брелков и осуществить поиск по системе, но на этом его функционал заканчивается – нужно авторизоваться для больших возможностей. Пользователь может зарегистрироваться в системе – для этого необходимо нажать на кнопку «Зарегистрироваться». После нажатия кнопки отображается страница регистрации, представленная на Рисунке 6.

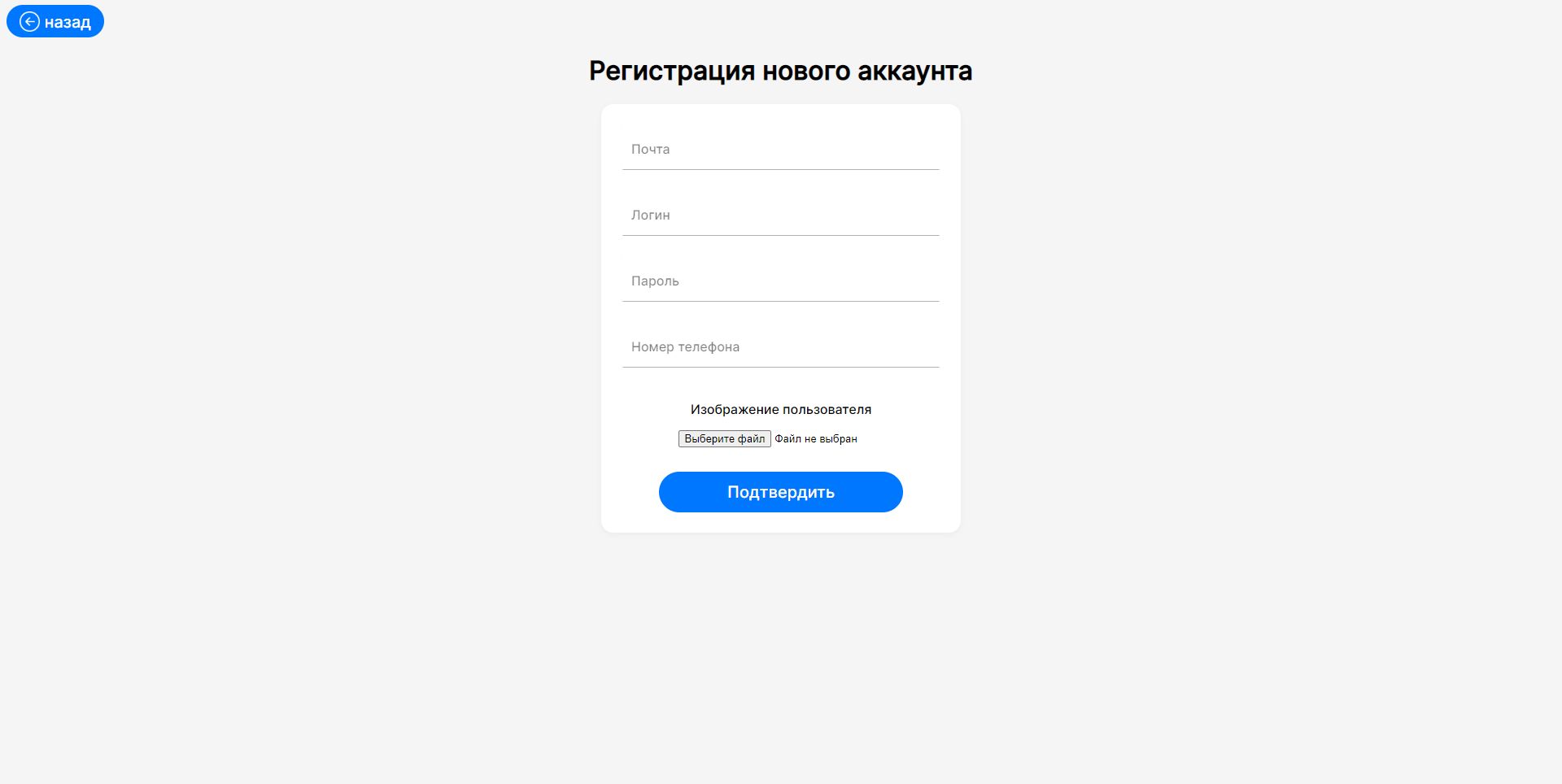


Рисунок 6. Страница регистрации пользователя

Пользователю необходимо ввести свою почту (в правильном формате), логин, пароль и номер телефона (в формате +7…), а также загрузить изображение пользователя. Если в полях ввода введены некорректные данные, или пользователь пытается зарегистрироваться под существующими данными под каждым полем отобразится текст ошибки. Демонстрация представлена на Рисунке 7.

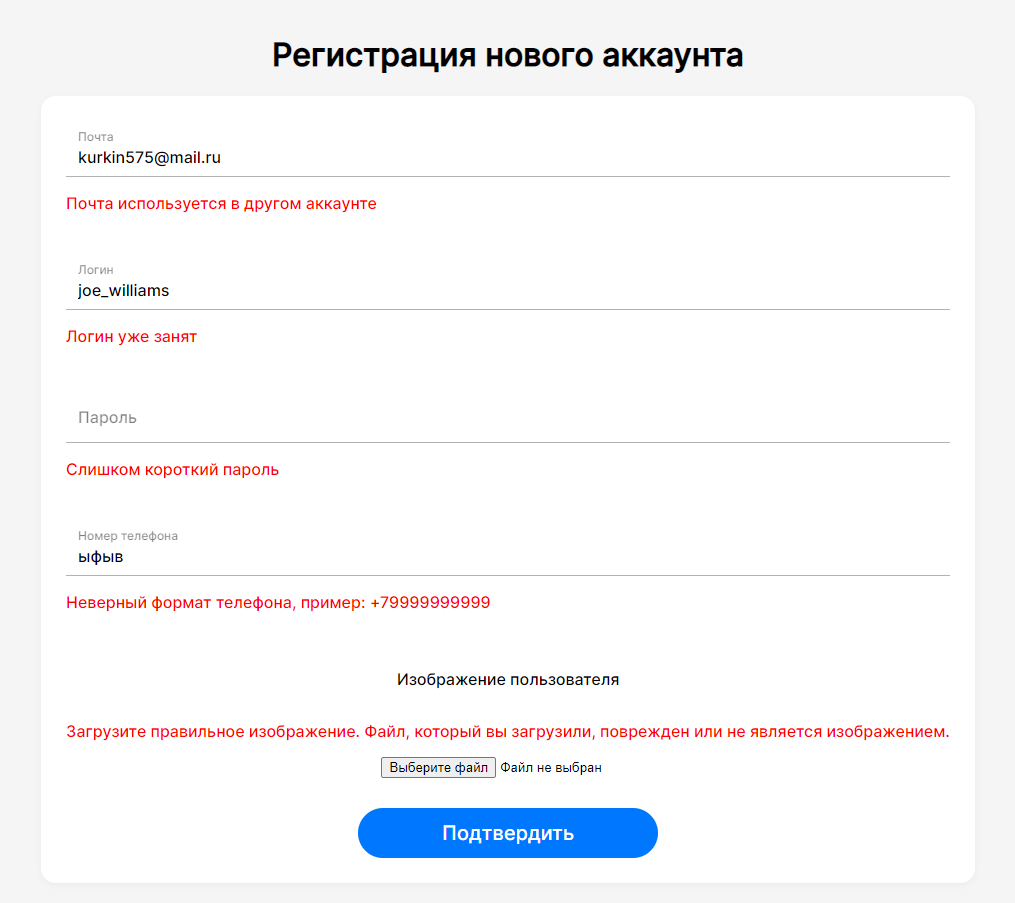
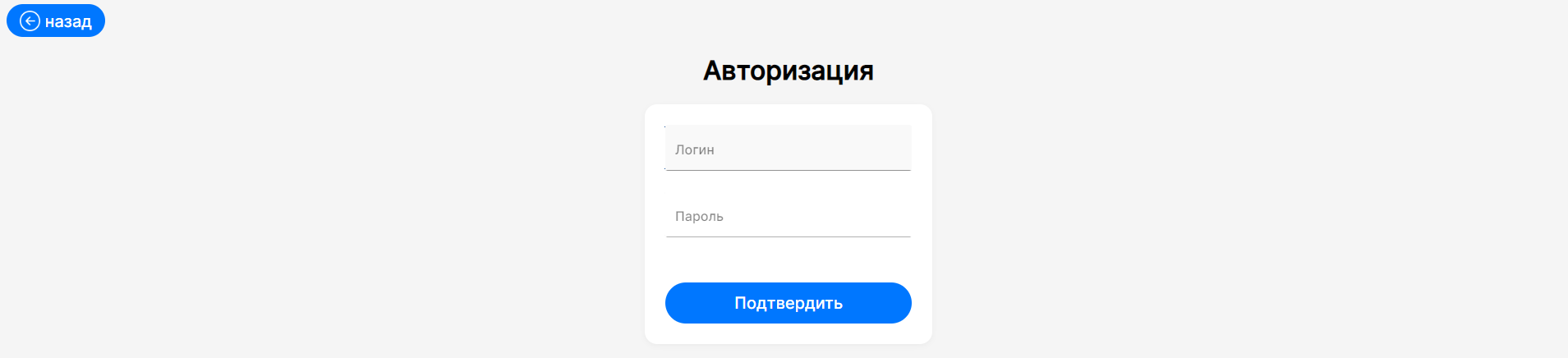


Рисунок 7. Демонстрация ошибок полей ввода

При устранении всех ошибок и вводе корректных значений, пользователь зарегистрируется в системе, и информация о нём занесётся в базу данных. После этого необходимо пройти авторизацию, нажав на кнопку «Войти» на главной странице. Отобразится страница авторизации, представленная на Рисунке 8.



Страница 8. Фрагмент страницы авторизации

Пользователю нужно ввести данные, указанные при регистрации, и он осуществит успешный вход в систему. В противном случае, отобразится ошибка о некорректных данных, фрагмент которой представлен на Рисунке 9.

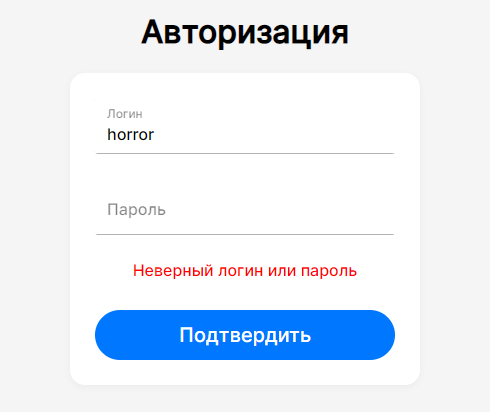


Рисунок 9. Фрагмент ошибки о некорректности введённых данных

Если вход в систему был успешен, то пользователь автоматически перейдёт на главную страницу. На Рисунке 10 представлена главная страница, если пользователь успешно авторизовался в системе.

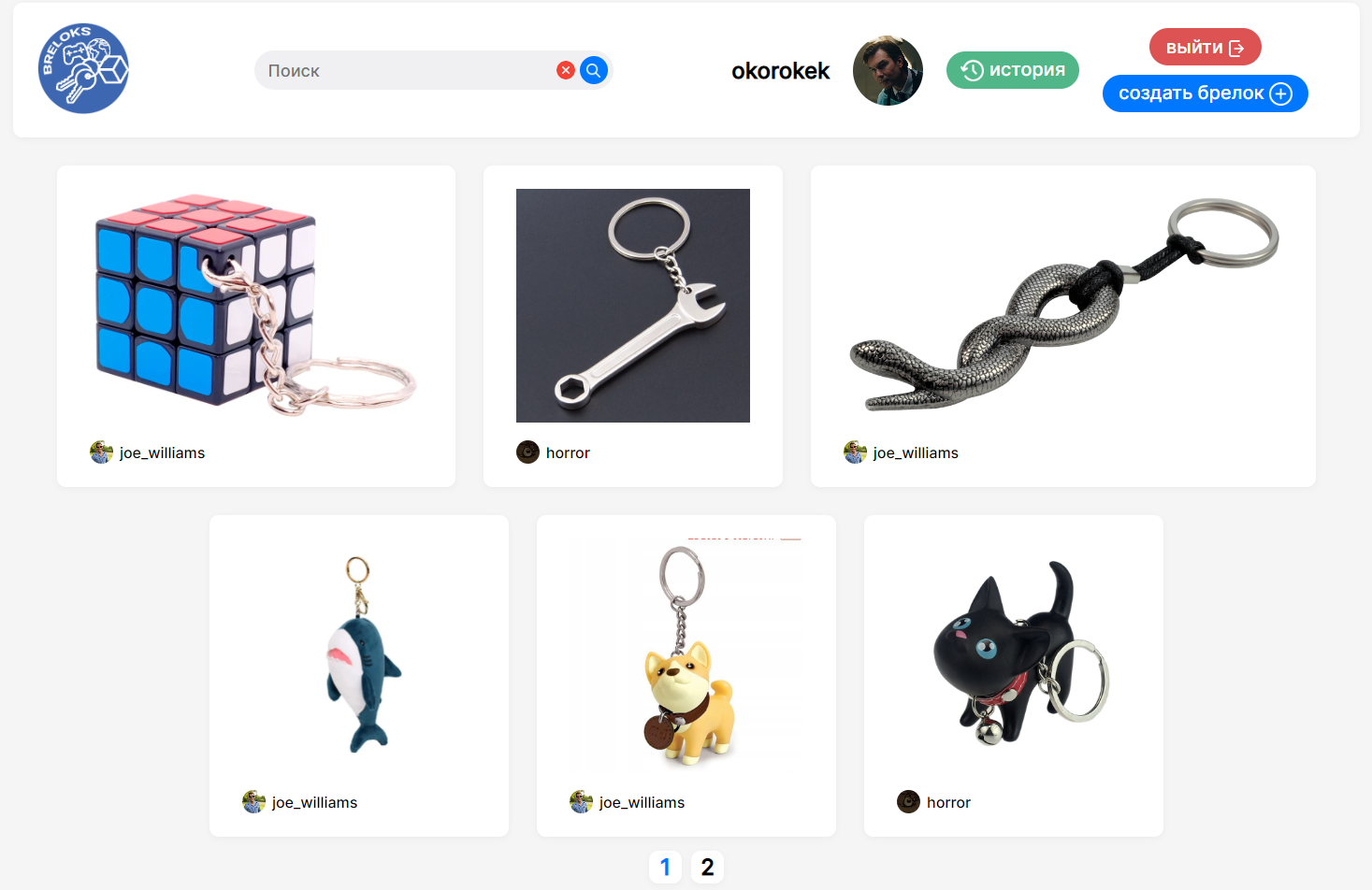


Рисунок 10. Главная страница авторизованного пользователя

Из представленного фрагмента видно, что теперь пользователю доступен больший функционал, например, создание брелка, просмотр истории и т.д. Для просмотра своей истории посещения других брелков, пользователю нужно нажать на кнопку «история». Результатом будет страница, представленная на Рисунке 11.

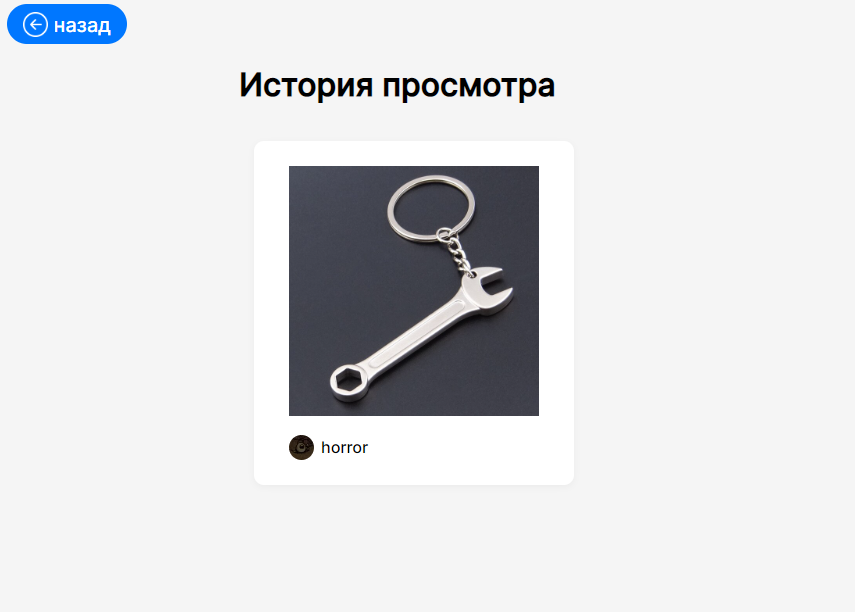


Рисунок 11. Фрагмент страницы истории посещения других брелков

Нажав на брелок, пользователь может перейти на соответствующую страницу с информацией о брелке.

Для создания собственного брелка необходимо нажать на кнопку «Создать брелок». Отобразится страница создания брелка, представленная на Рисунке 12.

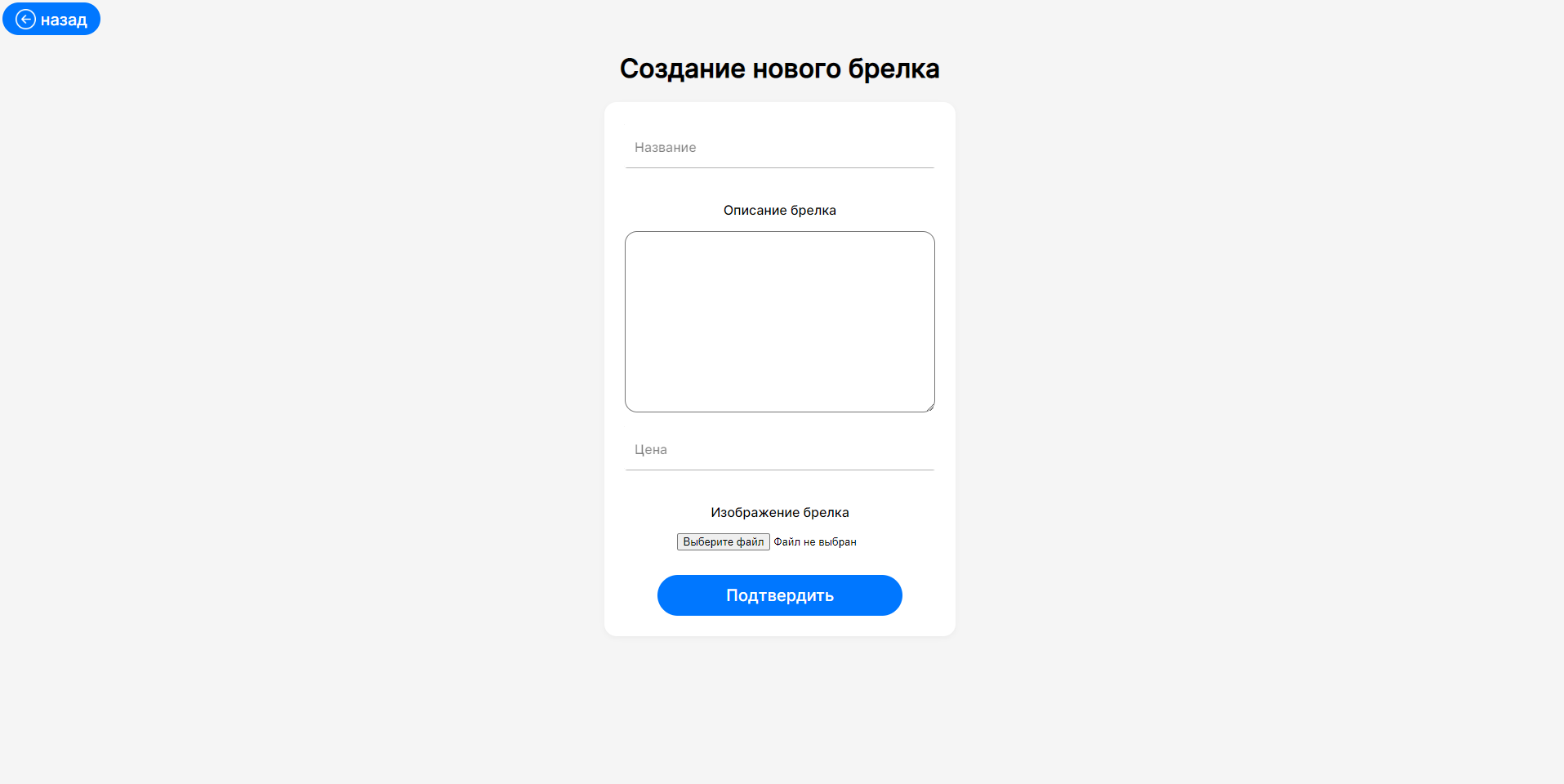


Рисунок 12. Страница создания брелка

Для создания информации о своём брелке в системе пользователю необходимо заполнить ряд представленных на рисунке полей. При попытке ввести некорректные данные, отображаются ошибки под каждым некорректным значением в поле ввода. Фрагмент представлен на Рисунке 13.

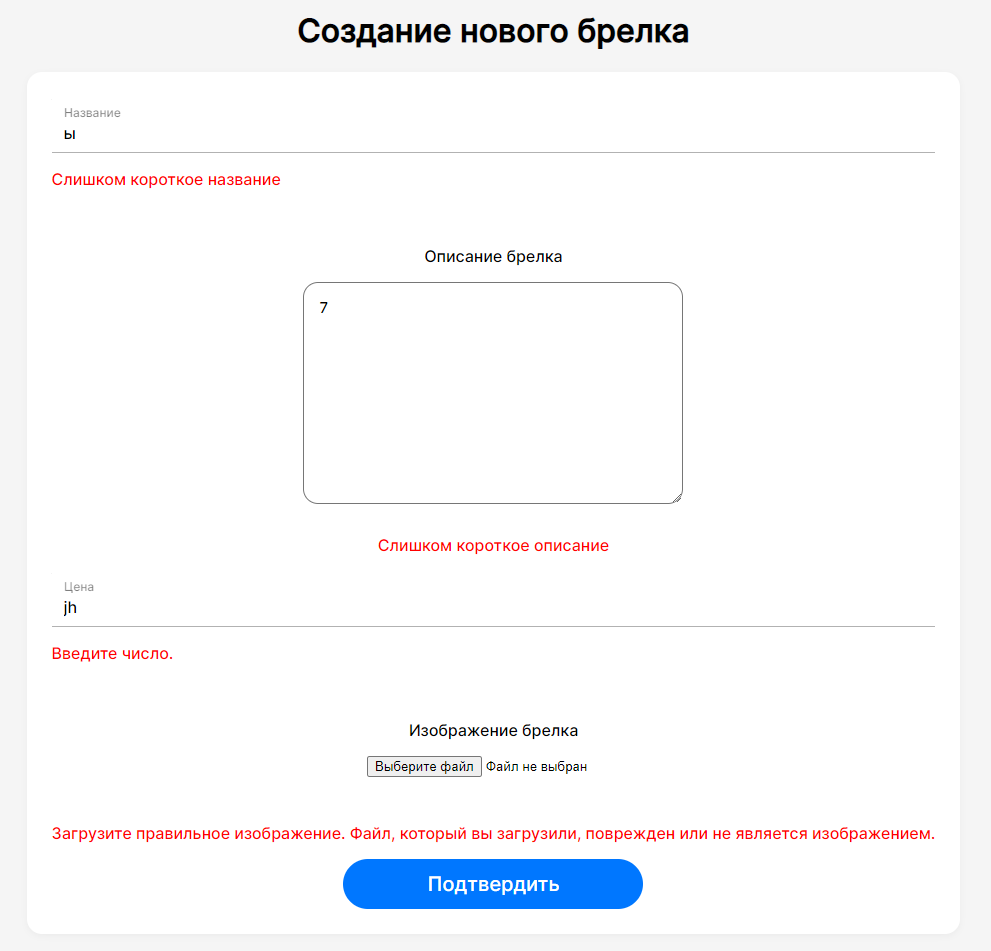


Рисунок 13. Фрагмент некорректных данных

Если данные о брелке были введены корректно, то на главной странице или при помощи инструмента поиска можно найти созданный брелок. Фрагмент представлен на Рисунке 14.

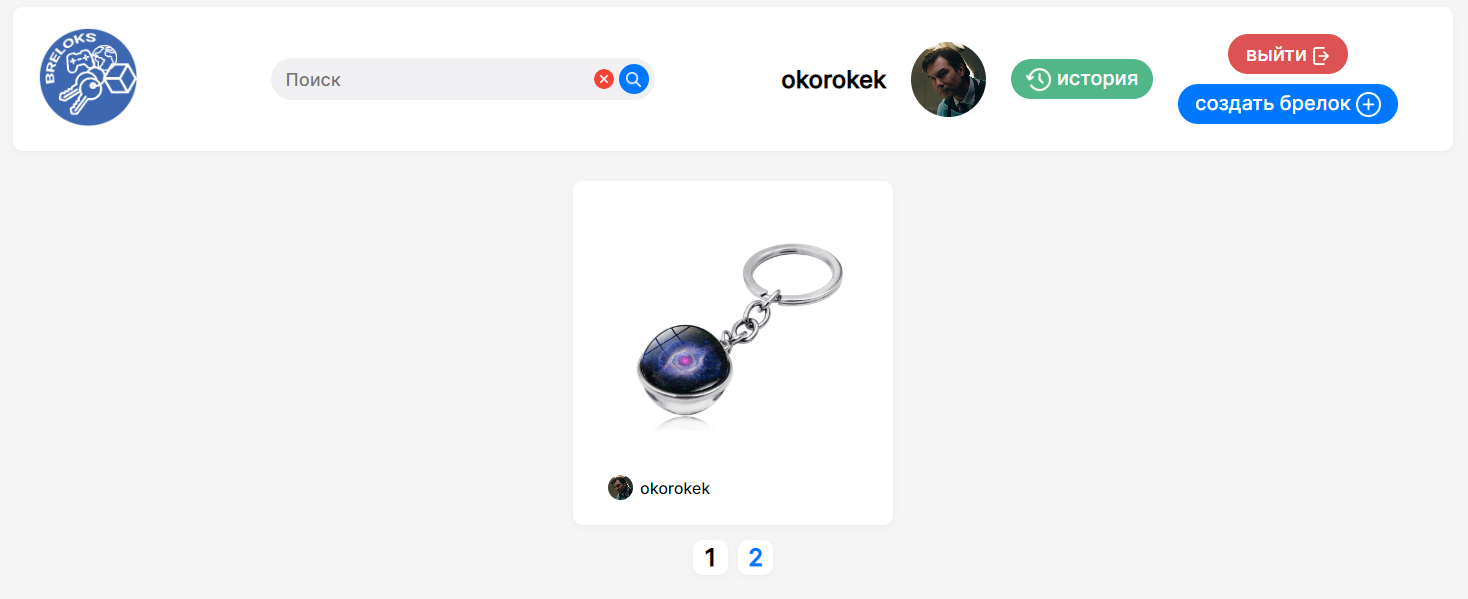


Рисунок 14. Брелок успешно создан

Для осуществления поиска по системе необходимо в поле ввода с надписью «Поиск» ввести поисковой запрос и нажать кнопку с лупой. Тогда отобразится страница с результатами поиска, фрагмент которой представлен на Рисунке 15.

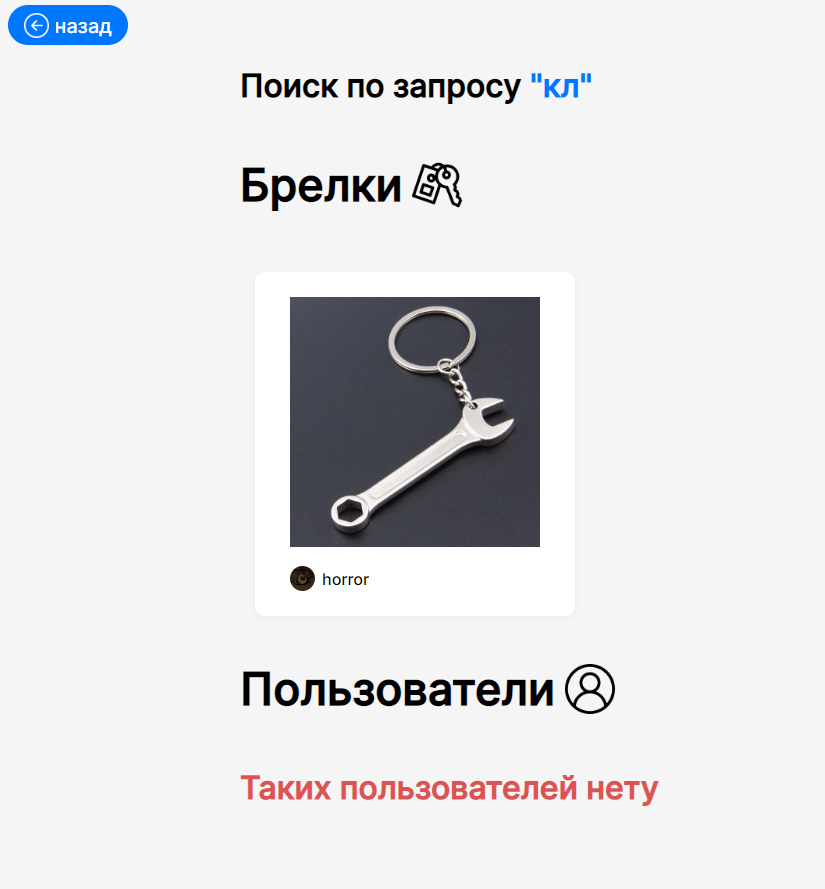


Рисунок 15. Фрагмент страницы с результатами поиска по запросу

Отобразится страница с результатами поиска; по каждому брелку или пользователю можно перейти на его страницу.

Фрагмент страницы с информацией о брелке представлен на Рисунке 16.

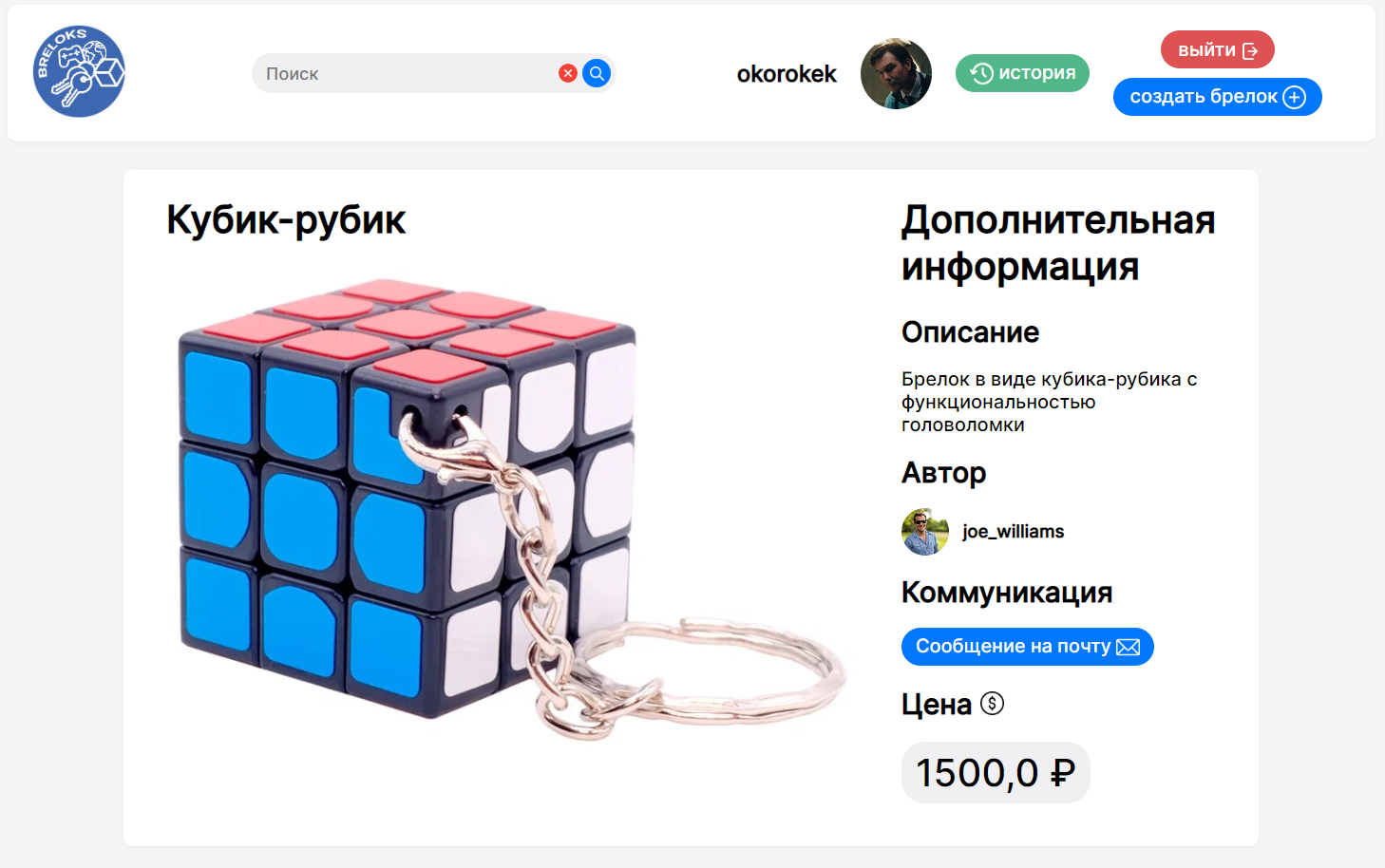


Рисунок 16. Фрагмент страницы с информацией о брелке

На этой странице можно просмотреть информацию о брелке, о пользователе, который его разместил, а также отправить сообщение на почту пользователя. При нажатии кнопки «Сообщение на почту» отобразится страница, фрагмент которой представлен на Рисунке 17.

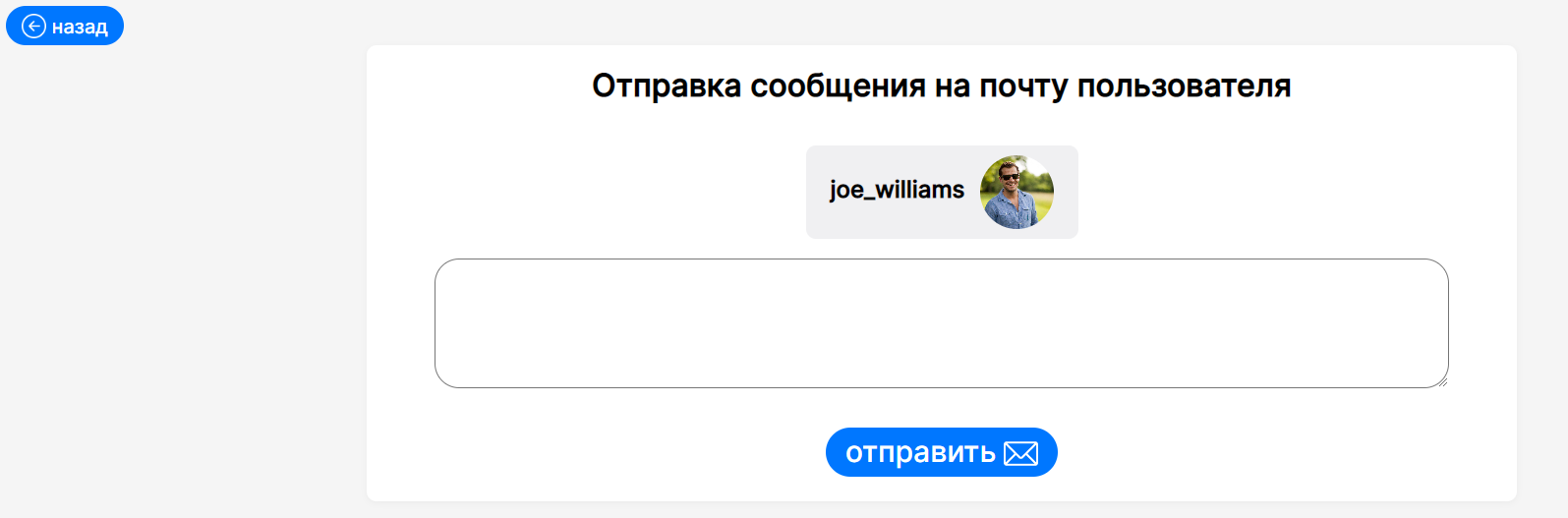


Рисунок 17. Отправка сообщения на почту

Для отправки сообщения необходимо ввести текст сообщения и нажать на кнопку «Отправить». После этого на почту пользователя придёт письмо с введённым текстом. Фрагмент представлен на Рисунке 18.

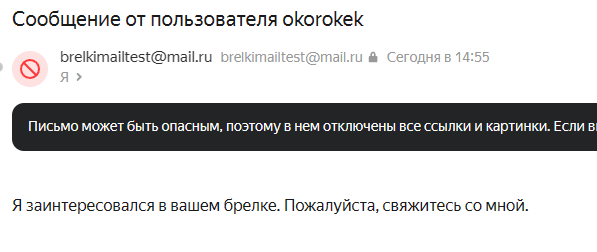


Рисунок 18. Сообщение успешно отправлено

Нижняя часть страницы брелка содержит секцию комментариев. Фрагмент представлен на Рисунке 19.

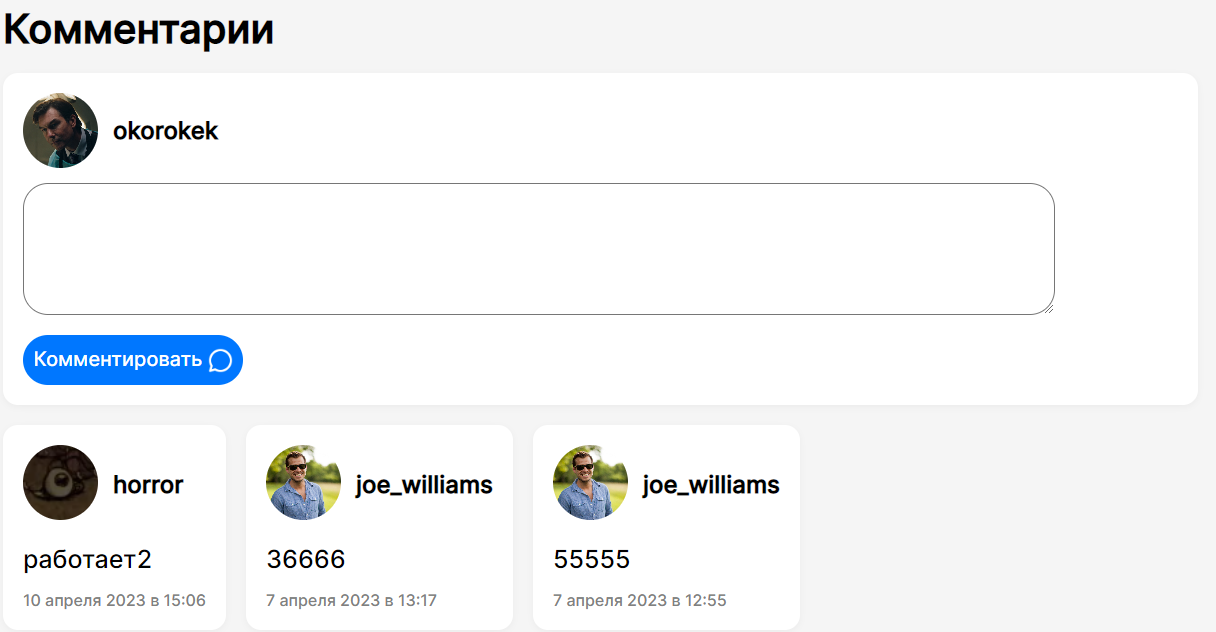


Рисунок 19. Секция комментариев брелка

Для отправки собственного комментария необходимо ввести текст в поле ввода и нажать «Комментировать». Результат представлен на Рисунке 20.

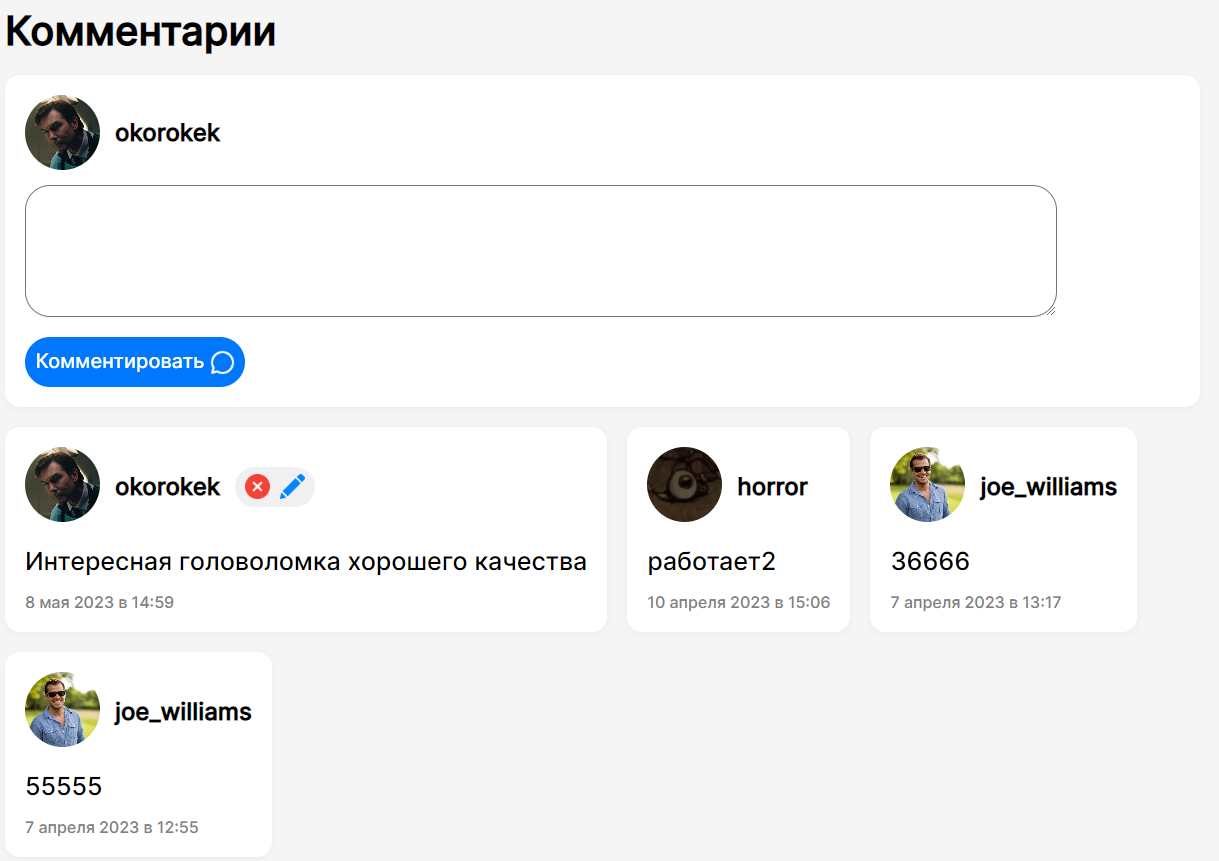


Рисунок 20. Комментарий отправлен

Свои комментарии пользователь может удалять или изменять. Для удаления необходимо нажать на иконку красного крестика, а для редактирования – синего карандаша. Фрагмент страницы редактирования комментария представлен на Рисунке 21.

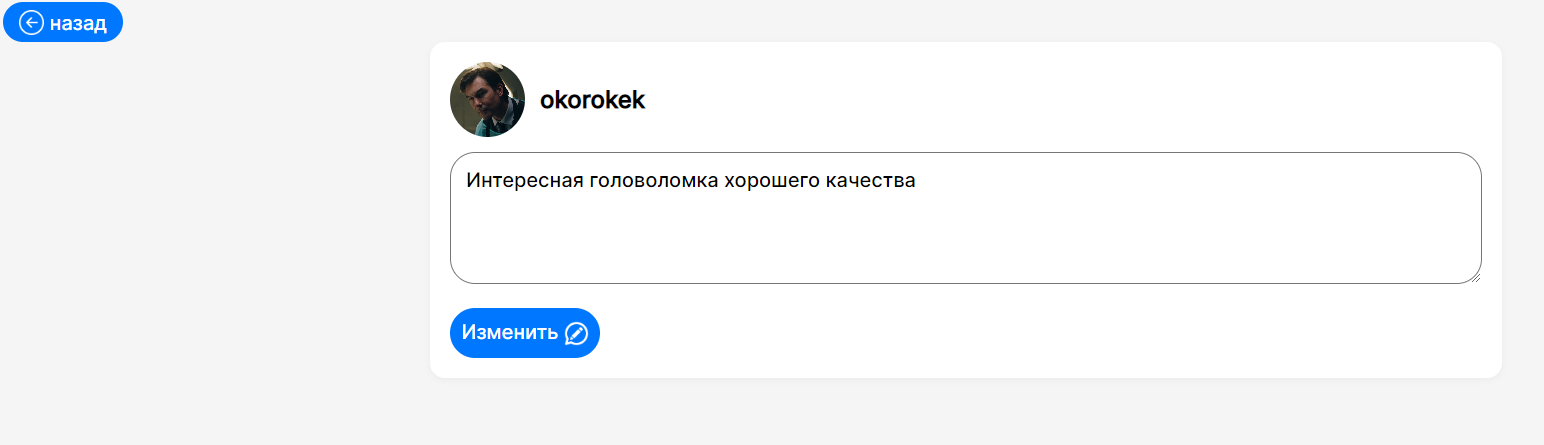


Рисунок 21. Редактирование комментария

Для изменения комментария нужно скорректировать его исходный текст и нажать кнопку «Изменить». После нажатия на кнопку комментарий изменится, и пользователь перейдёт обратно на страницу с брелком.

Для неавторизованного пользователя не будет возможности оставить комментарий, что видно из Рисунка 22.

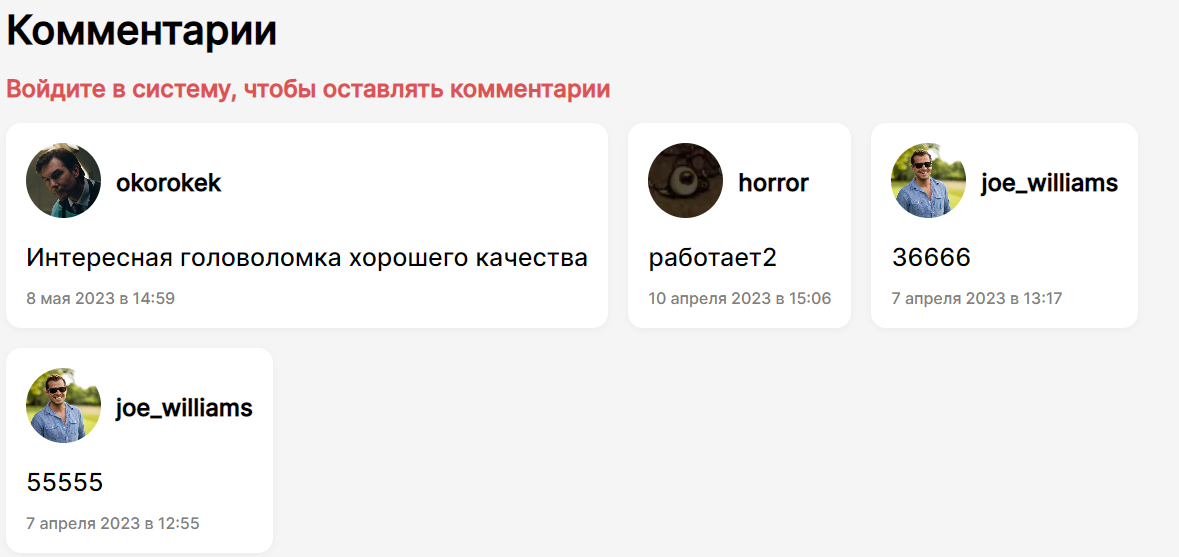


Рисунок 22. Для комментирования нужно войти в систему

Для просмотра информации о пользователе нужно перейти на его личную страницу, нажав на его изображение, например, из страницы поиска. Фрагмент личной страницы пользователя представлен на Рисунке 23.

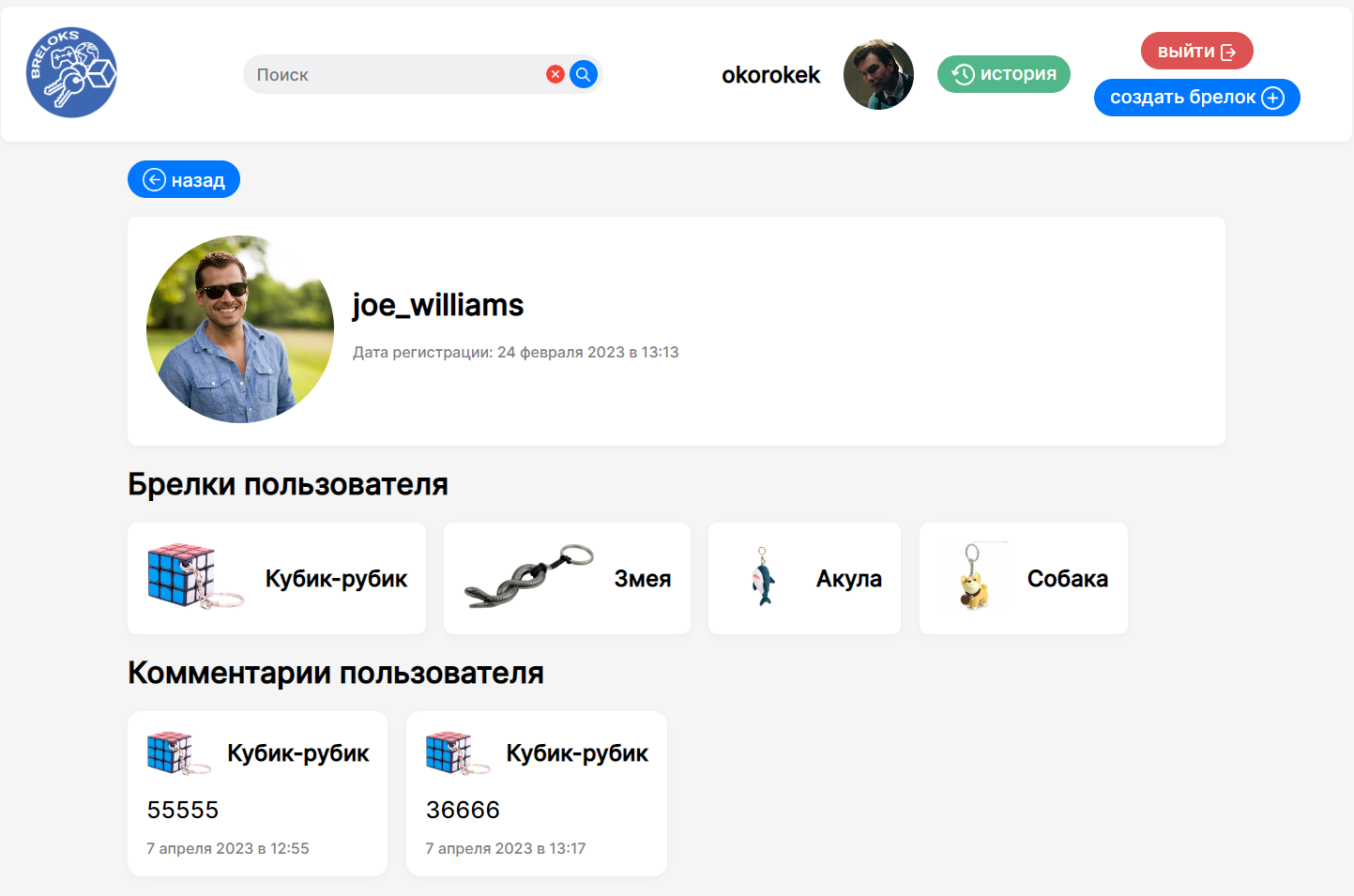


Рисунок 23. Фрагмент личной страницы пользователя

На этой странице можно просмотреть брелки пользователя, а также комментарии, который он оставил. Если же авторизовавшийся пользователь зашёл на свою страницу, то появится дополнительный функционал. Фрагмент представлен на Рисунке 24.

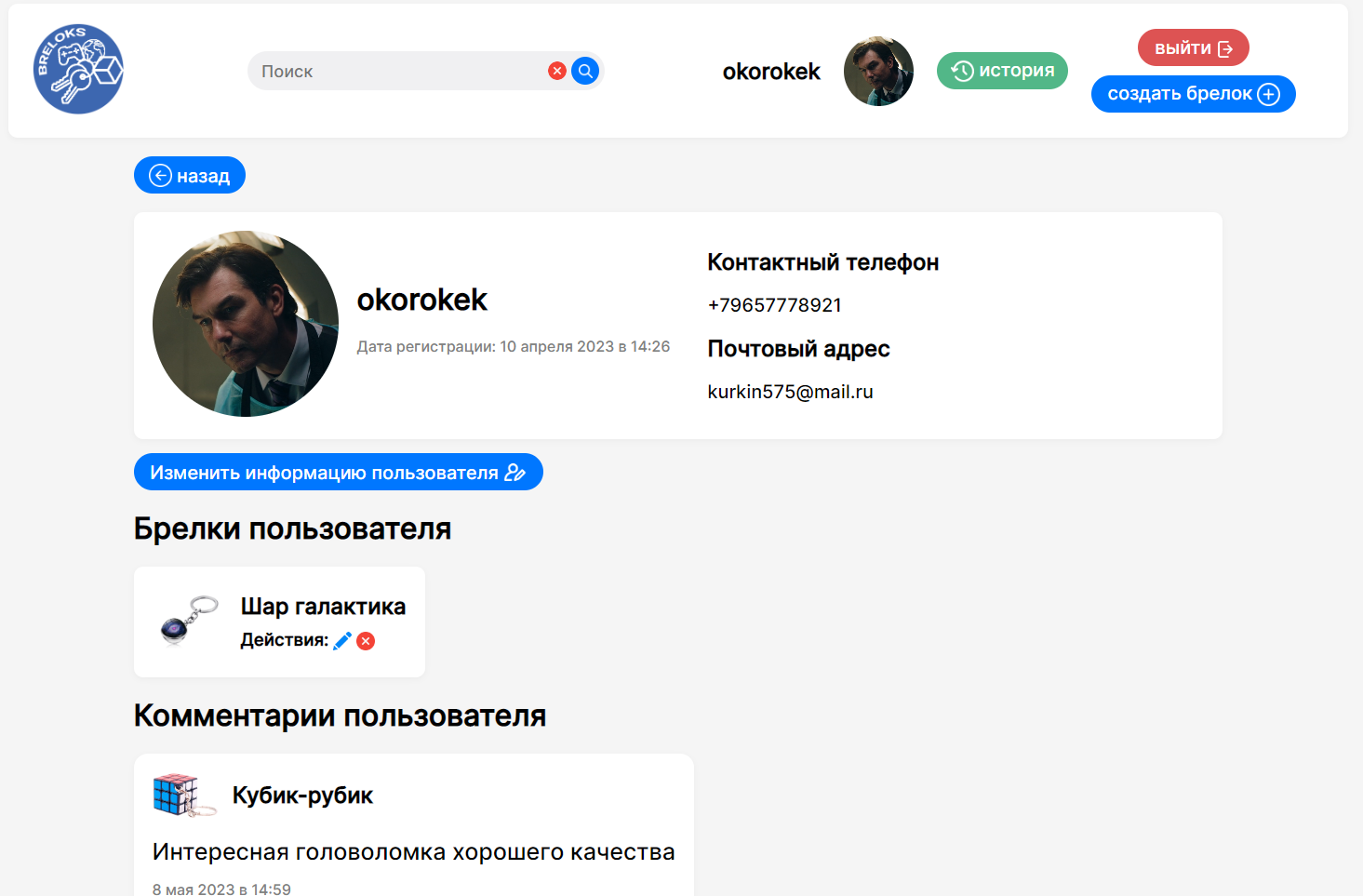


Рисунок 24. Пользователь зашёл на свою страницу

Появилась информация о конфиденциальных данных, а также возможность изменить информацию о своих брелках или удалить их. Для удаления нужно нажать на кнопку с красным крестиком, а для изменения – синий карандаш. После нажатия на кнопку редактирования, отобразится страница, аналогичная странице изменения комментария, но с соответствующими для брелка полями. После редактирования информации, информация о брелке изменится и в базе данных. Для изменения информации о пользователе, нужно нажать на соответствующую кнопку. Отобразится страница, фрагмент которой представлен на Рисунке 25.

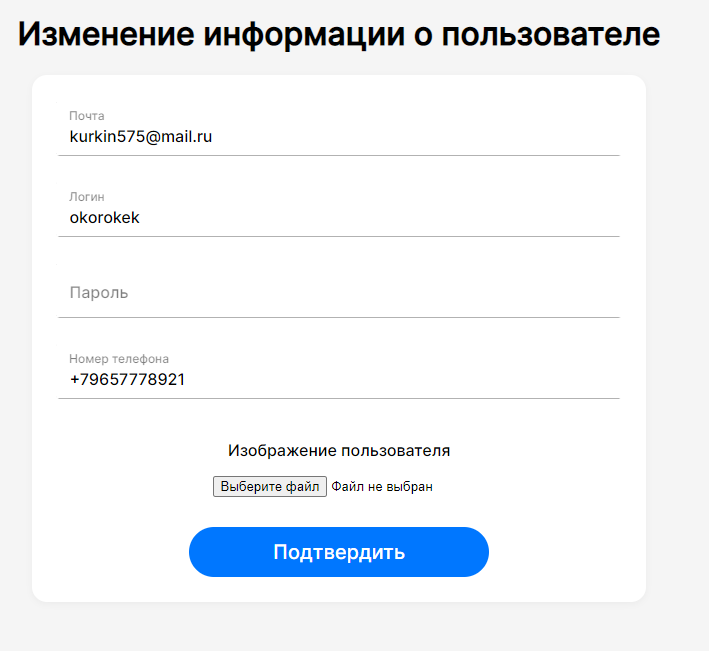


Рисунок 25. Изменение информации о пользователе

Если не изменять какую-либо информацию (например, почту), то она останется изначальной. При попытке задать почту или логин другого пользователя, отобразится ошибка, представленная на Рисунке 26.

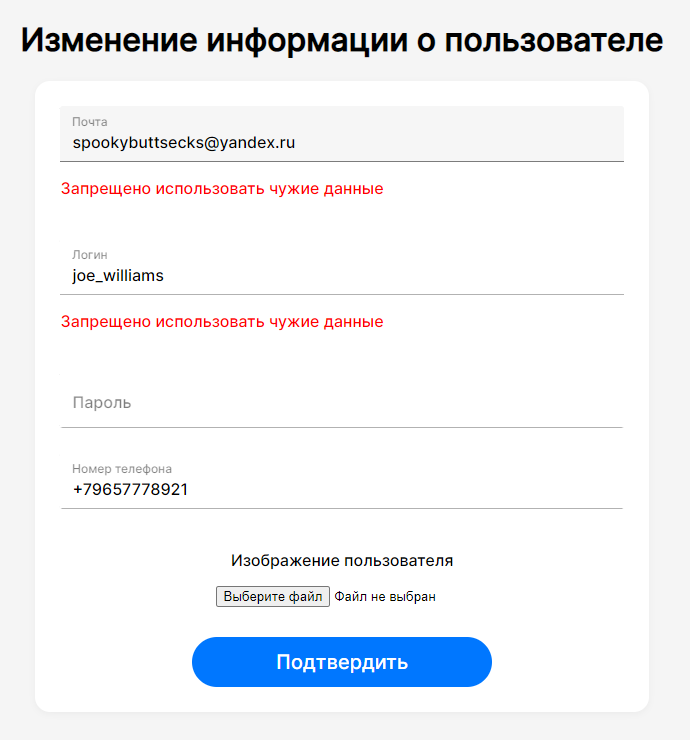


Рисунок 26. Попытка использовать чужие данные

При вводе корректных данных и нажатии на кнопку «Подтвердить», введённая информация изменится.

Для выхода из системы необходимо нажать на кнопку «Выйти» в меню пользователя. После нажатия отобразится главная страница анонимного пользователя.

**ВЫВОДЫ**

В результате проектирования и программирования удалось реализовать приложение, которые решает следующие задачи:

* Отображение брелков на главной странице;
* Поиск по брелкам и пользователям;
* Авторизация и регистрация пользователя;
* Просмотр истории посещения страниц брелков;
* Просмотр информации о брелке на его странице;
* Создание, изменение или удаление брелка и его данных;
* Комментирование брелков;
* Просмотр личной страницы пользователя, где он может изменить свои брелки или личную информацию;
* Коммуникация с пользователем посредством отправки сообщения на электронную почту.

На данном этапе не удалось реализовать следующий функционал:

* Просмотр 3D модели брелка;
* Коммуникация с пользователем посредством внутреннего чата.

Нереализованный функционал планируется к добавлению на этапе модернизации системы.

**ЛИЧНЫЙ ВЫВОД**

В ходе разработки проекта я познакомился с функциональным и удобным для работы фреймворком на языке Python – Django. Он понравился мне наличием понятной структуры (шаблоны, вид-функции, формы), средствами моделирования объектов базы данных (Django ORM), а также наличием ясной для понимания документации с примерами и подробным описанием методов и возможностей. Программирование с использованием данного фреймворка, безусловно, дало мне большой практический опыт.