ГОСУДАРСТВЕННОЕ УЧРЕЖДЕНИЕ ОБРАЗОВАНИЯ  
«ИНСТИТУТ ПОДГОТОВКИ НАУЧНЫХ КАДРОВ  
НАЦИОНАЛЬНОЙ АКАДЕМИИ НАУК БЕЛАРУСИ»

Кафедра естественнонаучных дисциплин и информационных технологий

**ВЫПУСКНАЯ РАБОТА**

по дисциплине

«ОСНОВЫ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ»

на тему:

«Информационные технологии в автоматизированных системах

управления»

Студент магистратуры  
(соискатель) \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(Фамилия, имя, отчество)

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**(название учреждения, в котором студент магистратуры (соискатель) выполняет свое диссертационное исследование)**

Руководитель выпускной

работы от кафедры \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(Ф.И.О)

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(подпись)

Рецензент \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(Ф.И.О)

Минск

2019

ОГЛАВЛЕНИЕ

[ВВЕДЕНИЕ 3](#_Toc28390301)

[ГЛАВА 1 ОБЗОР ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ В АВТОМАТИЗИРОВАННЫХ СИСТЕМАХ УПРАВЛЕНИЯ 4](#_Toc28390302)

[1.1 Современные информационные технологии в автоматизированных системах управления 4](#_Toc28390303)

[1.2 Техническое обеспечение информационных технологий в автоматизированных системах управления 4](#_Toc28390304)

[1.3 Программное обеспечение информационных технологий в автоматизированных системах управления 5](#_Toc28390305)

[1.4 Информационное обеспечение информационных технологий в автоматизированных системах управления 5](#_Toc28390306)

[ГЛАВА 2 АНАЛИТИЧЕСКАЯ ЧАСТЬ 6](#_Toc28390307)

[2.1 Ajax Systems 6](#_Toc28390308)

[2.2 BroadLink 10](#_Toc28390309)

[2.3 Xiaomi 12](#_Toc28390310)

[2.4 Fibaro 14](#_Toc28390311)

[2.5 Сравнительный анализ 17](#_Toc28390312)

[ГЛАВА 3 ПРАКТИЧЕСКАЯ ЧАСТЬ 18](#_Toc28390313)

[3.1 Объектно-ориентированный анализ и проектирование системы 18](#_Toc28390314)

[3.2 Проектирование модели 19](#_Toc28390315)

[3.3 Вычислительная система 25](#_Toc28390316)

[3.4 Концептуальный прототип 27](#_Toc28390317)

[3.5 Организация данных 29](#_Toc28390318)

[3.6 Функции: логическая и физическая организация 32](#_Toc28390319)

[3.7 Описание программного средства 34](#_Toc28390320)

[3.7 Функциональное тестирование 35](#_Toc28390321)

[ЗАКЛЮЧЕНИЕ 41](#_Toc28390322)

[СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ 42](#_Toc28390323)

ВВЕДЕНИЕ

Быстрое развитие информационных технологий и различных способов передачи данных приводят к появлению все новых и новых систем автоматизации (в том числе и систем домашней автоматизации). На сегодняшний день существует множество таких систем, которые решают многие повседневные задачи пользователей. Однако, несмотря на неоспоримые преимущества, такие системы не стали настолько популярными как, например, смартфоны, пользовательские компьютеры и другие устройства, без которых сложно представить жизнь современного человека.

Одной из самых популярных и востребованных систем автоматизации является технология Интернет вещей. Интернет вещей – это технология, которая объединяет объекты автоматизации в единую сеть и предоставляет возможность собирать статистику, получать информацию по состоянию данных объектов, а также позволяет дистанционно, через приложения, управлять подключенными к системе устройствам сети.

Данный подход подразумевает минимальное взаимодействие с такими устройствами – настройка, конфигурация сценариев, давать определенного вида инструкции и многое другое.

Интернет вещей - системы используют различные способы подключения устройств – это может быть как WiFi, Bluetooth, популярный сегодня ZigBee и другие спецификации беспроводной сети.

В данной работе будут рассматриваться существующие системы на примере систем домашней автоматизации. Будут рассматриваться преимущества, недостатки каждой системы.

ГЛАВА 1 ОБЗОР ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ В АВТОМАТИЗИРОВАННЫХ СИСТЕМАХ УПРАВЛЕНИЯ

* 1. Современные информационные технологии в автоматизированных системах управления

**1.1.1**. Применение информационных технологий в автоматизации процессов [1].

**1.1.2** Модели на основе искусственных нейронных сетей для построения поведения устройств в автоматизированных системах управления [2].

1.1.3 Информационные технологии в повышении точности измерений [3].

**1.1.4** Аппаратное резервирование в системах промышленной автоматизации [4].

**1.1.5** Информационные технологии в исследовании надежности автоматизированных систем управления [5].

**1.1.6** Применение информационных технологий в оптимизации процессов координации и регулирования [6].

* 1. Техническое обеспечение информационных технологий в автоматизированных системах управления

**1.2.1** Средство сбора информации – датчик температуры BME280 [7, 8].

**1.2.2** Исполнительное устройство – сервопривод SG90 [7, 9].

1.2.3 Программируемый логический контроллер ATmega2560 [7, 10].

**1.2.4** Сетевой адаптер Ethernet Shield W5100 [7, 11].

**1.2.5** Программатор CH341A [7, 10].

**1.2.6** Устройство ввода/вывода – LCD1602 LCD Keypad Shield [7, 12].

* 1. Программное обеспечение информационных технологий в автоматизированных системах управления

**1.2.1** Программа для хранения, обработки и регистрации объектов автоматизации Workbench [13].

**1.2.2** Программа для моделирования расположения устройств автоматизации Planner5d [14].

1.2.3 Программа для анализа статистических данных Microsoft Excel [15].

**1.2.4** Программа для запуска веб приложений Apache Tomcat [16].

**1.2.5** Программа для прошивки объектов автоматизации Arduino IDE [17].

* 1. Информационное обеспечение информационных технологий в автоматизированных системах управления

**1.2.1** Учебное пособие «Автоматизация технологических процессов и производств» (режим доступа: http://window.edu.ru/catalog/pdf2txt/030/66030/37472).

1.2.3 Автоматизация технологических процессов на производстве (режим доступа: https://ap-n.com/).

**1.2.4** Журнал Волгоградского технического университета (режим доступа: https://elibrary.ru).

**1.2.5** Энциклопедия АСУ ТП (режим доступа: https://www.bookasutp.ru).

**1.2.6** Сборник технических статей (режим доступа: http://www.teh-lib.ru).

ГЛАВА 2 АНАЛИТИЧЕСКАЯ ЧАСТЬ

Автоматизированные системы (или Многофункциональные системы) «Умный дом» набирают все большую популярность в жизни современного человека. Это связано с несколькими факторами:

1. С каждым годом возрастает техническая грамотность населения планеты (в связи с развитием технологической области во всем мире);
2. Данные системы со временем теряют в цене, становясь все более и более доступными для среднестатистического потребителя.

Такие системы сегодня устанавливают на объекты различных типов – многоквартирные дома, частные дома, коммерческие здания, здания государственного назначения и многие другие.

Аналитическая часть будет проводиться на наиболее популярных системах домашней автоматизации – AJAX, BroadLink, Fibaro и, кончено же, Xiaomi.

Сперва рассмотрим каждую систему по-отдельности, а затем сравним их в отношении между собой. На основании полученной информации сформируем критерии, по которым будем формировать (проектировать, разрабатывать) свою систему.

**2.1 Ajax Systems**

Ajax Systems – молодая украинская компания, основанная в 2011 году [18]. Основную часть их продукции занимают автоматизированные системы безопасности. Данные системы позволяют защититься от различного рода угроз от воздействий внешних факторов – ограблений, пожара, затопления. Также система Ajax позволяет управлять электропитанием, что позволяет рассматривать данную систему как систему «Умного дома» [18].

Так как производитель находится в Украине, то данная система поддерживает как украинский, так и русский язык, что является большим преимуществом для среднестатистического русскоговорящего пользователя перед другими зарубежными аналогами.

Все устройства системы Ajax Systems работают на зашифрованной двусторонней связи Jeweller [19]. Эта технология является собственной разработкой компании Ajax, что является одновременно как преимуществом, так и недостатком, так как невозможно подключить устройства от других производителей, работающих на беспроводной радиосвязи другого типа [19].

Центральный элемент системы – хаб, оснащен специальным устройством бесперебойной электросети, для возможности предоставления резервного питания в случае выключения всего электропитания (что вполне вероятно при попытке взлома жилища). В случае, если такое произойдет, то хаб способен автономно работать до 16 часов [20].

Данная система имеет минималистичный и презентабельный вид, что очень важно для конечного пользователя двадцать первого века. Все типы устройств системы Ajax представлены на рисунке 2.1.



**Рисунок 2.1 - Устройства системы Ajax Systems**

По словам производителя, дальность действия датчиков достигает двух тысяч метров на открытом пространстве. В условиях частного дома или квартиры, где сигналу будут препятствовать межкомнатные стены, данный показатель снижается. Однако, несмотря на это, данная система имеет огромный отрыв относительно других, более дешевых систем китайских производителей, где дальность связи составляет не более 10 метров (из-за особенности сети ZigBee) [21].

Как и в большинстве современных систем автоматизации, система от компании Ajax Systems, работает от батареек, а не от общей электросети (за исключением центрального элемента системы – хаба). Новый, собственный протокол связи позволяет наиболее энергоэффективно расходовать ресурс батареи, что позволяет устройствам автономно работать до семи лет. Кроме того, в случае если какое-либо из устройства практически полностью израсходовало ресурс батареи, приложение напомнит о необходимости замены соответствующим нотификационным сообщением [21].

Для управления всей системой от компании Ajax Systems, разработано специальное приложение Ajax Security System, в котором можно управлять каждым элементом автоматизации, следить за показателями датчиков, включать или отключать элементы системы [21].

В данной системе реализовано разделение на роли – администратор / пользователь. Администратор всегда может удалить или добавить нового пользователя для доступа к элементам системы и безопасности жилища [21].

К недостаткам данной системы можно отнести отсутствие камер видеонаблюдения, однако это проблема решается подключения камер сторонних производителей, предлагаемых, непосредственно, Ajax Systems.

В случае если система определит какие-то неполадки, пользователю сразу же (по словам производителя, за миллисекунды – 0.15 секунды) придет push - уведомление на телефон. Данная функция настраивается – это может быть как sms - уведомление, так и звонок на телефон.

Предусмотрена возможность взлома с использованием устройств, заглушающих определенные частоты. Система работает на двух типах передачи связи – сотовая связь и широкополосный интернет. В случае, если будет заглушена одна из них, хаб автоматически перейдет на другой тип передачи данных [21].

Одной из особенностей данной системы является то, что алгоритмы распознавания не дают сработать датчикам на небольшие объекты (ниже пятидесяти сантиметров и двадцати килограмм веса). Эта особенность позволяет содержать в доме домашних животных без вреда для работы системы.

Комплектация (базовый набор):

1. Интеллектуальная централь второго поколения;
2. Беспроводный датчик движения;
3. Беспроводный датчик открытия;
4. Беспроводный брелок [22].

Подводя итоги, можно сказать, что данная система решает две наиболее востребованные задачи пользователя:

1. Дает гарантию полной безопасности вашего имущества, так как данная система контролирует границы вашего дома на предмет взлома (и, как следствие, возможной кражи). Также система предусматривает защиту от газовой, электрической, пожарной и других угроз вашему имуществу;
2. Предоставляет комфортабельные и удобные функции автоматизации систем жизнеобеспечения помещений.

Преимущества:

1. Простой и понятный монтаж с наличием подробной инструкции;
2. Способ передачи данных между центральным элементом системы (хабом) и датчиками системы: беспроводной;
3. Большая дальность сигнала системы – до двух тысяч метров;
4. Поддержка WIFI и GSM сигналов;
5. Возможность получать уведомления различных типов – SMS, Push-сообщения, звонок на телефон);
6. Возможность предоставления полного либо частичного доступа необходимым пользователям;
7. Синхронизация оборудования посредством QR кода;
8. Управление системой на разных мобильных операционных системах: IOS и Android;
9. Наличие защиты / бампера от снятия любого датчика;
10. Независимые от электросети датчики;
11. Автономная работа центрального элемента управления – до шестнадцати часов;
12. Умная розетка может показывать расход электроэнергии при включенных в нее электроприборах. Отключается при критических перепадах напряжения;
13. Центральный элемент системы способен подключать до 100 устройств одновременно;
14. Наличие тревожной кнопки на брелке;
15. Относительно невысокая цена стартового набора – двести американских долларов и выше.

Недостатки:

1. Управление только через мобильные устройства. Невозможность управления через ПК и другие платформы;
2. Отсутствие собственной камеры видеонаблюдения. Необходимость приобретения камеры от сторонних производителей;
3. Базовый комплект не способен покрыть все, даже небольшое помещение. Необходимость приобретения дополнительного оборудования;
4. Отсутствие автономности датчиков – работа происходит только через непосредственно центральный элемент системы (хаб).

**2.2 BroadLink**

Так же, как и Ajax Systems, BroadLink – относительно молодая компания, основанная в 2013 году. Изначально компания являлась проектом, но была выбрана лучшим проектом венчурных инвестиций QPrize [23].

Проект был проинвестирован одним из самых крупных интернет-магазинов JD.com и ведущей компанией в отрасли безопасности интернет сетей - Qihu 360. Данные инвестиции дали большой толчок этому проекту и сегодня это один из крупнейших поставщиков решений домашней автоматизации. Продукция компании BroadLink сертифицирована и поставляется более чем в пятьдесят стран мира.

Данная система имеет ряд отличительных особенностей (как положительных, так и крайне отрицательных) в отношении своих конкурентов.

Как и практически все системы домашней автоматизации, система от BroadLink не нуждается в демонтаже чистовой отделки помещения, так как устройства не нуждаются в проводном подключении к сети. Однако каждое устройство нуждается в подключении к электросети и не работает от батареек, что является огромным минусом – в случае наличия большого количества датчиков понадобится огромное количества электроточек [24].

Для подключения всех объектов домашней автоматизации необходимо только наличие Wi-Fi. Каждое устройство можно внедрить в систему без каких-либо проблем – все устройства работают независимо от других и не влияют (в худшую сторону) на работоспособность устройств.

Особенностью данной системы является отсутствие центрального элемента автоматизации – хаба. Каждый объект (датчик, устройство) работает независимо от других устройств. За их состоянием следит мобильное приложение, в котором можно задать интересующие пользователя настройки [24].

Еще одной особенностью данной системы является наличие специального устройства, которое способно считывать инфракрасное излучение пультов управления телевизором, кондиционером и многим другим. В последствии это устройство способно самостоятельно управлять записанными командами пультов управления домашних систем. Необходимо лишь задать на каждую кнопку пульта управления соответствующей элемент в интерфейсе мобильного приложения от компании BroadLink.

Один из комплектов «Умного дома» от компании BroadLink представлен на рисунке 2.2.



**Рисунок 2.2 - Устройства BroadLink**

Черный элемент треугольной формы с надписью BroadLink, представленный на рисунке 2.2 (в центре изображения) – элемент, способный управлять домашними электронными техническими устройствами посредством ИК излучения [24].

Один из выделяющихся на фоне конкурентов элементов системы является многофункциональный датчик, способный получать показания сразу по пяти показаниям – влажности, уровня шума, температуры, уровень освещенности в помещении, чистота окружающего воздуха в помещении.

Также стоит отметить, что практически во устройствах BroadLink не просматривается единый формат в оформлении и внешнем виде устройств. Все они выглядят как набор совершенно не связанных между собой устройств, со своим особенным цветом, формой и композицией элементов управления. По внешней составляющей компания BroadLink сильно проигрывает своим конкурентам. Но данная особенность сказывается на цене устройств, они заметно дешевле своих конкурентов – базовый набор стоит от 150$ [25]. Но при покупке всего необходимого оборудования на большие площади помещений предложение компании BroadLink становится все более выгодным.

Преимущества:

1. Простой монтаж устройств без необходимости демонтажа чистовой отделки помещения;
2. Предлагает большое количество разнообразных датчиков (уровня шума, влажности, уровня освещенности в помещении, чистоты окружающего воздуха, температуры и многое другое);
3. Практически самая дешевая система автоматизации на рынке – от 150$;
4. Независимость устройств позволяет легко внедрять новые устройства в систему;
5. Система работает без центрального элемента управления – хаба;
6. Наличие собственной камеры видеонаблюдения;
7. Возможность управления и контролем системы из любой точки планеты (посредством интернет сети);
8. Возможность управления системой посредством мобильного приложения.

Недостатки:

1. Работа с системой только через мобильное приложение (без поддержки других систем);
2. Поддержка только Android платформы;
3. Плохая поддержка существующего приложения, регулярные вылеты системы и потеря связи с устройствами [26];
4. Отсутствия резервного источника питания всех элементов системы;
5. Пульт управления работает только на прием сигнала.
6. Относительно небольшая дальность сигнала – не больше пятидесяти метров.

**2.3 Xiaomi**

Xiaomi Corporation было основана в 2010 году. На сегодняшний день занимает четвертое место в Китае и шестое в мире по производству мобильных устройств [27].

Базовый комплект умного дома от Xiaomi включает в себя головное устройство системы, датчики открытия дверей и окон, датчик движения, «умная кнопка» [28].

Головное устройство играет роль не только центра управления системы автоматизации, но и является ночником, музыкальным центром. Для подключения системы необходимо скачать мобильное приложение Mi Home. Система позволяет задать определенные условия, при которых будут включаться объекты автоматизации. Например, можно задать такое условие, когда при открытии входной двери включается свет в помещении. Также можно задать полноценные сценарии работы объектов автоматизации, что позволяет более гибко настроить работу всей системы [28].

Стартовый набор (Smart Home Security Kit) от компании Xiaomi, включающий головное устройства, датчик открытия дверей, датчик открытия окон, датчик движения и «умную» кнопку, представлен на рисунке 2.3.



**Рисунок 2.3 - стартовый набор домашней автоматизации от компании Xiaomi**

Стоит такой комплект от 100$, что является минимальной ценой на рынке за аналогичных комплект. Однако, данный набор позволяет автоматизировать только одно помещение, что не всегда позволяет покрыть все потребности конечного пользователя [28].

Система от компании Xiaomi очень проста в управлении и подключении. Все объекты автоматизации (за исключением головного устройства) работают на батарейках, а срок автономной работы всех устройств – до 5 лет (по заявлению производителя).

Недостатком данной системы является необходимость приобретения переходников для розеток европейского типа. Также к минусам можно отнести различные подходы к подключению объектов автоматизации к головному устройству. Связано это с тем, что многие объекты автоматизации для компании Xiaomi производят другие производители, что в отсутствии единого стандарта подключения, приводит к хаотичному подходу к подключению устройств: QR-код, скрытая кнопка, многократное нажатие на сенсор и т.д.

Преимущества:

1. Простой и понятный монтаж с наличием подробной инструкции;
2. Способ передачи данных между центральным элементом системы (хабом) и датчиками системы: беспроводной;
3. Возможность масштабирования;
4. Единый, минималистичный стиль и компактность;
5. Использования энергоэффективного протокола ZigBee;
6. Гибкость системы – настраиваемые сценарии;
7. Наличие собственной камеры видеонаблюдения;
8. Независимые от электросети датчики;
9. Центральный элемент системы способен подключать до 30 устройств одновременно;
10. Минимальная цена стартового набора среди аналогов – сто американских долларов и выше.

Недостатки:

1. Управление только через мобильные устройства. Невозможность управления через ПК и другие платформы;
2. Маленький радиус действия сигнала – до десяти метров. Необходимость использования нескольких головных устройств в больших помещениях;
3. Центральный элемент системы не обладает источником резервного питания;
4. Базовый комплект не способен покрыть больше одного помещения. Необходимость приобретения дополнительного оборудования;
5. Отсутствие автономности датчиков – работа происходит только через непосредственно центральный элемент системы (хаб).

**2.4 Fibaro**

Польская компания Fibaro основана в 2010 году. Областью деятельности является IoT (Интернет вещей). Основной офис компании находится в Познани. Штат компании оставляет примерно четыреста человек [29].

Система автоматизации от компании Fibaro обладает качественной сборкой и стильным внешним видом. Все объекты автоматизации включаются в общую экосистему, которая настраивается под функции целевого помещения, а также под требования пользователя.

Доступ к пользовательским данным и данным дома защищен системами WAF и Anti-DDoS. Сообщения шифруются с использованием протокола TLS, а пароль шифруется с помощью алгоритма Bcrypt [30].

Система Fibaro обладает свойством Plug and Play, позволяющая подключить устройство сразу после его открытия из коробки. Также производитель внедрил функцию управления голосом, что сильно упрощает использование продуктом компании. Существует удаленный доступ ко всем объектам автоматизации: можно как отслеживать их состояние, так и получить возможность управления устройствами. Система Fibaro отслеживает внешние угрозы в виде попытки взлома, возникновения пожара, протечки воды и так далее.

Уникальной особенностью системы является возможность контроля энергопотребления всех устройств, находящихся в экосистеме Fibaro. Данная особенность позволяет найти источник высокого потребления электроэнергии, позволяя пользователю заменить ее более эффективное устройство, что, в свою очередь, сэкономит средства на оплате за электричество.

Система обладает встроенным термостатом для поддержания комфортной температуры в помещении. Производитель предоставляет множество устройств, позволяющих автоматизировать практически все бытовые действия в помещении и во вне. Так, например, Fibaro предлагает умную газонокосилку, которая способна выполнять свою работу без участия человека в определённое время, когда это будет удобно пользователю [30].

Основной недостаток системы от компании Fibaro – это ее цена. Так, например, один только умный домофон стоит большее тысячи долларов, а стартовый пакет обойдется не меньше чем в шестьсот долларов (информация на основании единственного дистрибьютера в Беларуси - Smart Control LLC) [31].

Основные устройства системы (головное устройство, умная розетка, пульт управления, датчик движения, датчик открытия двери, датчик дыма) представлены на рисунке 2.4.



**Рисунок 2.4 - основные устройства системы Fibaro**

Преимущества:

1. Способ передачи данных между центральным элементом системы (хабом) и датчиками системы: беспроводной;
2. Возможность рассылки сообщений на несколько устройств;
3. Единый, минималистичный стиль и компактность;
4. Использования популярного протокола Z-Wave [32];
5. Гибкость системы – настраиваемые сценарии;
6. Наличие собственной камеры видеонаблюдения;
7. Независимые от электросети датчики;
8. В датчик протекания воды оснащена сирена;
9. Возможность отслеживания уровня энергопотребления, а также отключение устройств при критических скачках напряжения;
10. Возможность увеличения радиуса сигнала за счет свойства всех объектов системы – способность ретрансляции сигнала;
11. Внедрено голосовое управление от компании Google.

Недостатки:

1. Сложность монтажа и необходимость работы специалиста;
2. Относительно небольшой радиус действия – пятьдесят метров. Проблема решается ретрансляцией сигнала;
3. Центральный элемент системы не обладает источником резервного питания;
4. Стоимость системы от шестисот долларов;
5. Требование к установке программного обеспечения на пользовательский компьютер. Мобильная версия приложения не содержит полный набор функций;
6. Отсутствие автономности датчиков – работа происходит только через непосредственно центральный элемент системы (хаб).

**2.5 Сравнительный анализ**

Сравнительный анализ всех рассмотренных систем отображен в таблице 2.1.

Таблица 2.1 - Сравнительный анализ систем автоматизации

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Система автоматизации | Наличие хаба | Протокол подключения | Стоимость базового комплекта | Наличие видеокамеры |
| Ajax Systems | + | Jeweller | от 200$ | - |
| BroadLink | - | - | от 150$ | + |
| Xiaomi | + | ZigBee | от 100$ | + |
| Fibaro | + | Z-Wave | от 600$ | + |

Продолжение таблицы 2.1

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Система автоматизации | Простота монтажа | Способ оповещения | Поддержка платформ | Поддержка GSM |
| Ajax Systems | + | SMS, Push, звонок | Mobile, PC | + |
| BroadLink | + | Push | Mobile | - |
| Xiaomi | + | Push | Mobile | - |
| Fibaro | - | SMS, Push | PC, Mobile (+-) | - |

Продолжение таблицы 2.1

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Система автоматизации | Поддержка ОС | Дальность соединения | Автономность датчиков | Резервное питание |
| Ajax Systems | IOS, Android, Windows | до 2000м | + | + |
| BroadLink | Android | до 50м | - | - |
| Xiaomi | Android, IOS (+-) | до 10м | + | - |
| Fibaro | IOS, Android, Windows | до 50м + ретрансляция | + | - |

Окончание таблицы 2.1

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Система автоматизации | Настраиваемые сценарии | Единый стандарт подключения | Единый стиль устройств | Резервное питание |
| Ajax Systems | - | + | + | + |
| BroadLink | - | - | - | - |
| Xiaomi | + | - | + | - |
| Fibaro | + | + | + | - |

ГЛАВА 3 ПРАКТИЧЕСКАЯ ЧАСТЬ

**3.1 Объектно-ориентированный анализ и проектирование системы**

Реализация автоматизированной системы управления «Умный дом» современного типа требует большого количества времени и средств, так как необходимо учитывать многие факторы при проектировании данной системы.

Современная система должна соответствовать следующим критериям:

1. Удобство использования;
2. Производительность;
3. Безопасность;
4. Доступность.

Для реализации автоматизированной системы необходимо обладать следующими навыками/знаниями:

1. Владение языком программирования, позволяющим реализовать веб сервис, обрабатывающий клиентские запросы.
2. Умение разворачивать сервер на выделенном пространстве.
3. Умение разворачивать, настраивать и поддерживать базу данных на выделенном пространстве.
4. Понимание аппаратных аспектов, работы с компьютерными сетями.
5. Понимание взаимодействия объектов автоматизации между собой либо между центральным элементом управления – хабом.

В связи с эти, существующие на сегодняшний день системы не удовлетворяют одному из самых важных критериев современных автоматизированных систем – доступность. В большинстве своем такие системы стоят от тысячи долларов и выше (с учетом покупки оборудования и установки), что позволить себе может далеко не каждый.

Исходя из изложенных выше факторов возникает задача по реализации доступной автоматизированной системы управления «Умный дом», способной покрыть основные потребности среднестатистического пользователя.

Потребности заключаются в следующем:

* Возможность удалённого управления системой;
* Возможность задания параметров автоматизации;
* Возможность отслеживания текущего состояния системы;
* Возможность самостоятельного внедрения новых устройств в систему.

**3.2 Проектирование модели**

Главной целью проектирования моделей является отображение функциональной структуры объекта, то есть производимые ими действия и связи между этими действиями. Наиболее распространенным средством моделирования данных являются диаграммы «сущность-связь» (ERD), которые предназначены для графического представления моделей данных разрабатываемой программной системы и предлагают некоторый набор стандартных обозначений для определения данных и отношений между ними. С помощью этого вида диаграмм можно описать отдельные компоненты концептуальной модели данных и совокупность взаимосвязей между ними, имеющих важное значение для разрабатываемой системы. Основными понятиями данной нотации являются понятия сущности и связи. При этом под сущностью понимается произвольное множество реальных или абстрактных объектов, каждый из которых обладает одинаковыми свойствами и характеристиками. В этом случае каждый рассматриваемый объект может являться экземпляром одной и только одной сущности, должен иметь уникальное имя или идентификатор, а также отличаться от других экземпляров данной сущности. Связь определяется как отношение или некоторая ассоциация между отдельными сущностями. Примерами связей могут являться родственные отношения типа «отец-сын» или производственные отношения типа «начальник-подчиненный». Другой тип связей задается отношениями «иметь в собственности» или «обладать свойством».

Графическая модель данных строится таким образом, чтобы связи между отдельными сущностями отражали не только семантический характер соответствующего отношения, но и дополнительные аспекты обязательности связей, а также кратность участвующих в данных отношениях экземпляров сущностей. Информационная модель базы представлена на диаграмме «Сущность-связь». Данная диаграмма представлена на рисунке 1.1 и на рисунке 1.2 в нотации Баркера.

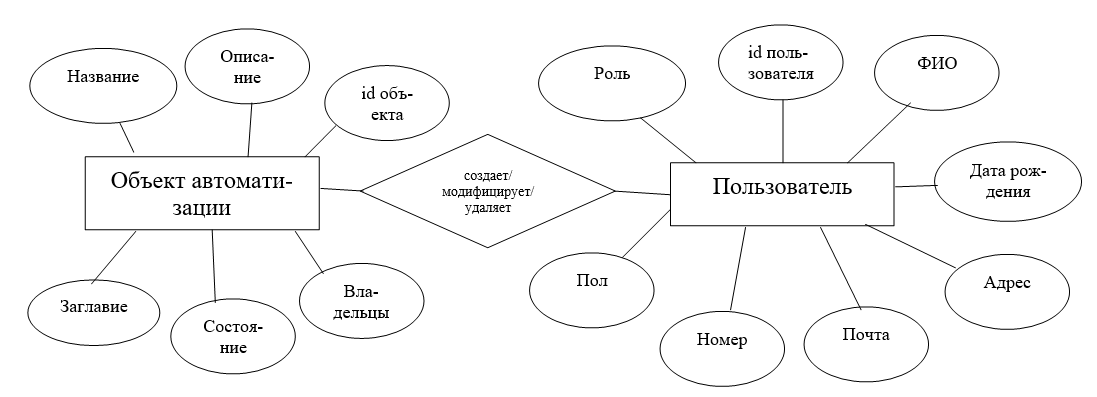
Исходя из исследования предметной области, можно выделить следующие сущности разработки: пользователь, объект автоматизации.

Для сущности «Пользователь» можно выделить следующие атрибуты:

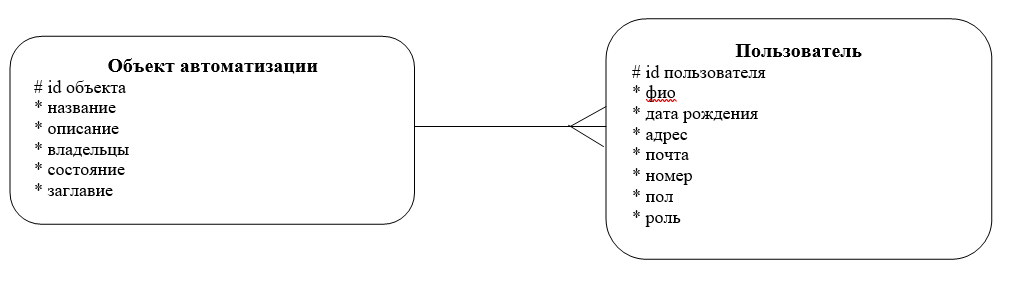
* фамилия имя и отчество пользователя;
* id пользователя;
* роль;
* пол;
* номер телефона;
* электронная почта;
* адрес;
* дата рождения.

Для сущности «Объект автоматизации» можно выделить следующие атрибуты:

* название;
* описание;
* id объекта;
* владельцы;
* состояние;
* заглавие;



**Рисунок 3.1 - Диаграмма ERD в нотации Ричарда Чена**



**Рисунок 3.2 - Диаграмма «сущность-связь» в нотации Баркера**

В рамках языка UML все представления о модели сложной системы фиксируются в виде специальных графических конструкций – диаграмм. В терминах языка UML определены следующие виды диаграмм: диаграмма вариантов использования, диаграмма классов, диаграмма деятельности, диаграмма последовательности, диаграмма компонентов.

Перечень этих диаграмм представляет собой неотъемлемую часть графической нотации языка UML, сам процесс объектно-ориентированного программирования (ООП) неразрывно связан с процессом построения этих диаграмм. Совокупность построенных таким образом диаграмм содержит всю информацию, необходимую для реализации проекта сложной системы.

Диаграмма – граф специального вида, состоящий из вершин в форме геометрических фигур, которые связаны между собой ребрами или дугами.

Суть диаграммы вариантов использования состоит в следующем: проектируемая система представляется в виде множества сущностей или актеров, взаимодействующих с системой с помощью, так называемых вариантов использования.

Варианты использования описывают не только взаимодействия между пользователями и сущностью, но также реакции сущности на получение отдельных сообщений от пользователей и восприятие этих сообщений за пределами сущности. Варианты использования могут включать в себя описание особенностей способов реализации сервиса и различных исключительных ситуаций, таких как корректная обработка ошибок системы. Множество вариантов использования в целом должно определять все возможные стороны ожидаемого поведения системы.

В данной проектируемой системе в качестве актера выступает пользователь системы автоматизации, который служит источником воздействия на моделируемую систему.

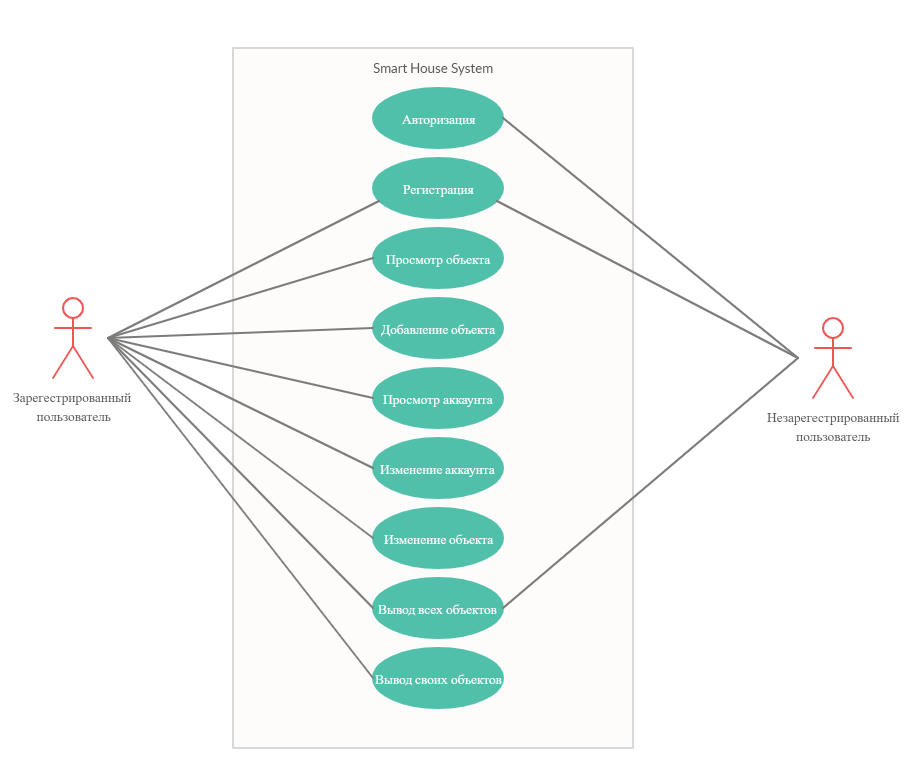
К основным функциям разрабатываемой программы относятся следующие функции:

* зарегистрировать нового пользователя;
* осуществить работу с базой данных;
* фиксировать данные о пользователе;
* добавить новый объект;
* управление объектами;
* отслеживание статуса существующих объектов;

К вспомогательным функциям программы относятся:

* вывод собственных объектов (добавленных конкретным пользователем);
* вывод информации о проекте.

Диаграмма вариантов использования представлена рисунке 3.3.



**Рисунок 3.3 - Диаграмма вариантов использования**

Центральное место в ООП занимает разработка логической модели системы в виде диаграммы классов. Нотация классов в языке UML проста и интуитивно понятна. Схожая нотация применяется и для объектов — экземпляров класса, с тем различием, что к имени класса добавляется имя объекта, и вся надпись подчеркивается.

Диаграмма классов служит для представления статической структуры модели системы в терминологии классов объектно-ориентированного программирования. Диаграмма может отражать различные взаимосвязи между отдельными сущностями предметной области, такими как объекты и подсистемы, а также описывать их внутреннюю структуру и типы отношений. На данной диаграмме не указывается информация о временных аспектах функционирования системы. С этой точки зрения диаграмма классов является дальнейшим развитием концептуальной модели проектируемой системы.

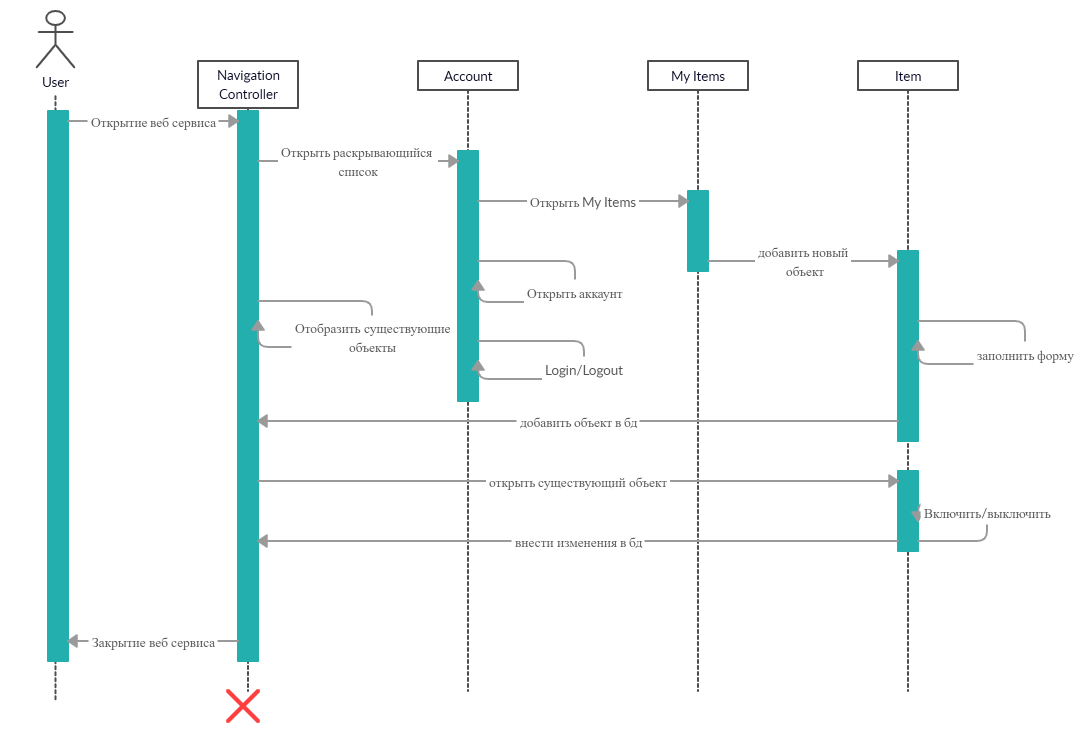
Диаграмма классов представляет собой некоторый граф, вершинами которого являются элементы типа «классификатор» и которые связаны различными типами структурных отношений. Следует заметить, что диаграмма классов может также содержать интерфейсы, пакеты, отношения и даже отдельные экземпляры, такие как объекты и связи. Когда говорят о данной диаграмме, имеют в виду статическую структурную модель проектируемой системы. Поэтому диаграмму классов принято считать графическим представлением таких структурных взаимосвязей логической модели системы, которые не зависят от времени.

Класс в языке UML служит для обозначения множества объектов, которые обладают одинаковой структурой, поведением и отношениями с объектами из других классов. Графически класс изображается в виде прямоугольника, который дополнительно может быть разделен горизонтальными линями на разделы или секции. Здесь указывается: имя класса, атрибуты и операции (методы).

При моделировании поведения проектируемой или анализируемой системы возникает необходимость детализировать особенности алгоритмической и логической реализации выполняемых системой операций. Для моделирования процесса выполнения операций в языке UML используются так называемые диаграммы деятельности. Каждое состояние на диаграмме деятельности соответствует выполнению некоторой элементарной операции, переход в следующее состояние срабатывает только при завершении этой операции. Графически диаграмма деятельности представляется в форме графа, вершинами которого являются состояния действия, а дугами - переходы от одного состояния действия к другому.

Диаграммы деятельности - частный случай диаграмм состояний. Основная цель использования таких диаграмм - визуализация особенностей реализации операций классов, когда необходимо представить алгоритмы их выполнения.

Для моделирования взаимодействия объектов в UML используется диаграмма последовательности. Диаграмма последовательности представлена на рисунке 3.4.



**Рисунок 3.4 - Диаграмма последовательности**

Все предыдущие диаграммы отражали концептуальные аспекты построения модели системы и относились к логическому уровню представления и оперировали понятиями, которые не имеют самостоятельного материального воплощения. Для описания реальных сущностей предназначен другой аспект модельного представления, а именно физическое представление модели. Для физического представления модели используются диаграмма компонентов.

Диаграмма компонентов описывает объекты реального мира – компоненты программного обеспечения. Эта диаграмма позволяет определить архитектуру разрабатываемой системы, установив зависимости между программными компонентами, в роли которых может выступать исходный, бинарный и исполняемый код.

Диаграмма содержит следующие компоненты:

Сервер:

* Базу данных MySQL;
* Собранный артефакт Java;
* Apache Tomcat Server.

Клиент:

* Браузер для обработки http запросов и взаимодействия с html;
* Html как способ представления страницы;
* JavaScript как логическая часть клиентской стороны.

Диаграмма компонентов представлена на рисунке

**3.3 Вычислительная система**

Конфигурация персонального компьютера, на котором будет разрабатываться программа:

* процессор Intel(R) Core(TM) i5-9400F CPU 2,9 GHz;
* материнская плата ASRock H310CM-DVS;
* оперативная память Corsair DDR4-1600MHz, 16320Mb;
* видеокарта GeForce GTX 1660 Ti (6 ГБ);
* винчестер TOSHIBA HDWD110 (931 ГБ);
* твердотельный накопитель GIGABYTE GP-GSTFS31240GNTD (223 ГБ);
* клавиатура и мышь Logitech Cordless MX3200.

Инструменты разработки:

* операционная система Microsoft Windows 10 Профессиональная;
* среда разработки IntelliJ IDEA 2019.3 x64 Ultimate Edition;
* язык программирования Java;
* MYSQL.

Для разработки данного проекта была использована операционная система Microsoft Windows 10 Профессиональная. В перспективе она призвана заменить собой устаревшую операционную систему Microsoft Windows 7.

Что касается среды разработки, то был выбран продукт компании Jet Brains. IntelliJ IDEA — линейка бесплатных интегрированных сред разработки, позволяющая создавать приложения различных типов (десктопные, консольные, веб).

Java — строго типизированный объектно-ориентированный язык программирования, разработанный компанией Sun Microsystems (в последующем приобретённой компанией Oracle). Разработка ведётся сообществом, организованным через Java Community Process, язык и основные реализующие его технологии распространяются по лицензии GPL. Права на торговую марку принадлежат корпорации Oracle.

Приложения Java обычно транслируются в специальный байт-код, поэтому они могут работать на любой компьютерной архитектуре с помощью виртуальной Java-машины. Дата официального выпуска — 23 мая 1995 года. На 2019 год Java — один из самых популярных языков программирования.

Программы на Java транслируются в байт-код Java, выполняемый виртуальной машиной Java (JVM) — программой, обрабатывающей байтовый код и передающей инструкции оборудованию как интерпретатор.

Достоинством подобного способа выполнения программ является полная независимость байт-кода от операционной системы и оборудования, что позволяет выполнять Java-приложения на любом устройстве, для которого существует соответствующая виртуальная машина. Другой важной особенностью технологии Java является гибкая система безопасности, в рамках которой исполнение программы полностью контролируется виртуальной машиной. Любые операции, которые превышают установленные полномочия программы (например, попытка несанкционированного доступа к данным или соединения с другим компьютером), вызывают немедленное прерывание.

Часто к недостаткам концепции виртуальной машины относят снижение производительности. Ряд усовершенствований несколько увеличил скорость выполнения программ на Java:

* применение технологии трансляции байт-кода в машинный код непосредственно во время работы программы (JIT-технология) с возможностью сохранения версий класса в машинном коде;
* обширное использование платформенно-ориентированного кода (native-код) в стандартных библиотеках;
* аппаратные средства, обеспечивающие ускоренную обработку байт-кода (например, технология Jazelle, поддерживаемая некоторыми процессорами архитектуры ARM).

MySQL является решением для малых и средних приложений. Входит в состав серверов WAMP, AppServ, LAMP и в портативные сборки серверов Денвер, XAMPP, VertrigoServ. Обычно MySQL используется в качестве сервера, к которому обращаются локальные или удалённые клиенты, однако в дистрибутив входит библиотека внутреннего сервера, позволяющая включать MySQL в автономные программы.

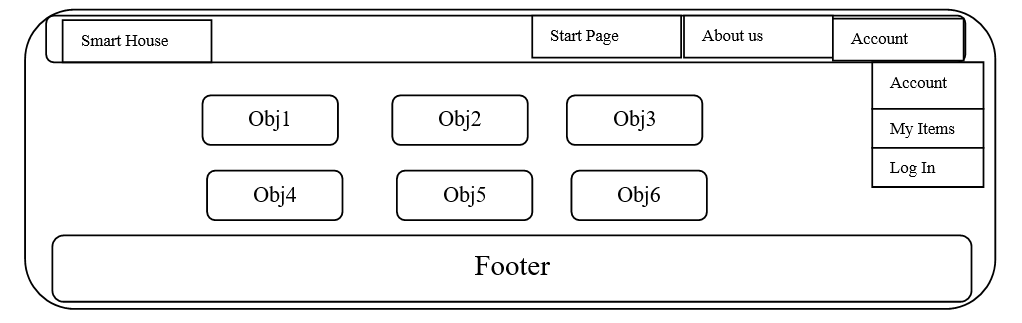
Гибкость СУБД MySQL обеспечивается поддержкой большого количества типов таблиц: пользователи могут выбрать как таблицы типа MyISAM, поддерживающие полнотекстовый поиск, так и таблицы InnoDB, поддерживающие транзакции на уровне отдельных записей. Более того, СУБД MySQL поставляется со специальным типом таблиц EXAMPLE, демонстрирующим принципы создания новых типов таблиц. Благодаря открытой архитектуре и GPL-лицензированию, в СУБД MySQL постоянно появляются новые типы таблиц.

**3.4 Концептуальный прототип**

Концептуальный прототип представляет собой описание внешнего пользовательского интерфейса – систему меню и страниц.

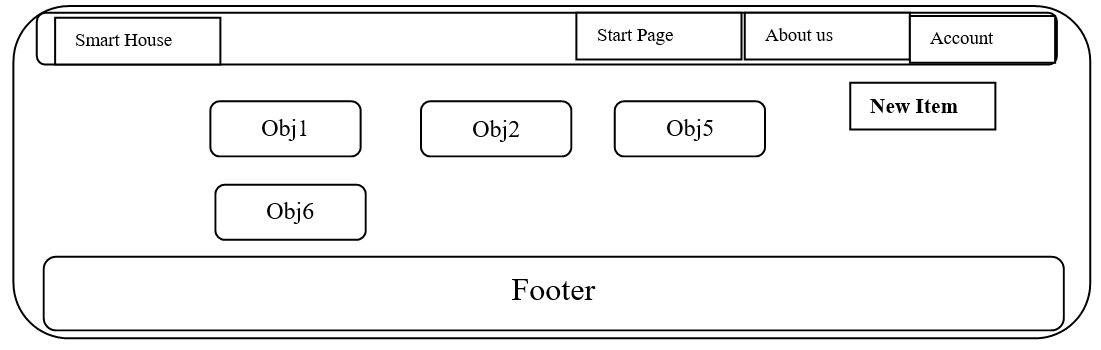
Все действия можно будет выполнить с помощью главной страницы, которая будет появляться сразу при открытии веб ресурса. Главная страница будет иметь список всех объектов автоматизации, позволять осуществлять различные манипуляции с объектами, а также получать справочную информацию о веб ресурсе. Используя меню, пользователь должен иметь возможность изменить аккаунт пользователя, свои пользовательские данные.

Главная страница программы будет представлена страницей Welcome на которой находятся основные элементы управления объектами (при переходе на конкретный объект). С помощью открывающегося меню Account, About и других на главной странице можно получить доступ к другим страницам работы с объектами, просмотра собственных объектов, добавление объектов, справочной информацией о сервисе и возможность выйти из аккаунта пользователя. Структура меню представлена в таблице 3.5.



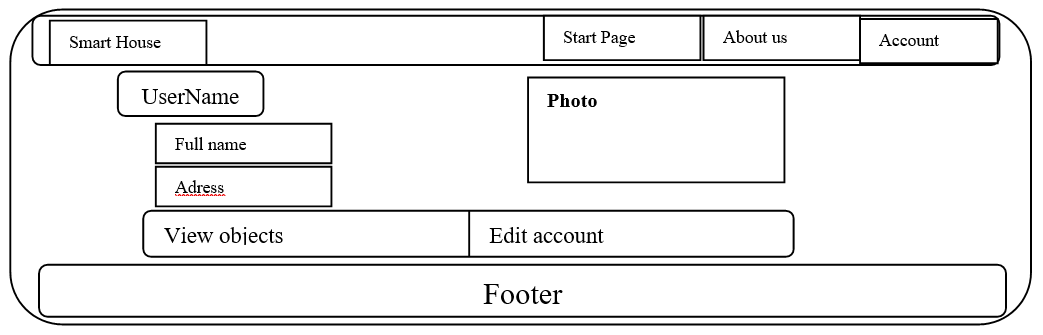
**Рисунок 3.5 - Структура главной страница**

Для отображения пользовательских объектов создана отдельная страница, на которой будет кнопка добавления нового объекта - рисунок 3.6.



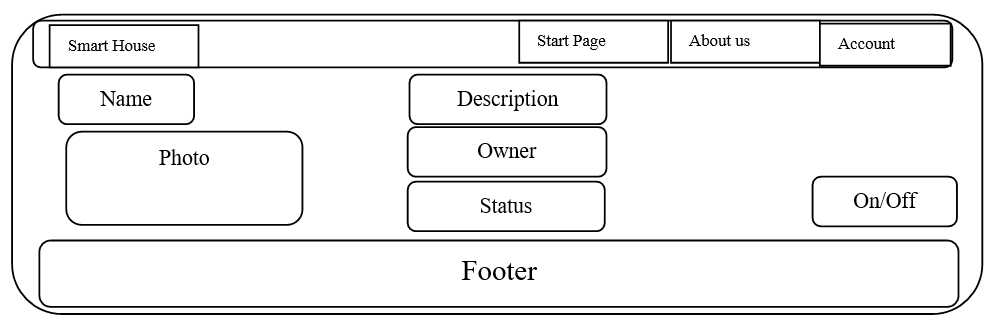
**Рисунок 3.6 - Структура страницы пользовательских объектов**

Для создания обновления пользовательских данных используется новая страница – рисунок 3.7.



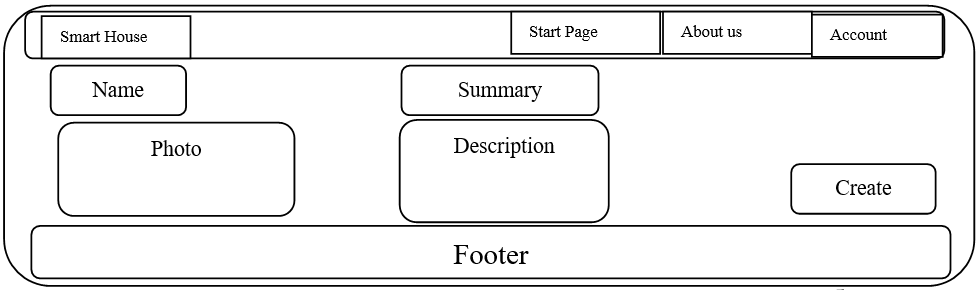
**Рисунок 3.7 - Структура страницы аккаунта пользователя**

Страница отображения объекта представлена на рисунке 3.8.



**Рисунок 3.8 - Структура страницы отображения объекта**

Страница создания объекта представлена на рисунке 3.9.



**Рисунок 3.9 - Структура страницы создания объекта**

**3.5 Организация данных**

MySQL является решением для малых и средних приложений. Входит в состав серверов WAMP, AppServ, LAMP и в портативные сборки серверов Денвер, XAMPP, VertrigoServ. Обычно MySQL используется в качестве сервера, к которому обращаются локальные или удалённые клиенты, однако в дистрибутив входит библиотека внутреннего сервера, позволяющая включать MySQL в автономные программы. Рассмотрим необходимые таблицы, описывающие объекты данной системы.

Таблица «user» хранит информацию о вкладах, структура приведена в таблице 3.1

Таблица 3.1 - Структура таблицы «user»

|  |  |  |  |
| --- | --- | --- | --- |
| Имя поля | Тип данных | Размерность | Описание |
| id | int | 11 | Идентификационный номер пользователя |
| username | varchar | 45 | Имя пользователя в системе вкладчика |
| password | varchar | 100 | Закодированный пароль |
| email | varchar | 45 | Электронная почта |
| fio | varchar | 100 | Фамилия имя отчество |
| telephone | varchar | 45 | Номер телефона |
| birth | varchar | 45 | Дата рождения |

Продолжение таблицы 3.1

|  |  |  |  |
| --- | --- | --- | --- |
| city | varchar | 45 | Город |
| create date | varchar | 45 | Дата создания аккаунта |
| status | varchar | 45 | Состояние |
| photo | varchar | 45 | Идентификатор фотографии |
| sex | varchar | 45 | Пол |
| secret | varchar | 45 | Секретное слово (для авторизации) |

Таблица «role» хранит информацию о существующих ролях, которыми может обладать пользователь. Структура приведена в таблице 3.2.

Таблица 3.2 — Структура таблицы «role»

|  |  |  |  |
| --- | --- | --- | --- |
| Имя поля | Тип данных | Размер,  байт | Описание |
| id | int | 11 | Идентификационный номер роли |
| name | varchar | 45 | Название роли |

Таблица «user\_role» хранит информацию о связи пользователя и конкретной роли. В данной таблице находятся следующие поля: идентификатор пользователя, идентификатор роли. Структура приведена в таблице 3.3.

Таблица 3.3 — Структура таблицы «user\_role»

|  |  |  |  |
| --- | --- | --- | --- |
| Имя поля | Тип данных | Размер,  байт | Описание |
| id\_user | int | 11 | Идентификационный номер пользователя |
| id\_role | int | 11 | Идентификационный номер роли |

Таблица «item» хранит информацию о созданных пользовательских объектах. В данной таблице находятся следующие поля: идентификатор объекта, имя, ссылка на внешние файлы, заглавие, описание, состояние объекта. Структура приведена в таблице 3.4.

Таблица 3.4 — Структура таблицы «item»

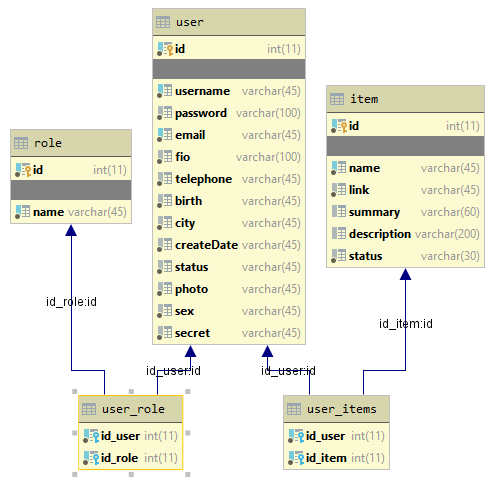
|  |  |  |  |
| --- | --- | --- | --- |
| Имя поля | Тип данных | Размерность | Описание |
| id | int | 11 | Идентификационный номер объекта |
| name | varchar | 45 | Имя объекта в системе |
| link | varchar | 45 | Идентификатор графического материала |
| summary | varchar | 60 | Заголовок |
| description | varchar | 200 | Описание объекта |
| status | varchar | 30 | Состояние объекта |

Таблица «user\_items» хранит информацию о связи пользователя и существующего пользовательского объекта. Структура приведена в таблице 3.5.

Таблица 3.5 — Структура таблицы «user\_items»

|  |  |  |  |
| --- | --- | --- | --- |
| Имя поля | Тип данных | Размер,  байт | Описание |
| id\_user | int | 11 | Идентификационный номер пользователя |
| id\_item | int | 11 | Идентификационный номер объекта |

На рисунке 3.10 представлена физическая структура базы данных



**Рисунок 3.10 - Физическая структура базы данных**

**3.6 Функции: логическая и физическая организация**

На основании диаграммы вариантов использования системы «Умный дом» были реализованы следующие функции:

* авторизация;
* регистрация;
* просмотр объекта;
* добавление объекта;
* просмотр аккаунта;
* изменение аккаунта;
* изменение объекта;
* вывод всех объектов;
* вывод своих объектов.

Рассмотрим основные функции программы.

За открытие стартовой страницы отвечает метод:

@RequestMapping(value = {**"/"**, **"/welcome"**}, method = RequestMethod.***GET***)  
**public** String main(Model model, HttpServletRequest request) {  
 Authentication auth = SecurityContextHolder.*getContext*().getAuthentication();  
 String name = auth.getName();  
 Locale locale = RequestContextUtils.*getLocale*(request);  
 ResourceBundle bundle = ResourceBundle.*getBundle*(**"locales.messages"**, locale);  
 model.addAttribute(**"keys"**, bundle.getKeys());  
 model.addAttribute(**"items"**, **itemService**.getAllItems());  
 **return "/welcome"**;  
}

За добавление объекта отвечает метод:

@RequestMapping(value = **"/item/add"**, method = RequestMethod.***POST***)  
 **public** String addItem(@RequestParam(**"file"**) MultipartFile file, HttpServletRequest request) {  
 Authentication auth = SecurityContextHolder.*getContext*().getAuthentication();  
 String username = auth.getName();Item item = **new** Item();  
 item.setLink(**"-1"**);User owner = **userService**.findByUsername(username);item.setName(request.getParameter(**"name"**));  
 item.setSummary(request.getParameter(**"summary"**));  
 item.setStatus(**"OFF"**);  
 **itemService**.save(item);  
 item = **itemService**.getByNameAndLink(item.getName(), item.getLink());  
 owner.getOwnItems().add(item);  
 **userService**.update(owner);**return "/my\_items"**;  
 }

За открытие страницы аккаунта отвечает метод:

@RequestMapping(value = **"/account/{name}"**, method = RequestMethod.***GET***)  
**public** String account(@PathVariable String name, Model model) {  
 User currentUserFull = **userService**.findByUsername(name);  
 model.addAttribute(**"user"**, currentUserFull);  
 **return "account"**;  
}

За регистрацию отвечает метод:

@RequestMapping(value = **"/registration"**, method = RequestMethod.***POST***)  
**public** String registration(@ModelAttribute(**"userForm"**) User userForm, BindingResult bindingResult, Model model) {  
 **userValidator**.validate(userForm, bindingResult);  
  
 **if** (bindingResult.hasErrors()) {  
 **return "registration"**;  
 }  
  
 **userService**.save(userForm);  
 **securityService**.autoLogin(userForm.getUsername(), userForm.getConfirmPassword());  
  
 **return "redirect:/welcome"**;  
}

За авторизацию отвечает метод:

@RequestMapping(value = **"/login"**, method = RequestMethod.***GET***)  
**public** String login(Model model, String error, String logout) {  
 **if** (error != **null**) {  
 System.***out***.print(error);  
 model.addAttribute(**"error"**, **"Username or password is incorrect."**);  
 }  
  
 **if** (logout != **null**) {  
 model.addAttribute(**"message"**, **"Logged out successfully."**);  
 }  
  
 **return "login"**;  
}

**3.7 Описание программного средства**

Предметной областью решаемой задачи является домашняя автоматизация системы внедряемых объектов, которые могут выполнять операции без непосредственного участия человека.

Из всех задач, которые будет решать разрабатываемое программное средство, можно выделить ряд основных:

* предоставление списка существующих объектов;
* возможность поиска собственных объектов;
* возможность отслеживания состояния объекта;
* возможность управлять объектом без прямого участия;
* возможность управления пользовательской информацией.

Разрабатываемое программное средство позволяет управлять пользовательскими объектами без непосредственного участия пользователя в процессе. С помощью данного программного средства пользователь сможет отслеживать состояние внедренных объектов, что является очень полезной функцией (например, проверить, выключен ли нагревательный прибор).

Предусмотрены средства защиты и разграничения прав доступа к данным, так как один сервер будет обрабатывать запросы многих пользователей системы. То есть к серверу могут посылать запросы пользователи разных жилых помещений, которые не владеют информацией друг о друге.

Входными данными при добавлении пользователя являются поля соответствующей таблицы, а именно:

* имя пользователя (уникально);
* пароль
* электронная почта (заполняется в процессе редактирования аккаунта);
* фио (заполняется в процессе редактирования аккаунта);
* номер телефона (заполняется в процессе редактирования аккаунта);
* дата рождения (заполняется в процессе редактирования аккаунта);
* город (заполняется в процессе редактирования аккаунта);
* статус (модерируется администратором);
* фото (заполняется в процессе редактирования аккаунта);
* пол (заполняется в процессе редактирования аккаунта);
* секретное слово.

Входными данными для создания объекта являются:

* название в системе
* идентификатор графического материала (опционально);
* заголовок;
* описание;

Выходные данные представлены в виде таблиц, хранящихся на сервере базы данных MySQL.

3.7 Функциональное тестирование

Технические требования. Минимальные системные требования к пользователю системы: любое устройство с выходом в сеть интернет.

Минимальные системные требования к администратору системы (на примере пользовательского ПК):

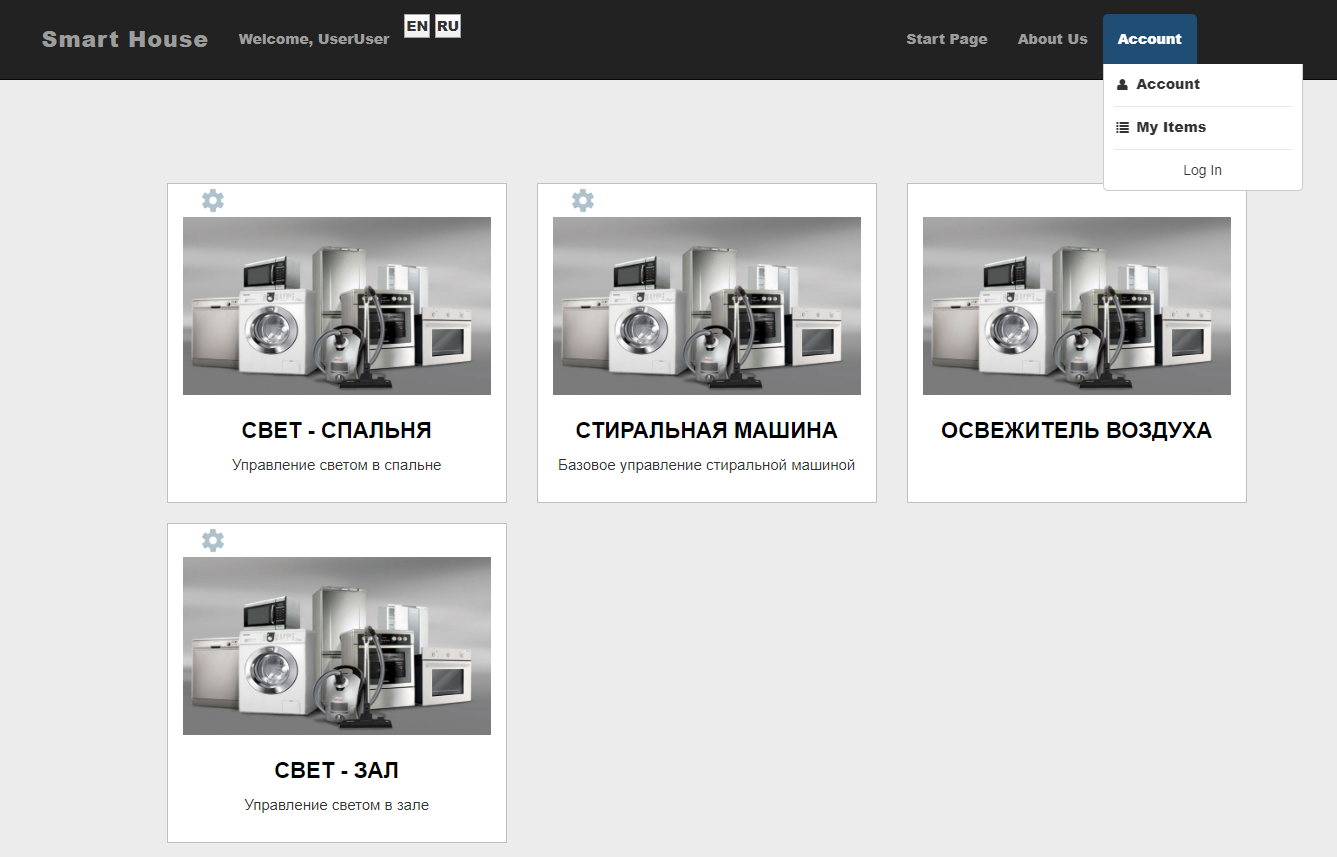
* 100 Мб свободного места на винчестере;
* GB свободной оперативной памяти;
* процессор Intel Core I3 2100 MHz или совместимый аналог;
* операционная система семейства Microsoft Windows / Unix;
* клавиатура, мышь.

В процессе написания программного продукта необходимо производить тестирование на правильность работы приложения. Одной из основных задач тестирования является устранение ошибок, происходящих при вводе данных.

Функциональное тестирование – это тестирование функций приложения на соответствие требованиям. Оценка производится в соответствии с ожидаемыми и полученными результатами (на основании функциональной спецификации), при условии, что функции отрабатывали на различных значениях.

Тестирование программы будет производиться последовательно, переходя из одной части программы в другую. Во время теста будут проверяться все действия с программой, навигация пунктам меню, которые может произвести пользователь. После чего, все собранные и найденные ошибки будут исправлены.

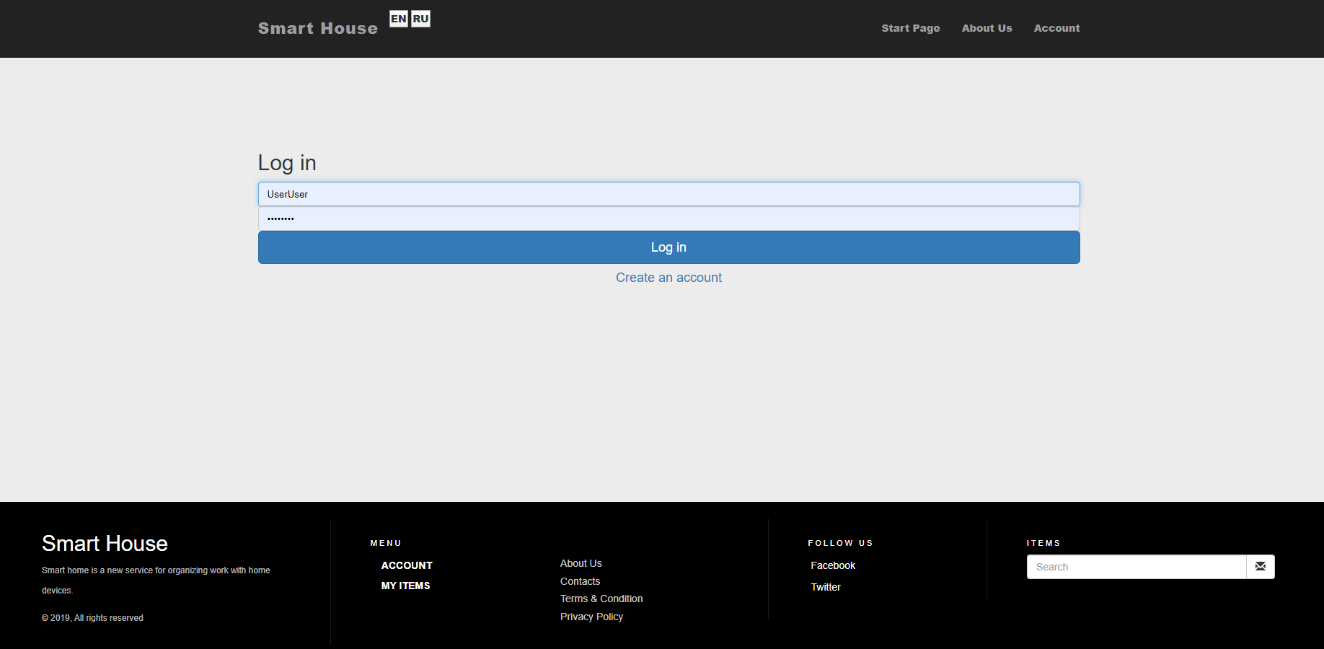
На рисунке 3.11 представлена главная страница веб-приложения.



**Рисунок 3.11 - Главная страница веб сервиса**

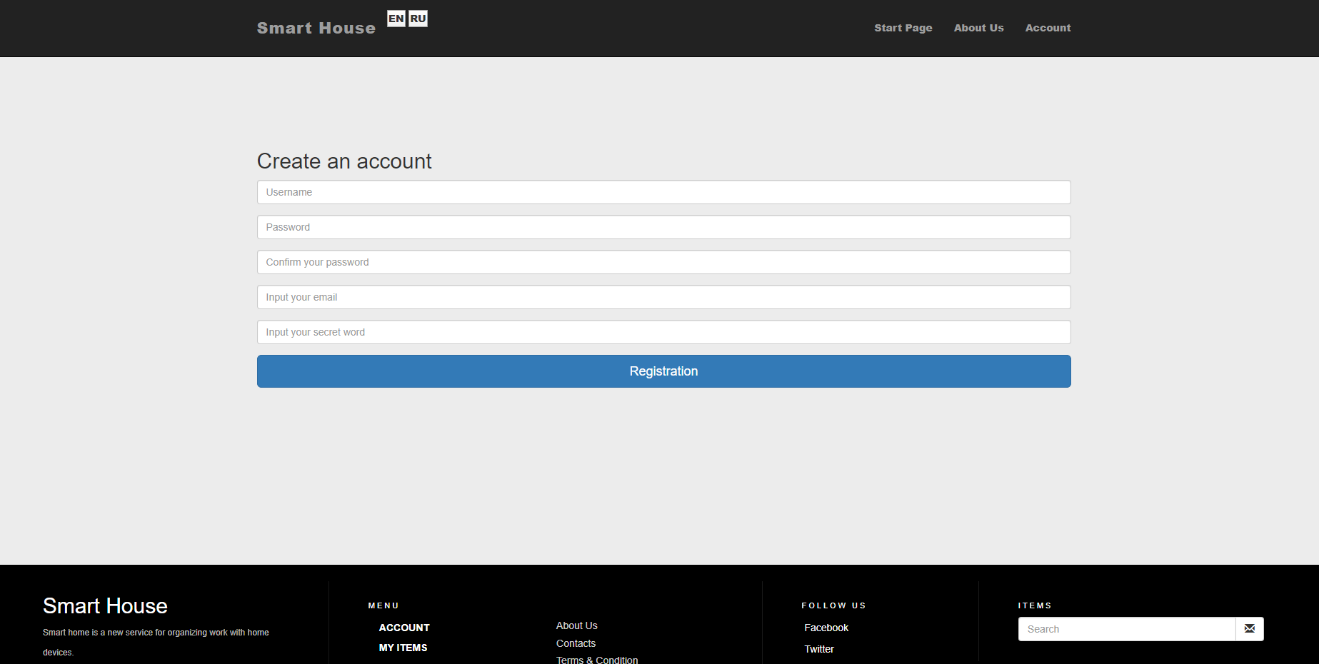
На данном этапе необходимо проверить программный продукт методами, описанными выше. При запуске программы появляется главное окно программы, вид которого приведен на рисунке 3.11, из этого окна происходят некоторые действия программы.

В программном продукте предусмотрены доступ к данной программе с помощью пароля и логина пользователя, для защиты информации предусмотрено разграничение прав пользователей. При попытке перехода на страницу объекта произойдет перенаправление на страницу логина с возможностью регистрации. На рисунке 3.12 представлена страница авторизации.



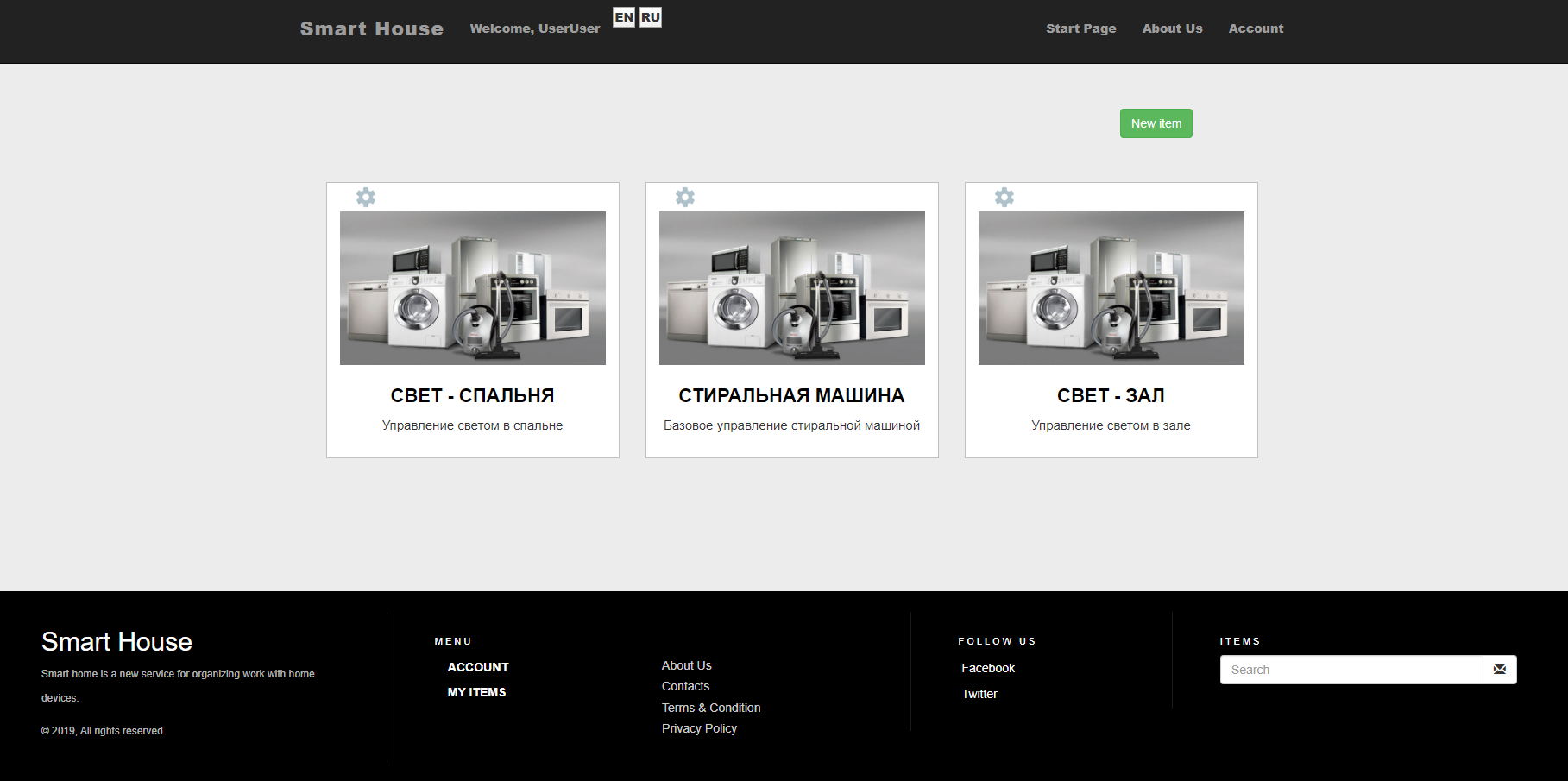
**Рисунок 3.12 - Страница авторизации**

На рисунке 3.13 представлена страница регистрации. Отметим, что на данном этапе шифруется пароль алгоритмом Bcrypt, а сессия защищена csrf токеном.



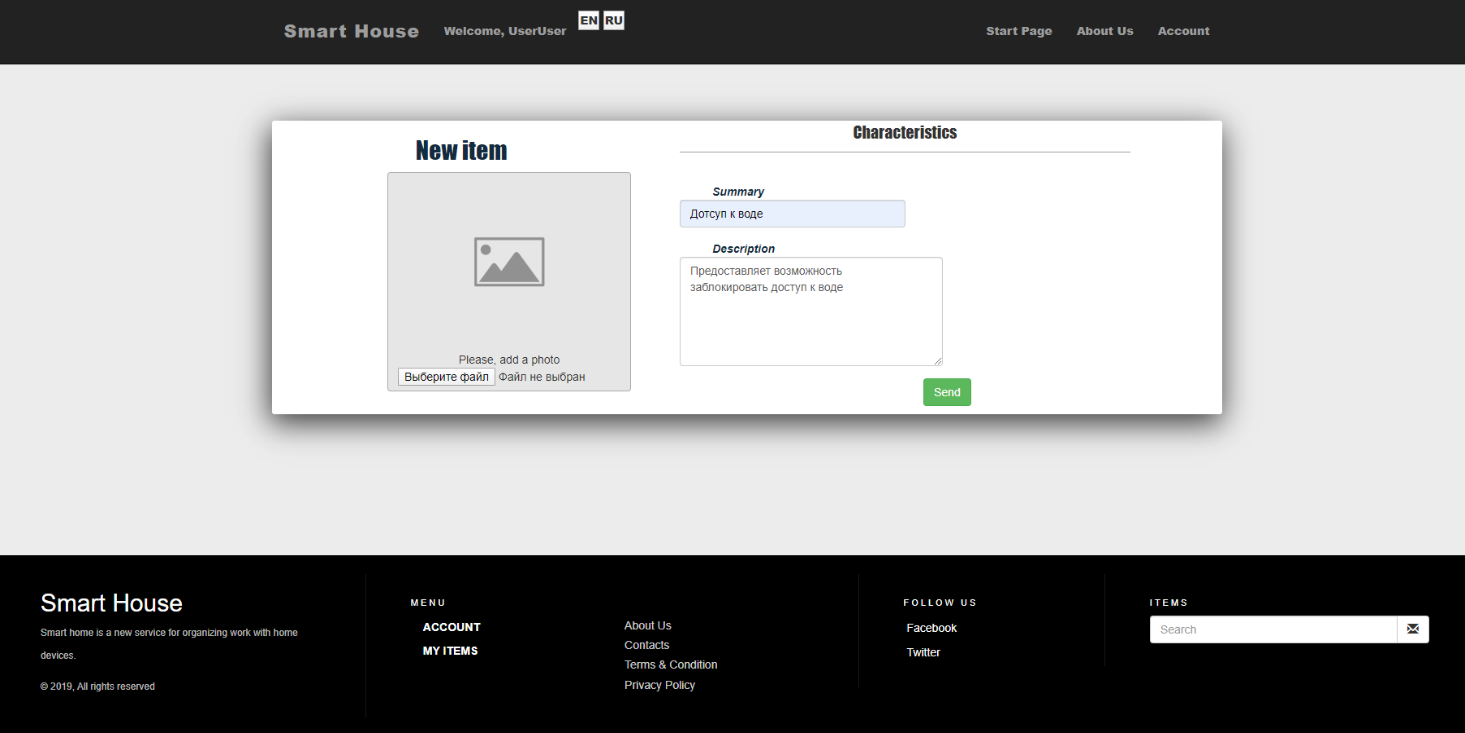
**Рисунок 3.13 - Страница регистрации**

Протестируем функцию добавления нового объекта. Для этого на главной странице необходимо выбрать открыть раскрывающийся список “Accounts”. В нем выбрать пункт “My items”. Страница пользовательских объектов представлена на рисунке 3.14



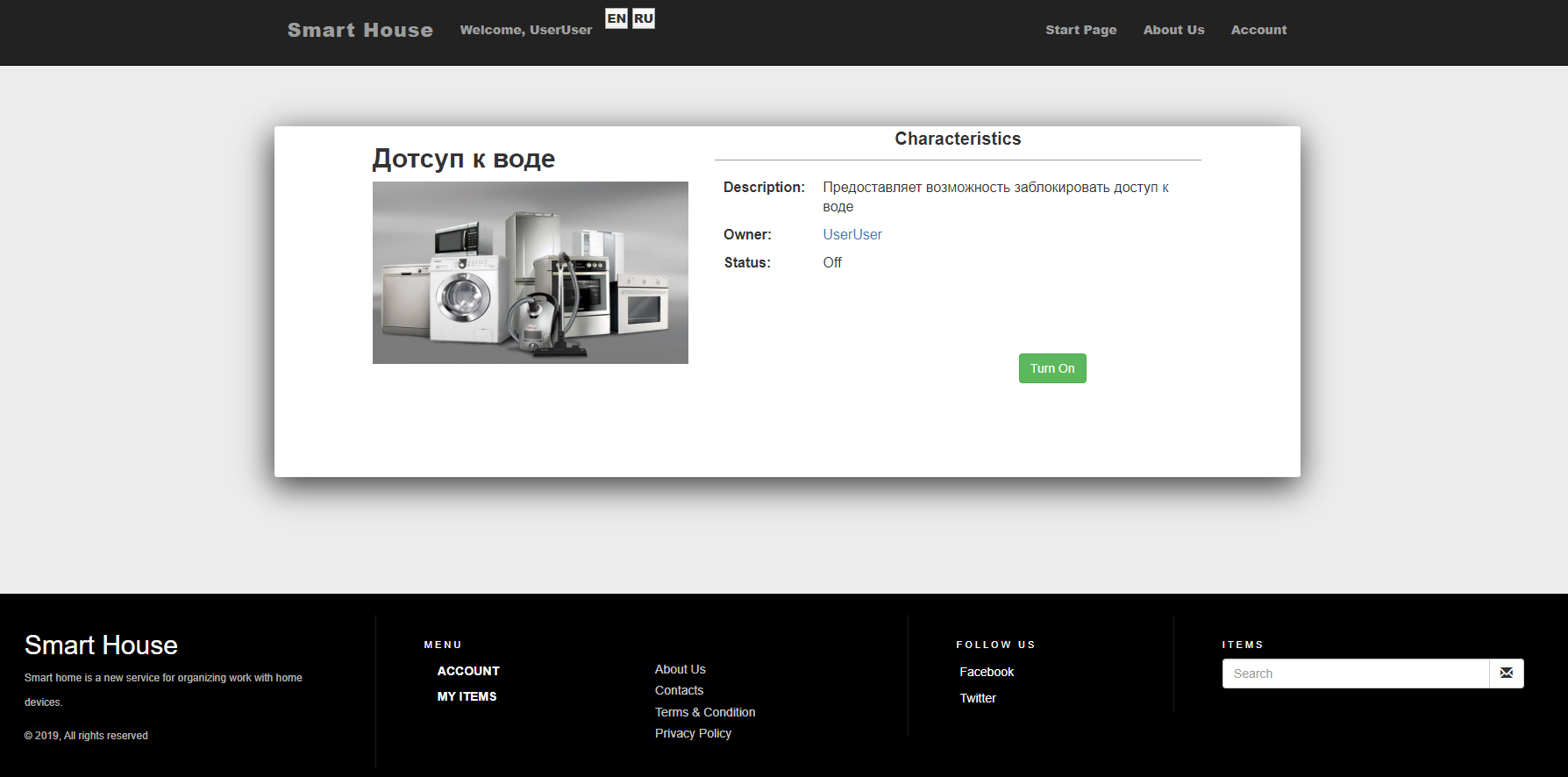
**Рисунок 3.14 - Страница пользовательских объектов**

После нажатия на кнопку “New Item” откроется окно создания объекта. Страница создания объекта предоставлена на рисунке 3.15.



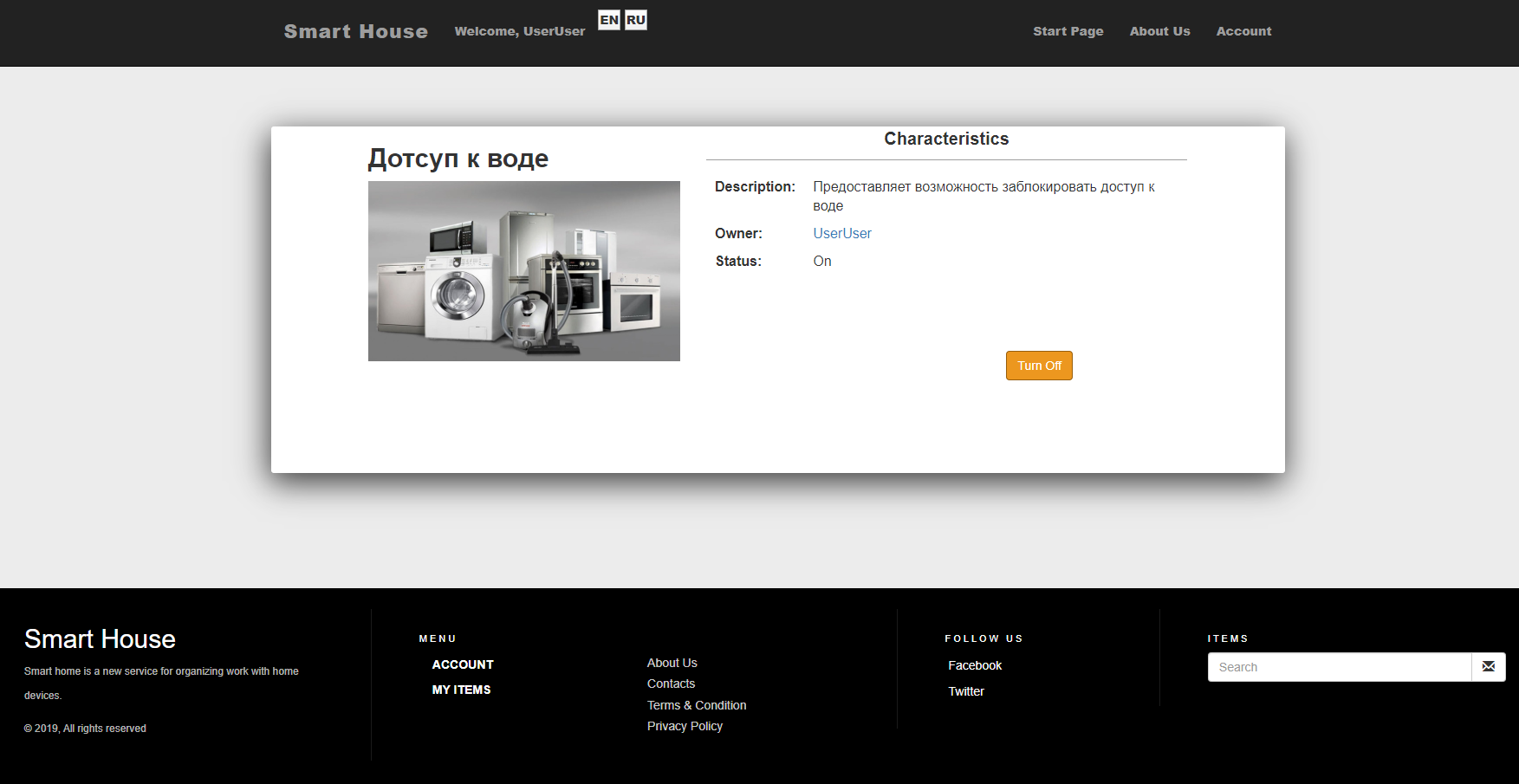
**Рисунок 3.15 - Страница создания объекта**

При нажатии на кнопку “Send” происходит обработка входных данных с последующим перенаправлением на страницу объекта. Страница созданного объекта предоставлена на рисунке 3.16.



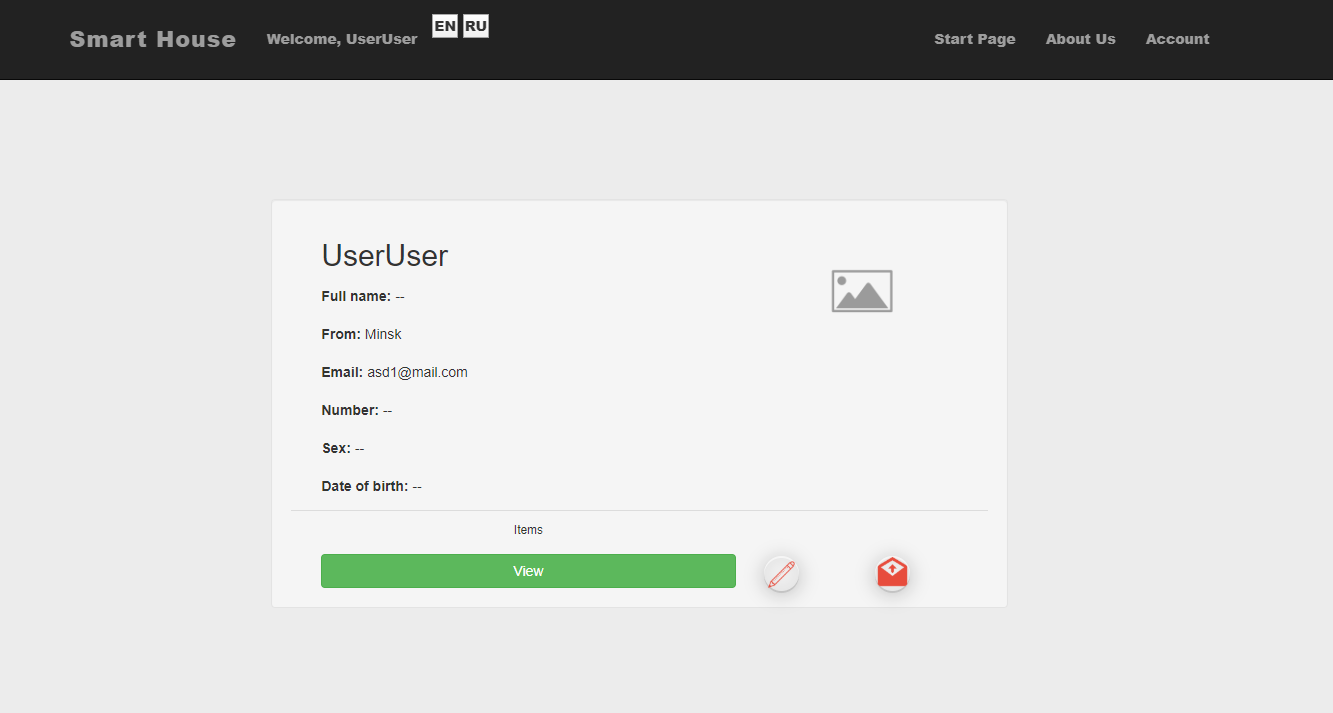
**Рисунок 3.16 - Страница отображения объекта**

На данной странице видно, что токующий объект находится в состоянии Off, т.е. выключен. При нажатии на кнопку “Turn On” происходит включение объекта. Результат продемонстрирован на рисунке 3.17.



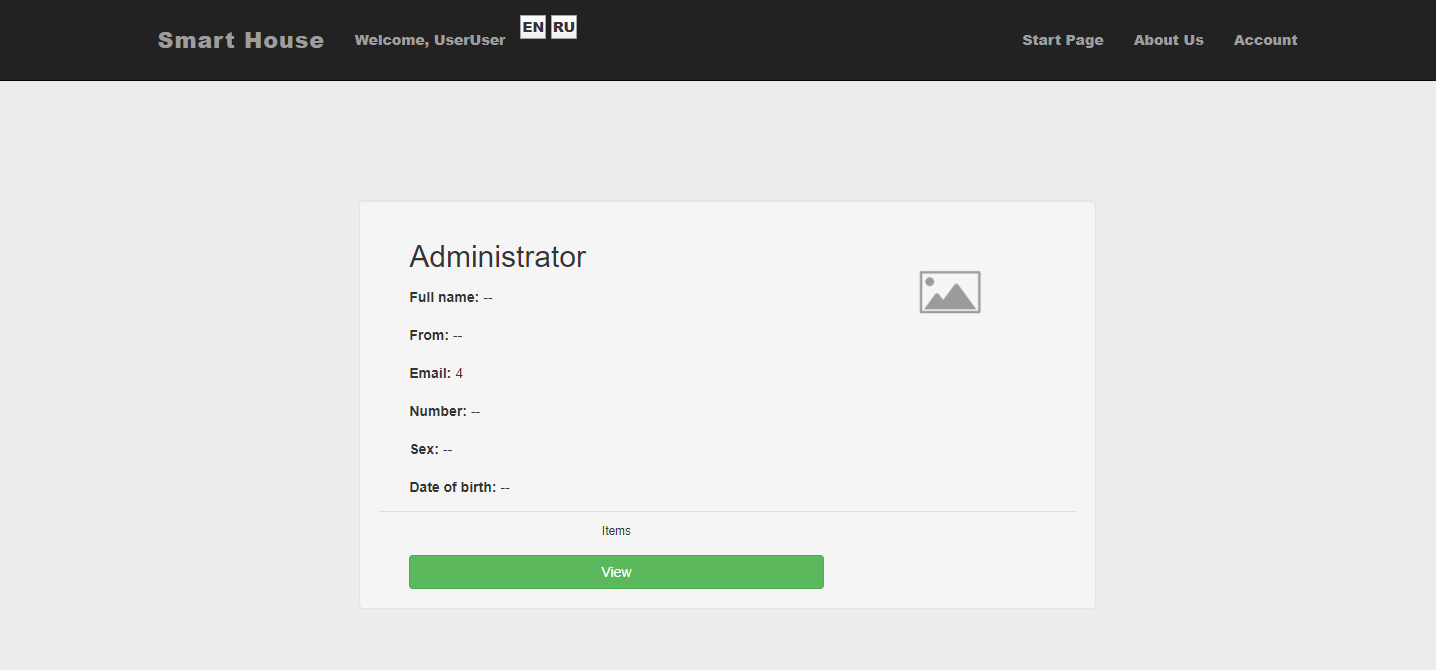
**Рисунок 3.17 - Страница включенного объекта**

При нажатии на кнопку «Account» в выпадающем списке Account, откроется страница пользователя, представленная на рисунке 3.18.



**Рисунок 3.18 – Страница своего аккаунта**

При переходе на страницу другого пользователя остается возможность только просмотреть добавленные им объекты. Возможность редактирования аккаунта и написания письма разработчикам пропадает. Страница внешнего пользователя представлена на рисунке 3.19.



**Рисунок 3.19 – Страница аккаунта другого пользователя**

ЗАКЛЮЧЕНИЕ

Таким образом в ходе работы рассмотрены некоторые из существующих систем домашней автоматизации: Ajax Systems, BroadLink, Xiaomi и Fibaro. Найдены преимущества и недостатки данных систем. На основании рассмотренных систем домашней автоматизации проведен сравнительный анализ, позволяющий комплексно оценить свойства систем.

Проведена работа по проектированию клиент-серверного приложения. Построена диаграмма вариантов использования, диаграмма последовательности, функциональная SADT диаграмма нулевого уровня, функциональная SADT диаграмма первого уровня детализации. Спроектирована база данных для хранения пользовательской информации и характеристик объектов автоматизации. Отображена физическая структура базы данных.

Реализовано и развернуто (на локальной машине) клиент-серверное веб приложение. Проведено базовое функциональное тестирование, отображающее основную функциональность реализованного приложения.

Таким образом, спроектировано и реализовано (подробности реализации https://github.com/IvanTkachev/Smart-House-Client) веб приложение, с которым будем взаимодействовать конечный пользователь. Данная система удовлетворяет сформированным требованиям: удобство в использовании, производительность, доступность, безопасность. Система регистрации и авторизации реализована посредством фреймворка Spring Security, пароли шифруются алгоритмом Bcrypt (аналогично дорогостоящей системе от компании Fibaro), а сессия пользователя защищена с помощью CSRF-токена.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Ротач, В.Я. Теория автоматического управления / В.Я. Ротач. – 2-е изд. – Москва : МЭИ, 2004. – 400 с.
2. Нейронные сети в системах управления процессами измельчения и обогащения руд черных металлов : материалы 3 Междунар. науч. конф, Казань, октябрь 2014 г. / Бук. – Казань, 2015. – 98 с.
3. Повышение точности путем многократных измерений / В.В. Денисенко // Современные технологии автоматизации. – 2009. – №1. – С. 96.
4. Аппаратное резервирование в системах промышленной автоматизации / В.В. Денисенко // Современные технологии автоматизации. – 2008. – №3. – С. 92–100.
5. Дианов, В.Н. Диагностика и надежность автоматических систем / В.Н. Дианов. – Москва : МГИУ, 2005. – 160 с.
6. Современные алгоритмы автоматического регулирования и их использование на предприятиях / Э.Л. Ицкович // Автоматизация в промышленности. – 2007. – №6. – С. 39–44.
7. Состав АСУ ТП // TEH-LIB.RU [Электронный ресурс]. – 2010. – Режим доступа : <http://www.teh-lib.ru/atpip/sostav-asu-tp/Tehnicheskoe-matematicheskoe-i-programmnoe-obespechenie-ASU-TP.html>. – Дата доступа : 13.10.2019.
8. GROVE - TEMP & HUMI & BAROMETER SENSOR (BME280) // Arduino [Электронный ресурс]. – 2019. – Режим доступа : <https://store.arduino.cc/grove-temp-humi-barometer-sensor-bme280>. – Дата доступа : 15.10.2019.
9. FEETECH MINI SERVO MOTOR 120 DEGREES 9G // Arduino [Электронный ресурс]. – 2019. – Режим доступа : <https://store.arduino.cc/feetech-mini-servo-motor-120-degrees-9g>. – Дата доступа : 15.10.2019.
10. ARDUINO UNO REV3 // Arduino [Электронный ресурс]. – 2019. – Режим доступа : <https://store.arduino.cc/arduino-uno-rev3>. – Дата доступа : 15.10.2019.
11. ARDUINO MKR ETH SHIELD // Arduino [Электронный ресурс]. – 2019. – Режим доступа : <https://store.arduino.cc/arduino-mkr-eth-shield>. – Дата доступа : 16.10.2019.
12. LCD1602 LCD KEYPAD SHIELD // arduinos.by [Электронный ресурс]. – 2019. – Режим доступа : <http://arduinos.by/shields/lcd-display>. – Дата доступа : 17.10.2019.
13. About MySQL // mysql.com [Электронный ресурс]. – 2019. – Режим доступа : <https://www.mysql.com/products/workbench>. – Дата доступа : 17.10.2019.
14. Дизайн итнерьера // planner5d.com [Электронный ресурс]. – 2019. – Режим доступа : <https://planner5d.com/ru/> – Дата доступа : 18.10.2019.
15. Предложения Office // microsoft.com [Электронный ресурс]. – 2019. – Режим доступа : <https://www.microsoft.com/ru-by/store/collections/alloffice?cat0=devices&icid=CatNavAllOffice>. – Дата доступа : 18.10.2019.
16. About Apache Tomcat // tomcat.apache.org [Электронный ресурс]. – 2019. – Режим доступа : <http://tomcat.apache.org/> – Дата доступа : 18.10.2019.
17. Arduiono IDE // Arduino [Электронный ресурс]. – 2019. – Режим доступа : <https://www.arduino.cc/en/main/software>. – Дата доступа : 19.10.2019.
18. О нас // Ajax Systems [Электронный ресурс]. – 2019. – Режим доступа : <https://ajax.systems/ru/about/>. – Дата доступа : 21.10.2019.
19. Как работает Ajax // Ajax Systems [Электронный ресурс]. – 2019. – Режим доступа : <https://ajax.systems/ru/how-it-works/>. – Дата доступа : 21.10.2019.
20. Продукты Ajax // Ajax Systems [Электронный ресурс]. – 2019. – Режим доступа : <https://ajax.systems/ru/products/#block2>. – Дата доступа : 21.10.2019.
21. Почему Ajax // Ajax Systems [Электронный ресурс]. – 2019. – Режим доступа : <https://ajax.systems/ru/why-ajax/>. – Дата доступа : 21.10.2019.
22. Стартовые комплекты // Ajax Systems [Электронный ресурс]. – 2019. – Режим доступа : <https://ajax.systems/ru/products/> – Дата доступа : 21.10.2019.
23. О BroadLink // broadlink.com [Электронный ресурс]. – 2019. – Режим доступа : <https://broadlink.com.ua/pro-kompaniyu/>. – Дата доступа : 24.10.2019.
24. Что такое Умный Дом BroadLink // broadlink.ru [Электронный ресурс]. – 2019. – Режим доступа : <https://broadlink.ru/chto-takoe-umniy-dom-broadlink>. – Дата доступа : 24.10.2019.
25. Устройства BroadLink // aliexpress.com [Электронный ресурс]. – 2019. – Режим доступа : <https://ru.aliexpress.com/popular/broadlink-smart-home.html>. – Дата доступа : 24.10.2019.
26. E-Control // play.google.com [Электронный ресурс]. – 2019. – Режим доступа : <https://play.google.com/store/apps/details?id=com.broadlink.rmt&hl=ru>. – Дата доступа : 24.10.2019.
27. О компании Xiaomi // mi-shop.by [Электронный ресурс]. – 2019. – Режим доступа : <https://mi-shop.by/about/>. – Дата доступа : 26.10.2019.
28. Комплект Умный дом Mi Smart Home Kit // mi.by [Электронный ресурс]. – 2019. – Режим доступа : <https://mi.by/catalog/umnyy_dom_1/komplekt_umnyy_dom_mi_smart_home_kit/>. – Дата доступа : 26.10.2019.
29. О компании Fibaro // mi.by [Электронный ресурс]. – 2019. – Режим доступа : <https://www.fibaro.com/ru/about-us/>. – Дата доступа : 28.10.2019.
30. Почему Fibaro // mi.by [Электронный ресурс]. – 2019. – Режим доступа : <https://www.fibaro.com/ru/why-fibaro/>. – Дата доступа : 28.10.2019.
31. Устройства Fibaro // z-wave.by [Электронный ресурс]. – 2019. – Режим доступа : <https://z-wave.by/product-category/po-proizvoditeljam/fibaro/>. – Дата доступа : 28.10.2019.
32. Member Companies of the Z-Wave Alliance // z-wavealliance.org [Электронный ресурс]. – 2019. – Режим доступа : <https://z-wavealliance.org/z-wave_alliance_member_companies/>. – Дата доступа : 28.10.2019.