

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ
федеральное государственное бюджетное образовательное учреждение
высшего образования
«Тольяттинский государственный университет»

Институт математики, физики и информационных технологий
Кафедра «Прикладная математика и информатика»

02.03.03 Математическое обеспечение и администрирование информационных систем

ТЕХНОЛОГИЯ ПРОГРАММИРОВАНИЯ

БАКАЛАВРСКАЯ РАБОТА

на тему Разработка мобильного и web-клиентов для информационной системы «Умный дом»

Студент _____ А.А. Каменский _____
Руководитель _____ А.В. Очеповский _____

Допустить к защите
Заведующий кафедрой к.тех.н, доцент, А.В. Очеповский _____

« _____ » _____ 2016 г.

Тольятти 2016

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ
федеральное государственное бюджетное образовательное учреждение
высшего образования
«Тольяттинский государственный университет»

Институт математики, физики и информационных технологий
Кафедра «Прикладная математика и информатика»

УТВЕРЖДАЮ
Зав. кафедрой «Прикладная
математика и информатика»
_____ А.В. Очеповский
« ____ » _____ 2016 г.

ЗАДАНИЕ
на выполнение бакалаврской работы

Студент Каменский Александр Александрович

1. Тема Разработка мобильного и web-клиентов для информационной системы «Умный дом»

2. Срок сдачи студентом законченной выпускной квалификационной работы 13.06.2016

3. Исходные данные к выпускной квалификационной работе: информация с датчиков аппаратного устройства, расположенная на сервере системы, требования к функциональным характеристикам: обеспечения канала связи с сервером по типу JSON, система аутентификации пользователя в системе, требование к средству реализации: HTML, PHP5, MySQL, CSS, JavaScript.

4. Содержание выпускной квалификационной работы (перечень подлежащих разработке вопросов, разделов):

Введение

1. Глава 1. Анализ инженерного комплекса «Умный дом»

2. Глава 2. Планирование разработки клиентской части системы «Умный дом»

3. Глава 3. Реализация программных модулей web-клиента и Android приложения для взаимодействия с сервером системы «Умный дом»

Заключение

5. Ориентировочный перечень графического и иллюстративного материала презентация, включающая блок-схемы работы приложения, графики, диаграммы, экранные формы, демонстрирующие работоспособность программного продукта

6. Дата выдачи задания «11» января 2016 г.

Заказчик, менеджер по проектам
ООО «100 линий»

С.А. Мальцев

Руководитель выпускной
квалификационной работы

А.В. Очеповский

Задание принял к исполнению

А.А. Каменский

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ
федеральное государственное бюджетное образовательное учреждение
высшего образования
«Тольяттинский государственный университет»

Институт математики, физики и информационных технологий
Кафедра «Прикладная математика и информатика»

УТВЕРЖДАЮ
Зав. кафедрой «Прикладная
математика и информатика»
_____ А.В. Очеповский

« ____ » _____ 2016 г.

КАЛЕНДАРНЫЙ ПЛАН
выполнения бакалаврской работы

Студента Каменский Александр Александрович
по теме Разработка мобильного и web-клиентов для информационной системы
«Умный дом»

Наименование раздела работы	Плановый срок выполнения раздела	Фактический срок выполнения раздела	Отметка о выполнении	Подпись руководителя
Выбор и утверждение темы ВКР	14.01.2016	14.01.2016	Выполнено	
Анализ состояния вопроса	20.02.2016	20.02.2016	Выполнено	
Разработка алгоритма сбора и передачи информации с сервера для Android и Web клиента	15.03.2016	15.03.2016	Выполнено	
Программная реализация предложенных решений	29.03.2016	29.03.2016	Выполнено	
Апробация предложенных решений на реальных данных	30.04.2016	30.04.2016	Выполнено	
Оформление текста бакалаврской работы	5.05.2016	5.05.2016	Выполнено	
Подготовка презентации к выступлению	10.05.2016	10.05.2016	Выполнено	
Предварительная защита ВКР	20.05.2016	20.05.2016	Выполнено	

Корректировка ВКР согласно сделанным замечаниям	1.06.2016	1.06.2016	Выполнено	
Проверка ВКР в системе «Антиплагиат.ВУЗ»	16.06.2016	16.06.2016	Выполнено	
Сдача пояснительной записки ВКР и реализованного программного приложения	16.06.2016	16.06.2016	Выполнено	

Руководитель выпускной
квалификационной работы

_____ А.В. Очеповский

Задание принял к исполнению

_____ А.А. Каменский

Аннотация

Тема: «Разработка мобильного и webклиентов для информационной системы «Умный дом»»

Актуальность темы выпускной квалификационной работы связана со значительным распространением инженерного комплекса «Умный дом» и заключается в необходимости разработки рекомендаций по совершенствованию работы в рассматриваемой области.

Целью данной бакалаврской работы является разработка мобильного и webклиентов для информационной системы «Умный дом».

Для достижения поставленной цели в ходе работы были поставлены и решены следующие задачи:

- анализ существующих мобильных и web клиентов;
- проектирование мобильного и web клиентов для информационной системы «Умный дом»;
- реализация мобильного и web клиентов для информационной системы «Умный дом»;
- тестирование реализованных клиентов.

Работа состоит из введения, трех глав, заключения, списка использованной литературы и приложений.

В работе использованы современные системы и технологии проектирования мобильного и web клиентов: PHP, HTML, CSS, JavaScript, MySQL и др.

Первая глава посвящена обзору и анализу предметной области.

Вторая глава посвящена разработке мобильного и web клиентов для информационной системы «Умный дом».

Третья глава посвящена реализации и тестированию мобильного и web клиентов для информационной системы «Умный дом».

В заключении сформулированы основные выводы, которые были сделаны в процессе написания бакалаврской работы и описаны результаты практической реализации проделанной работы.

Выпускная квалификационная работа содержит пояснительную записку объемом 41 страниц, включая 24 рисунков, одну таблицу.

Оглавление

Введение.....	4
Глава 1 Анализ предметной области	6
1.1 Описание системы	6
1.2 Модель работы информационной системы AS-IS.....	10
1.3 Формирование требований к новой информационной системе .	13
1.4 Выбор метода для создания аппаратной части информационной системы «Умный дом»	15
Глава 2 Моделирование и проектирование информационной системы .	17
2.1 Общая архитектура системы «Умный дом»	17
2.2 Структура web сайта информационной системы «Умный дом»	19
2.3 Общая структура мобильного приложения «Умный дом»	20
Глава 3 Реализация аппаратной платформы для информационной системы «Умный дом».....	23
3.1 Выбор программных средств для разработки web сайта системы «Умный дом»	23
3.2 Выбор программных средств для разработки мобильного приложения системы «Умный дом»	29
3.3 Кодирование клиентской части системы «Умный дом»	30
3.4 Нагрузочное тестирования клиента информационной системы «Умный дом»	39
Заключение	41
Список используемой литературы	42
Приложение А Листинг кода файла create_user.php	44
Приложение Б Листинг кода файла login.php	45
Приложение В Листинг кода файла forgot_password.php.....	46
Приложение Г Листинг кода файла mail.php	47
Приложение Д Листинг кода авторизации android клиента	48

Введение

В настоящее время у множества компаний связанных с инновационными разработками в сфере IT и робототехники возникает потребность в разработке инженерного комплекса «Умный дом».

«Умный дом» — это высокотехнологичная система, позволяющая объединить все коммуникации в одну и поставить её под управление искусственного интеллекта, программируемого и настраиваемого под все потребности, и пожелания хозяина [1].

Целесообразность внедрения проектов «Умный дом» очевидна, она складывается из целого ряда факторов, в том числе экономических. Так, снижение расходов на электроэнергию может достигать 30%, на воду – 41%, на тепло – 50%. Из-за постоянно повышающихся тарифов на газ и электроэнергию вопрос экономии затрат на эксплуатацию становится наиболее актуальным.

По прогнозам экспертов, спрос на «умную» недвижимость будет расти по мере того, как данные инженерные комплексы будут совершенствоваться в расширении функционала аппаратной части, а также в улучшении своего программного обеспечения.

Часто подобные комплексы представляют собой различные аппаратные устройства, делающие акцент на функциональных характеристиках, предоставляя тем самым пользователю более широкий спектр применения конкретного устройства. Не редко разработчики данных комплексов уделяют гораздо меньше внимания системному подходу к комплексу «Умный дом» и безопасности личных данных своих пользователей.

Актуальность темы выпускной квалификационной работы связана со значительным распространением инженерного комплекса «Умный дом» и заключается в необходимости разработки рекомендаций по совершенствованию работы в рассматриваемой области.

Объектом данной выпускной квалификационной работы является инженерный комплекс «Умный дом» разрабатываемый по заказу ООО «100 Линий».

В качестве **предмета** выпускной квалификационной работы будет рассмотрена клиентская часть данного комплекса, выраженная в Web-сайте и мобильном приложении для операционной системы Android.

Целью данной выпускной квалификационной работы является реализация мобильного приложения для аппаратной части инженерного комплекса «Умный дом», а также Web-сайт для данной системы, выявлению актуальных проблем, возникающих при организации системного подхода к инженерному комплексу «Умный дом».

Для достижения указанной цели поставлены следующие **задачи**: проанализировать существующие технические возможности для реализации Web-сайтов, а также Android приложений, выбрать на основании анализа наиболее подходящий способ реализации данных программных продуктов.

Выполнение данных задач позволило бы включить, реализуемую в ходе выполнения выпускной квалификационной работы, клиентскую часть инженерного комплекса «Умный дом» в единую систему, состоящую из нескольких схожих аппаратных блоков, нескольких пользователей и единого сервера, а также обеспечить сохранение данных в конфиденциальности. Что в свою очередь могло бы повысить популярность и спрос на данные инженерные комплексы в России.

В ходе выполнения выпускной квалификационной работы планируется выполнять анализ литературы и технической документации относительно создания Web-сайтов и мобильных приложений на базе Android с целью нахождения наиболее оптимальной для разработки программного обеспечения способного выполнять поставленные задачи. Выполнить систематизацию информации относительно обнаруженных данных с целью сравнить и выявить наиболее подходящий способ реализации данных задач. Произвести практическую работу и проанализировать полученные результаты.

На данный момент большинство существующих комплексов работают по системе передачи данных от одного конкретного устройства - одному конкретному пользователю, создавая ряд трудностей для компаний, предоставляющих услуги внедрения систем по типу «Умный дом», так и для самих пользователей обладающим несколькими устройствами в одном помещении или разных.

Практическая значимость данной выпускной квалификационной работы заключается в том, чтобы усовершенствовать инженерный комплекс «Умный дом», посредством реализации программного обеспечения, позволяющего включать данную аппаратную часть в систему с многими пользователями или устройствами и реализовать возможность пользователю или группе пользователей ознакомиться с данными предоставленными аппаратной частью, посредством Web-интерфейса или мобильного приложения.

Структура выпускной квалификационной работы обусловлена предметом, целью и задачами исследования. Работа состоит из введения, трёх глав, заключения и приложения.

Введение раскрывает актуальность, определяет степень научной разработки темы, объект, предмет, цель, задачи и методы исследования, раскрывает теоретическую и практическую значимость работы.

В первой главе производится анализ теоретической частью, с целью выборки требований к итоговому программному продукту.

Во-второй главе производится выборка средств реализации и технологий для выполнения поставленных задач программирования.

В третьей главе выполняется анализ произведённых реализаций и тестирование программного продукта.

В заключении подводятся итоги исследования, формулируются выводы по, рассматриваемой в ходе выполнения работы, по теме.

Глава 1 Анализ предметной области

1.1 Описание системы

Система «Умный дом» — это высокотехнологичная система, позволяющая объединить все коммуникации в одну и поставить её под управление искусственного интеллекта, программируемого и настраиваемого под все потребности, и пожелания хозяина [7].

Несмотря на широкое распространение систем «Умный дом» по всему миру, для большинства населения России эта инновация и по сей день остаётся исключительно термином. Медленное внедрение систем «Умный дом» объясняется многими факторами, один из которых – отсутствие понимания принципа работы подобных технологий и реально получаемой экономической выгоды. Безусловно, отдельные элементы «умных систем», такие как световые датчики движения или датчики протечки воды, уже давно не являются чем-то особенным и повсеместно распространены в домах и квартирах жителей таких мегаполисов, как Москва, Санкт-Петербург, Екатеринбург и Красноярск. Последнее время популярностью среди заказчиков стали пользоваться разнообразные системы безопасности, управления освещением, «мультирум» и «домашний кинотеатр».

Под архитектурой системы умного дома следует понимать совокупность модулей необходимых для её полноценной работы. Данные модули должны быть связаны между собой определённым образом, для обеспечения стабильной и корректной работы всей системы.

В данный момент системы «Умный дом» применяются в различных целях, они позволяют организовать эффективное управление системой освещения, отопления, водо- и электроснабжения, кондиционирования и вентиляции, видео наблюдения. Так же подобные системы позволяют собирать информацию о помещении в котором размещены с целью дальнейшей обработки и логистики.

Данная система могла бы помочь усовершенствовать практически все методы контроля за помещениями от частных загородных домов, офисов, до типовых многоквартирных домов.

В качестве возможного заказчика для готовой информационной системы «Умный дом», разрабатываемой в данной выпускной квалификационной работе, будет взята типовая организация Жилищно-коммунального хозяйства (в дальнейшем «ЖКХ»).

ЖКХ – это комплекс отраслей экономики, обеспечивающий функционирование всей инфраструктуры в различных поселениях [14]. Данный комплекс в частности занимается обслуживанием различных помещений и объектов города, в том числе и типовых многоквартирных домов. Более подробная структура организации ЖКХ представлена на рисунке 1.1.

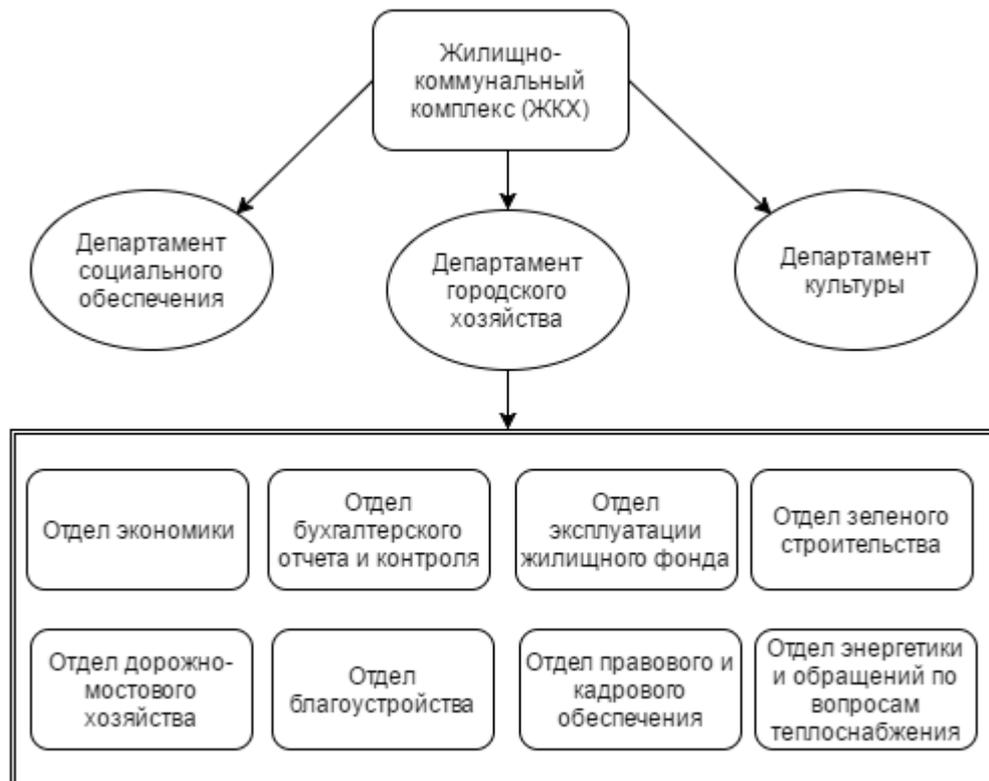


Рисунок 1.1 – Структура типового ЖКХ

В общем случае типовой ЖКХ выполняет следующие задачи:

— энергоснабжение – обеспечение обслуживаемых помещений соответствующим типом энергии (газ, электричество);

- теплоснабжение— обеспечение обслуживаемых помещений поставками горячей воды и тепловой энергии, обеспечение работы котельных;
- водоснабжение – прокладка, ремонт и обслуживание водопроводных труб, водозабор и очистка воды в многоквартирных домах;
- составление, хранение и использование цифровых карт города, включающих: сети коммунальных ресурсов и канализации, транспортные пути, здания, сети коммуникаций и др.;
- системы учета и расчетов за потребление коммунальных ресурсов и жилищно-коммунальных услуг, включая интернет порталы ГИС ЖКХ России в каждом регионе с личными кабинетами каждого пользователя ресурсами и услугами ЖКХ;
- вентиляция – централизованное кондиционирования воздуха и вентиляция;
- вывоз мусора;
- лифтовое хозяйство – обеспечение стабильной работы лифтов в обслуживаемых помещениях;
- капитальный ремонт и модернизация зданий;
- уборка дорог и содержание прилегающих к дому территорий.

ЖКХ выполняет, как приведённые в списке мероприятия по благоустройству жилых зон, так и множество других.

Следует рассмотреть более детально деятельность департамента городского хозяйства, в свою очередь, данный департамент занимается обеспечением реализации государственной и муниципальной политики в сфере городского хозяйства, направленной на обеспечение надежного, эффективного функционирования и развития жилищного хозяйства, объектов благоустройства, озеленения, сбора, вывоза, переработки и захоронения бытовых и промышленных отходов, сетей тепло, -водо, -газо, электроснабжения, водоотведения, обеспечение экологически благоприятных условий жизнедеятельности населения, в том числе рационального

природопользования, охраны окружающей среды в соответствии с общими целями, задачами и программами развития городского округа.

Данный департамент занимается следующими задачами в процессе своей работы:

1. Разработка и реализация стратегии развития городского хозяйства на основе современных технологий.
2. Формирование, организация исполнения и контроль за исполнением муниципального заказа на территории городского округа в вопросах энергоснабжения, благоустройства, жилищно-коммунальных услуг, озеленения, переработки и захоронения бытовых отходов.
3. Организация и контроль работы по развитию конкурентной среды предоставления жилищно-коммунальных услуг населению.
4. Разработка и реализация общей стратегии охраны окружающей среды, предусматривающей снижение негативного воздействия выбросов и выбросов вредных веществ в природную среду.
5. Нормативное правовое обеспечение природоохранной деятельности в пределах предоставленных полномочий.
6. Осуществление контроля за соблюдением природопользователями требований муниципальных правовых актов, касающихся порядка вывоза, утилизации и переработки отходов, а также мест размещения отходов производства и потребления.
7. Организация мероприятий по экологическому воспитанию, образованию и просвещению населения городского округа, экологическое информирование населения.
8. Организация контроля в области охраны окружающей среды на территории городского округа.

Исходя из данных задач, можно сделать вывод о том, что внедрение информационной системы «Умный дом» может благотворно сказаться на деятельности данного департамента. В частности внедрение данной информационной системы будет производиться на типовом отделе «Энергетики

и обращений по вопросам теплоснабжения». Данный отдел выбран в качестве наиболее подходящего подразделения департамента ЖКХ для внедрения разрабатываемой информационной системы.

1.2 Модель работы информационной системы AS-IS

В данный момент отдел «Энергетики и обращений по вопросам теплоснабжения» занимается помимо прочего, обеспечением многоквартирных домов электроэнергией, а также сбором информации о многоквартирных домах, для проведения соответствующих мероприятий по работам, связанным с терморегуляцией в многоквартирных домах.

На текущий момент система сбора и обработки температурных данных о домах происходит по следующим принципам, представленным на рисунке 1.2.

Диспетчерская



Рисунок 1.2 – Схема передачи данных «AS-IS»

В данной системе сбор данных происходит единственным контроллером и показания контроллера снимаются с основных источников передачи тепла в

многоквартирный дом. При достаточной простоте, данная система осуществляет температурный контроль по текущим нормативным документам и регламентациям. Но данная система имеет целый ряд недостатков вызывающих, как дискомфорт жильцов многоквартирного дома, так и нерациональное использования средств. Основными недостатками данной системы можно считать следующее:

- Отсутствие возможности контролировать распределение уровня тепла внутри различных помещений многоквартирного дома;
- недостаточно точный сбор статистики по помещению;
- отсутствие удобного для пользователя средства работы с собранной информацией.

Отслеживание температурного режима на узлах передачи тепла, не выявляет реальной температуры внутри квартир жильцов. В результате чего крайне частой является ситуация, когда у жильцов вызывает дискомфорт слишком сильное отопление при достаточно высокой температуре или же недостаточная степень обогрева помещений при заморозках.

Недостаточный сбор статистики в помещениях не позволяет собирать данные для более рационального распределения выделяемых ресурсов.

Недостаточно удобный для диспетчеров функционал работы с собранными данными, в текущей ситуации он может в лучшем случае наблюдать неисправность системы или её корректную работу, выраженную на текущий момент лишь приемлемым температурным уровнем главного узла.

На данный момент используется модель «как есть» на рисунке 1.1.

Таким образом будет создан новый web клиент с дополнительными функциями, а так же с доступом не только для сотрудников ЖКХ, отвечающих за это, но и обычным пользователям, у которых установлены данные датчики и они в режиме онлайн смогут посмотреть показания датчиков, статус работы датчиков, график изменения за месяц температуры, а так же связаться с сотрудниками ЖКХ для помощи.

1.3 Формирование требований к новой информационной системе

В основном, выделенные проблемы в модели которая используется в данный момент касается прибора снимающего показания в многоквартирном доме, но серьёзные нарекания вызывает и клиентская часть информационной системы, ведь в текущем виде она не предоставляет достаточно полной подробной информации о состоянии системы. Так же в ней отсутствует интуитивно понятный интерфейс и грамотное преподнесение собранных данных, поэтому для улучшения обнаруженных проблем в текущем виде системы следует изменить клиентскую часть. В текущий момент наиболее удачным решением может являться дополнение уже готовой системы – новой, которая в свою очередь, будет работать вместе со старой, дополняя нехватку данных получаемых ранее.

В качестве требуемого комплекса может использоваться информационная система «Умный дом», а именно её часть которая позволяет просматривать данные полученные от аппаратной части системы умный дом и проанализированной на сервере.

Принципиальное отличие от модели рассмотренной ранее, будет не только в создании новой клиентской части целиком, но и в том, что доступ к части данных будет и у сторонних пользователей, в частности владельцы квартир разместивших у себя аппаратную платформу системы «Умный дом».

Помимо получение конечных данных о температуре внутри жилой зоны, гибкость аппаратной части информационной системы «Умный дом» позволят собирать множество данных для статистики и при правильной организации

полученных данных можно будет сделать выводы для более рационального распределения выделяемых ресурсов на многоквартирный дом.

На основании проведённого выше анализа представляется возможным выдвинуть функциональные требования к аппаратной части разрабатываемой информационной системе:

- клиентская часть должна быть представлена мобильным клиентом для пользователей и web-сайтом для операторов ЖКХ;

- web сайт должен обладать двумя состояниями, для операторов и для владельцев квартир с размещёнными аппаратными платформами системы «Умный дом»

- web сайт должен обладать модулем регистрации пользователя;

- web сайт должен обладать системой аутентификации;

- web сайт должен обладать модулем обратной связи пользователя с технической поддержкой портала;

- web сайт должен обладать разделом web-представительства, где будет рассказано о назначении сайта;

- web сайт должен обладать функционалом для создания диаграмм и вывода показаниями в различных частях города (Должен быть реализован вывод данных в зависимости от выбранных параметров адреса)

- мобильный клиент должен открываться на мобильных устройствах под управлением системы Android;

- мобильный клиент должен содержать модуль авторизации;

- мобильный клиент должен демонстрировать статус работы аппаратной платформы и текущие показания датчиков.

В результате выдвинутых функциональных требований следует сделать выбор о том каким образом лучше осуществить подход для разработки аппаратной части для информационной системы «Умный дом».

В данный момент существует множество решений для создания Webсайтов, и может потребоваться сравнение данных методов для выбора оптимального по отношению к поставленным задачам.

1.4 Выбор метода для создания аппаратной части информационной системы «Умный дом»

В итоге следует сделать выбор о целесообразности реализации сайта посредством возможностей программного обеспечения для управления содержимым (ContentManagementSoftware, в дальнейшем CMS) или же написанием сайта на различных языках программирования с помощью Framework. Данное сравнение произведём посредством анализа в таблице 1.1

Таблица 1.1 – Сравнение методов улучшения клиентской части системы «Умный дом»

Требования к модулю информационной системы «Умный дом»	CMS	FrameWork
Написание логического аппарата сайта	Логический аппарат в CMS крайне тяжело внедрить, в основном он реализуется посредством специальных расширений.	При написании кода можно организовать практически любую логику, функционал языков JS и PHP позволяет это.
Подключение к сторонней БД	Все без исключения CMS вносят в БД сервера много лишних таблиц для стабильной работы общей системы CMS что может пагубно сказаться при внедрении сайта.	Забор данных и внесение правок можно организовать любым образом удобным для информационной системы.
Гибкость дополнительного функционала сайта	Дополнительный функционал реализуется либо посредством приобретения расширений либо написанием собственным с использованием специальных библиотек конкретной CMS.	Достаточная гибкость обеспечивают языки JS и PHP в совокупности с Framwork модулями.
Гибкость разработки основных функций сайта	Реализация основных функций возможно, но зачастую посредством отсутствия гибкости она будет в частичном виде.	Реализация вполне возможна, но требует знаний Web-программирования.

Цена разработки	Низкая цена разработки компенсируется покупкой дополнительных расширений предоставляемых в основном платно.	Цена разработки зависит от сложности реализуемых функций.
-----------------	---	---

В результате анализа приведённой таблицы следует сделать вывод о необходимости разработки клиентской части без использования CMS.

Таким образом можно сделать вывод о том, что текущая модель «как есть» имеет ряд недостатков, описанных ранее. Было принято решение о разработке новой информационной системы, были сформулированы требования к новой информационной системе, методы для создания ряда модулей новой информационной системы.

Глава 2 Моделирование и проектирование информационной системы

2.1 Общая архитектура системы «Умный дом»

Под архитектурой системы «Умный дом» следует понимать совокупность модулей необходимых для её полноценной работы. Данные модули должны быть связаны между собой определённым образом, для обеспечения стабильной и корректной работы всей системы. Структура информационной системы «Умный дом» представлена на рисунке 2.1.

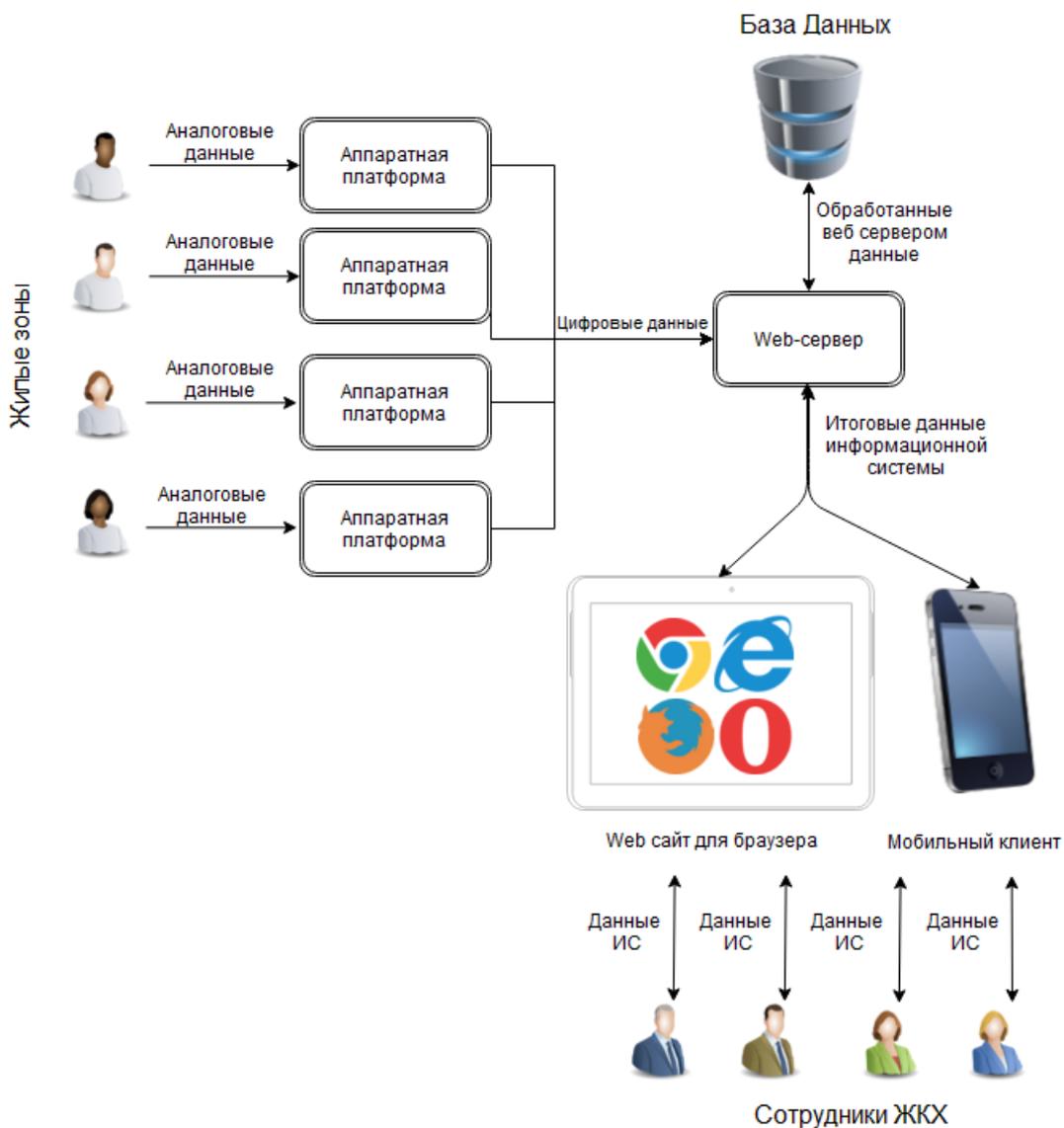


Рисунок 2.1 – Общая структура информационной системы «Умный дом»

Как видно из представленной структуры информационной системы «Умный дом» её можно условно разделить на три основных модуля: аппаратная платформа, сервер, клиент.

Каждый из этих модулей берёт на себя равную по значимости роль в работе всей информационной системы.

Аппаратная платформа представлена контроллером, размещенным в жилой зоне многоквартирного дома, обеспечивающим контроль итоговой температуры, вместо сбора температурных данных на узле в технической части здания. Точность сбора показаний будет зависеть от количества аппаратных платформ размещённых в доме, но даже одна подобная аппаратная платформа может помочь передать информацию о нерациональном теплообмене внутри многоквартирного дома.

В результате работы контроллера размещённого в жилой зоне происходит процесс перевода таких аналоговых данных, как температура или влажность в цифровые данные для их дальнейшей шифровки, формулировки исходящего пакета в требуемом для сервера виде и их дальнейшей передачи.

Web-сервер представлен программным продуктом, размещённом на удалённом сервере. Данный программный продукт обеспечивает приём входящих от аппаратной платформы данных. Расшифровку этих данных и заполнение полученной информации в базу данных сервера. Так же на сервере может производиться различный анализ данных с целью выявления проблем с теплообменом внутри многоквартирного дома или же анализ других полученных от аппаратной части данных в зависимости от их наличия.

Помимо описанных задач Web-сервер выполняет роль промежуточного звена между клиентской частью и данными прошедшими обработку сервером. Он обрабатывает запросы пришедшие с клиентской части и формирует данные для передачи формам веб-страниц или приложений.

Клиентская часть в информационной системе представлена сайтом выполняющим роль передатчика информации обработанной сервером – конечному пользователю. Так же сайт будет разделён на несколько областей видимости: для клиентов и для персонала. Мобильное приложения в свою очередь будет являться клиентом для конечного пользователя (владельца жилой зоны).

2.2 Структура web сайта информационной системы «Умный дом»

Под структурой web сайта информационной системы «Умный дом» следует понимать совокупность модулей необходимых для полноценной работы сайта. Данные модули должны быть связаны между собой определённым образом, для обеспечения стабильной и корректной работы всей системы. Структура web сайта информационной системы «Умный дом» представлена на рисунке 2.2.

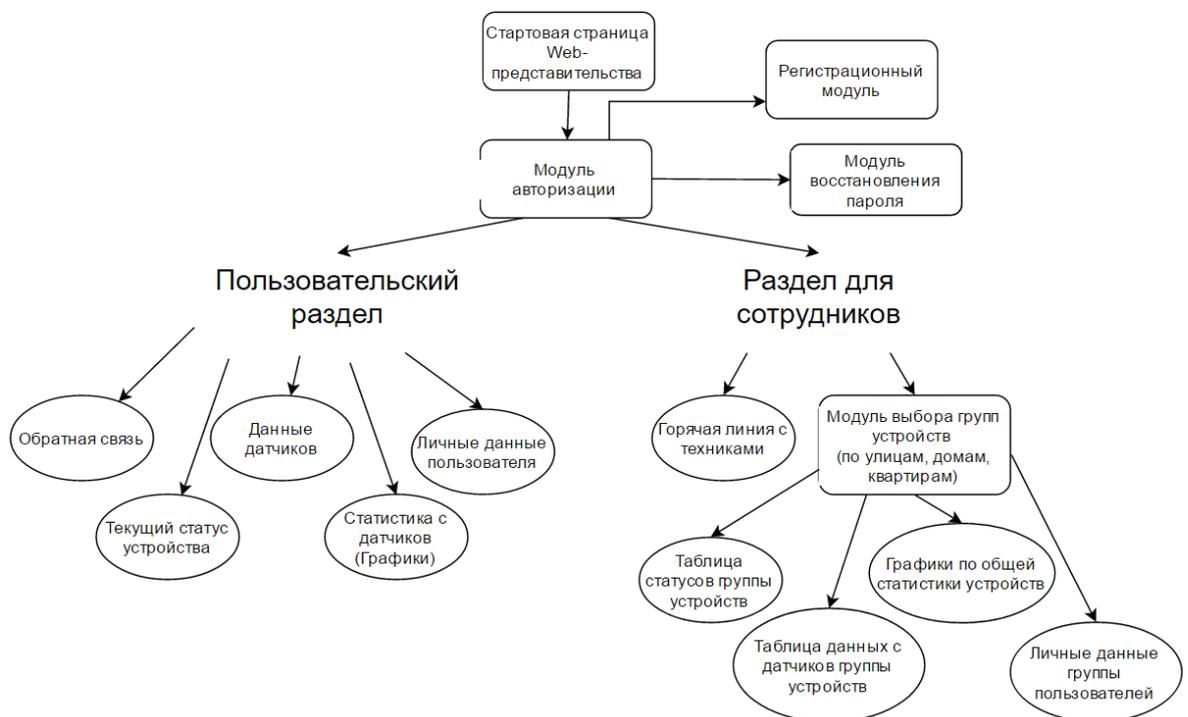


Рисунок 2.2 – Общая структура сайта «Умный дом»

Как видно на представленном рисунке после процедуры авторизации сайт делится на два состояния. На то, что видит пользователь-владелец устройства, и

то что будет видеть оператор ЖКХ. Пользователь может наблюдать лишь личные, приватные данные. Эти данные являются ничем иным, как данные принятые и обработанные веб-сервером от аппаратной платформы системы умный дом размещённой в квартире пользователя. Он может наблюдать как температурные данные, так и данные по влажности и движению непосредственно около аппаратной платформы. Так же все эти данные пользователь может вывести в виде графика, который отразит статистику – за это отвечает отдельный модуль работы сайта. Так же пользователь может наблюдать свои личные данные и вносить в них некоторые правки с целью отображения более точной статистики. В разработанной системе пользователь может так же связаться с сотрудником ЖКХ с целью подачи каких-либо обращений об отсутствии работы системы обогрева или других нарушениях в работе системы.

2.3 Общая структура мобильного приложения «Умный дом»

Под структурой мобильного приложения информационной системы «Умный дом» следует понимать совокупность модулей необходимых для полноценной работы сайта. Данные модули должны быть связаны между собой определённым образом, для обеспечения стабильной и корректной работы всей системы. Мобильное приложение, по большей степени, будет реализовано для конечного пользователя, для просмотра данных датчиков, а так же статуса активности датчиков. Конечный пользователь должен будет быть авторизован под своим логином и паролем, для того что бы увидеть данные датчиков со своей жилой зоны. В свою очередь Android клиент делает запрос на сервер в базу данных, для получения нужных данных для отображения. Структура мобильного приложения информационной системы «Умный дом» представлена на рисунке 2.3.

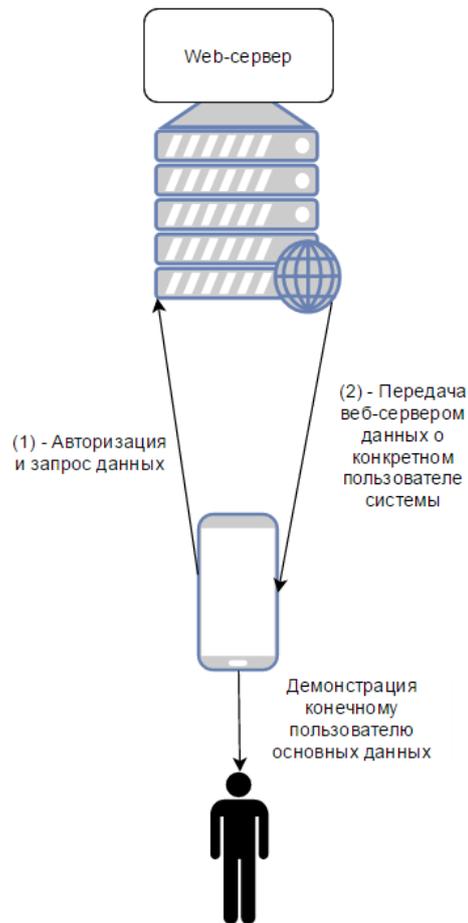


Рисунок 2.3 – Структура мобильного приложения

Как видно из представленной схемы, мобильное приложение работает независимо от работы сайта, не создавая тем самым дополнительную нагрузку на логический аппарат вебсайта. Android клиент запрашивает данные с сервера и отображает их в личном кабинете конечного пользователя, а так же у сотрудников ЖКХ.

На данный момент диаграмма классов Android приложения представлена на рисунке 2.4.

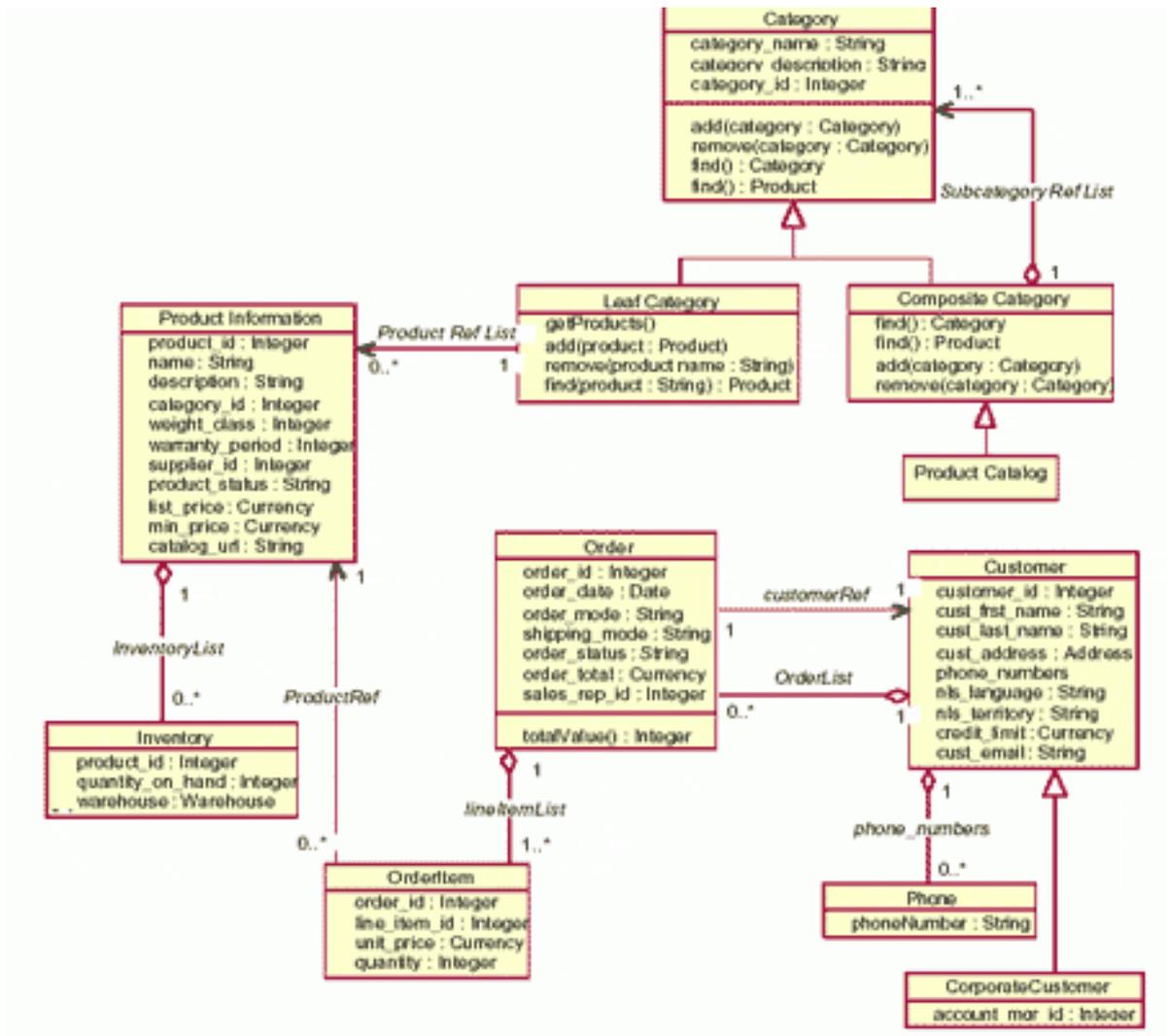


Рисунок 2.4 – Диаграмма классов Android приложения

На данной диаграмме классов представлена структура Android приложения, которая реализована на данный момент.

Рассмотрев выше изложенное в данной главе, мы смоделировали и спроектировали мобильное и web клиенты для информационной системы «Умный дом», которая будет использоваться в ЖКХ.

Глава 3 Реализация аппаратной платформы для информационной системы «Умный дом»

3.1 Выбор программных средств для разработки web сайта системы «Умный дом»

HTML5 – это язык, используемый для структурирования и отображения контента. Аббревиатура расшифровывается как HyperTextMarkupLanguage [9]. Являясь пятой версией HTML, в то же время он представляет собой не продолжение HTML4 или XHTML, а новый язык WebApplications 1.0. Знакомое название он носит в целях эффективности продвижения на рынок; по этой же причине в качестве его основы был выбран HTML.

Новая версия еще дорабатывается, однако в на сегодняшний день в браузерах частично реализована ее поддержка. Ее функционал обеспечивает следующие возможности:

- геолокация. Эта функция позволяет определять местонахождение пользователя и указывать его на карте, выводить информацию о находящихся поблизости точках обслуживания и других объектах, выстраивать маршруты;
- воспроизведение видео и аудио;
- локальное сохранение информации. Сайтам предоставляется возможность сохранять данные на локальном ПК и обращаться к ним в дальнейшем;
- поддержка фоновых вычислений (JavaScript запускается в браузере в фоновом режиме);
- возможность работы страниц в режиме оффлайн;
- инструменты рисования, позволяющие при помощи JavaScript создавать графические объекты (линии, фигуры), удобно видоизменять их, выполнять заливку градиентами.

Помимо этого, в формах появились новые элементы для таких параметров, как дата и время, поиск, выбор цвета и т.д. Изменения затронули и теги для разметки страниц – добавлены новые и исключены устаревшие, ряд тегов подвергся модификации. Верстка веб-страниц требует понимания изменений, присущих новому языку, и способа перевода на него страниц.

CSS – это язык, использующийся, как правило, для создания оформления HTML-документа. При помощи этого инструмента описывается внешний вид последнего, составленный с применением языка разметки. Аббревиатура расшифровывается как CascadingStyleSheets [20].

Несмотря на то, что рассматриваемый язык чаще всего используется для определения внешнего вида документа HTML, он применяется не только для вышеуказанного типа документов, но и для таких, как XHTML, XUL и SVG. Однако последние случаи находятся за пределами данной темы, поэтому они здесь не будут рассмотрены [16].

Этот инструмент был разработан для того, чтобы отделить логическую структуру веб-документа от его внешнего вида. Как известно, описание структуры осуществляется посредством HTML. То, какой вид обретет эта структура, определяется при помощи CSS.

Раздельное описание логической структуры и внешнего вида страницы обеспечивает гибкое управление ее внешним видом. Помимо этого, сокращаются повторения кода, появления которых невозможно было бы избежать при описании внешнего вида посредством HTML.

Рассматриваемый язык обеспечивает разработчику возможность использования для страницы и ее конкретных элементов разнообразных типов и размеров шрифта, выбора цвета элементов, установления интервалов между ними, размещения на странице отдельных блоков и пр.

При использовании данного средства программирования для определения внешнего вида HTML-документа необходимо установить взаимосвязь между ним и стилями. Иными словами, документ должен «получить информацию» о том, что его оформление будет осуществляться посредством CSS.

Существует ряд способов установления связи между документом и языком, информирующих браузер о необходимости применения стилей либо к странице целиком, либо к отдельным ее элементам.

Размещение таблиц стилей возможно как в самом документе, к которому будет применяться оформление, либо в файле с расширением .css. Последний является стандартным текстовым файлом и содержит инструкции, касающиеся внешнего оформления элемента, а также его расположения на странице. Кроме того, в файле содержатся примечания к приведенным инструкциям.

В CSS Frameworks продуманы многочисленные аспекты верстки, к примеру, возможность работы при различных разрешениях экрана, поддержка многих браузеров и т.д. При использовании этого инструмента верстальщик освобождается от необходимости концентрироваться на них; ему нужно лишь указать, что надлежит отображать, и задать параметры отображения. Всю остальную работу выполняют Frameworks. Это существенно сокращает сроки верстки. Bootstrap широко распространен, что является дополнительным плюсом – поддержка вашего кода окажется проще для других верстальщиков.

Однако использование Frameworks имеет и минус – это нагрузка страницы лишними стилями, которые будут присутствовать и в том случае, если они не используются. Данный инструмент прекрасно подходит для создания прототипов и конечных вариантов страниц, для которых дизайн является второстепенным. В качестве примера можно привести страницы администрирования.

При наличии сугубо специфичного дизайна использование Frameworks для его верстки может породить больше сложностей, чем использование «родных» средств; однако этот вариант все же допустим.

Разработка Bootstrap осуществлялась с использованием LESS – динамического языка, предназначенного для стилевого оформления. Его разработчиком является Алексис Селлиер. Этот язык ускорил процесс разработки, а также привнес в него больше простоты и позитивности.

Сегодня существует ряд способов использования стилей Bootstrap. Новым пользователям, не работающим с LESS, сам Bootstrap советует загрузить скомпилированный Bootstrap с официального портала и поместить его в свой проект, не производя никаких изменений. После этого следует создать пустой CSS-файл и выполнить его подключение после bootstrap.css.

При использовании LESS хранение всех переменных Bootstrap осуществляется в файлах с расширением .less. При необходимости разработчик осуществляет их компиляцию в формат CSS (ручным или автоматическим способом). В HTML подключаются только скомпилированные файлы .css. Это вариант обеспечивает наибольшую гибкость.

PHP уже много лет является одним из лидеров по популярности среди языков программирования и веб-разработки. Его название означает HypertextPreProcessor (на русском это будет звучать примерно как «преобработчик гипертекста») [2-3]. Чтобы объяснить это понятие, следует упомянуть о том, что существует 2 типа языков – клиентский и серверный [5-6]. Примером первого типа является JavaScript. Эта группа языков функционирует в браузере пользователя. Выполнение всех действий и команд, задаваемых, к примеру, на JavaScript, осуществляется браузером. (Здесь, для простоты, не рассматривается серверный JavaScript.) Следовательно, обработка одного и того же кода, составленного разработчиком, будет происходить в том браузере, который используется конкретным пользователем, посещающим эту страницу. Браузер также именуется клиентом.

PHP относится ко второму типу языков. В случае с этой группой ситуация иная. Отдельный сайт всегда располагается на сервере – компьютере высокой производительности, специально отведенном для хранения множества сайтов. Выполнение команд и скриптов, созданных на PHP, происходит именно на нем. После выполнения скрипта PHP сервер передает браузеру результаты операции. Последний представляет ее в удобном для пользователей виде [11].

Если речь идет о PHP-странице, то здесь присутствуют два этапа. Сперва PHP-код выполняется обработчиком (PHP-интерпретатором); результатом этой

операции является простой HTML-код. Затем последний, в свою очередь, обрабатывается в браузере. Этот этап аналогичен тому, что имеет место в случае с HTML-страницей, однако для нее он является единственным [13].

Следует отметить отличную совместную работу PHP и HTML. Помимо этого, в код HTML допускается вставлять PHP-код, а при помощи PHP возможно выводить разметку HTML.

CodeInteger представляет собой готовый каркас для приложений, разрабатываемых с его помощью. Он включает в себя наиболее распространенные библиотеки. Создание новых приложений с помощью этого универсального инструмента избавляет от излишних сложностей – нет необходимости думать о том, как распределять его основные модули, структурировать директории и т.д [12].

У этого средства имеется несколько существенных преимуществ, которые выделяют его из ряда остальных фреймворков. К ним относятся:

- использование модели MVC, с успехом применяющейся при создании приложений различной специфики;
- поддержка большого количества БД;
- подробное и понятное руководство с примерами, позволяющее в непродолжительное время изучить этот инструмент;
- высокая скорость работы – его скорость генерации страниц считается эталонной.

Во введении документации отражены дополнительные сильные стороны данного инструмента.

JavaScript – это язык программирования, чаще всего применяющийся в браузерах для того, чтобы обеспечить интерактивность страниц. Однако это не единственная задача, решаемая с его помощью. Если же рассматривать именно ее, то главной функцией JavaScript можно назвать управление элементами DOM-модели страницы [18-19].

DOM представляет собой объективную модель документа. По-английски это название звучит как DocumentObjectModel. В соответствии с ней документ

(к примеру, веб-страницу) можно представить как дерево объектов, которым присущи свойства, позволяющие выполнять с ним определенные действия – получать, изменять и удалять узлы, а также изменять связи между последними. Выполнение этих действий с элементами страницы возможно при помощи JavaScript.

Добавление кода JavaScript на страницу осуществляется при помощи тега `script`. Его лучше всего размещать внутри тега `head`, однако это не обязательное условие.

Один документ может содержать неограниченное количество контейнеров `script`. В то же время нет необходимости указывать атрибут `«type='text/javascript'»`, поскольку значение `javascript` присутствует по умолчанию.

Этот язык дает возможность активно видоизменять структуру страницы и управлять ее элементами. Он широко применяется при создании анимации, визуальных эффектов (к примеру, увеличение, уменьшение, перемещение и др.). Также с его помощью создаются галереи фотографий и ротаторы контента, обеспечивающие последовательный показ установленных объектов (например, изображений). Язык нередко применяется и при проверке данных, вводимых пользователем в форму.

Инструмент содержит специальные библиотеки, предоставляющие кросс-браузеры интерфейс к методам DOM. Это дает пользователю возможность использования функций, созданных в определенной библиотеке, и обретать нужный результат, совместимый с различными браузерами. Помимо этого, используя библиотеки, разработчики избавляются от необходимости подробного изучения JavaScript и обретают удобные средства управления объектной моделью документа.

Highcharts

Highcharts представляет собой библиотеку, созданную на языке JavaScript. Ее предназначением является создание чартов; с ее помощью можно вставлять в сайт или веб-приложение интерактивные графики, содержащие

анимацию. В чартах реализована поддержка большого количества видов диаграмм – круговых, линейных, колоночных и т.д.

Чарты совместимы со всеми распространенными браузерами, в том числе с Safari. Работа с InternetExplorer возможна, начиная с 6 версии. Чарты доступны на бесплатной основе при частном (некоммерческом) использовании. Желающим использовать их в целях бизнеса они обойдутся как минимум в 80 долларов.

Различные режимы вывода графиков на странице можно комбинировать друг с другом. Любой из них можно отключить в реальном времени. Данные для создания графиков можно брать как из JavaScript, так и из файлов, хранящихся на ПК или сервере либо из базы данных.

Преимущества чартов:

- простота вывода, доступность использования для лиц, не имеющих опыта в веб-разработке;
- увеличение конкретных областей;
- возможность использования скинов и тем оформления;
- поддерживаются tooltip, X-ось date-time;
- малый вес – 18 кб.

Таким образом, библиотека Highcharts будет хорошим выбором подходить для построения графиков на web сайте, для отображения изменения температуры за целый месяц.

3.2 Выбор программных средств для разработки мобильного приложения системы «Умный дом»

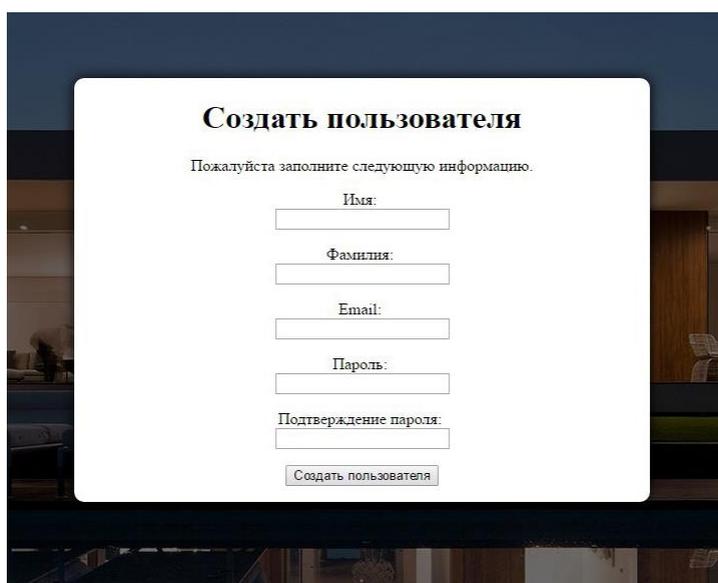
В ходе выполнения бакалаврской работы для написания Android приложения клиентской части для информационной системы «Умный дом», будет использоваться язык программирования Java и среда разработки Eclipse IDE в котором будет установлен плагин расширения ADT (AndroidDevelopmentTools).

Android предлагает пользовательский плагин для Eclipse IDE, который называется AndroidDevelopmentTools (ADT) [10]. Этот плагин обеспечивает мощную интегрированную среду для разработки программы. Он расширяет возможности Eclipse, что позволяет быстро разрабатывать новые проекты Android, включая создание интерфейса приложения, отладку приложений, а также экспорт подписанных пакетов приложений (APKs) для распространения.

Так же в ходе разработки приложения будут применяться сторонние API в зависимости от функциональных требований к разрабатываемому модулю. Подробнее о средствах реализации под операционную систему Android будет рассказано в демонстрационных примерах написания мобильного клиента для информационной системы «Умный дом» [4].

3.3 Кодирование клиентской части системы «Умный дом»

Кодирование клиентской части системы умный дом, заключается в написании программного кода различных клиентских модулей информационной системы. В данном случае на рисунке 3.1 демонстрируется форма регистрации нового пользователя в системе, она предлагает пользователю корректно заполнить поля, для внесения его данных в базу данных web сервера.



Создать пользователя

Пожалуйста заполните следующую информацию.

Имя:

Фамилия:

Email:

Пароль:

Подтверждение пароля:

Рисунок 3.1 – Модуль регистрации пользователя

Программный код данной реализации представлен на рисунке 3.2.

```

<?php echo form_open("auth/create_user");?>

<p>
    <?php echo lang('create_user_fname_label', 'first_name');?> <br />
    <?php echo form_input($first_name);?>
</p>

<p>
    <?php echo lang('create_user_lname_label', 'last_name');?> <br />
    <?php echo form_input($last_name);?>
</p>

<?php
if($identity_column!='email') {
    echo '<p>';
    echo lang('create_user_identity_label', 'identity');
    echo '<br />';
    echo form_error('identity');
    echo form_input($identity);
    echo '</p>';
}
?>

<p>
    <?php echo lang('create_user_email_label', 'email');?> <br />
    <?php echo form_input($email);?>
</p>

<p>
    <?php echo lang('create_user_phone_label', 'phone');?> <br />
    <?php echo form_input($phone);?>
</p>

<p>
    <?php echo lang('create_user_password_label', 'password');?> <br />
    <?php echo form_input($password);?>
</p>

<p>
    <?php echo lang('create_user_password_confirm_label', 'password_confirm');?> <br />
    <?php echo form_input($password_confirm);?>
</p>

<p><?php echo form_submit('submit', lang('create_user_submit_btn'));?></p>
<?php echo form_close();?>

```

Рисунок 3.2 – Программный код для регистрации нового пользователя

Как видно из рисунка 3.3 форма авторизации является единой формой для всех групп пользователей клиентской части сайта. Она предлагает пользователю пройти процесс аутентификации в информационной системе.

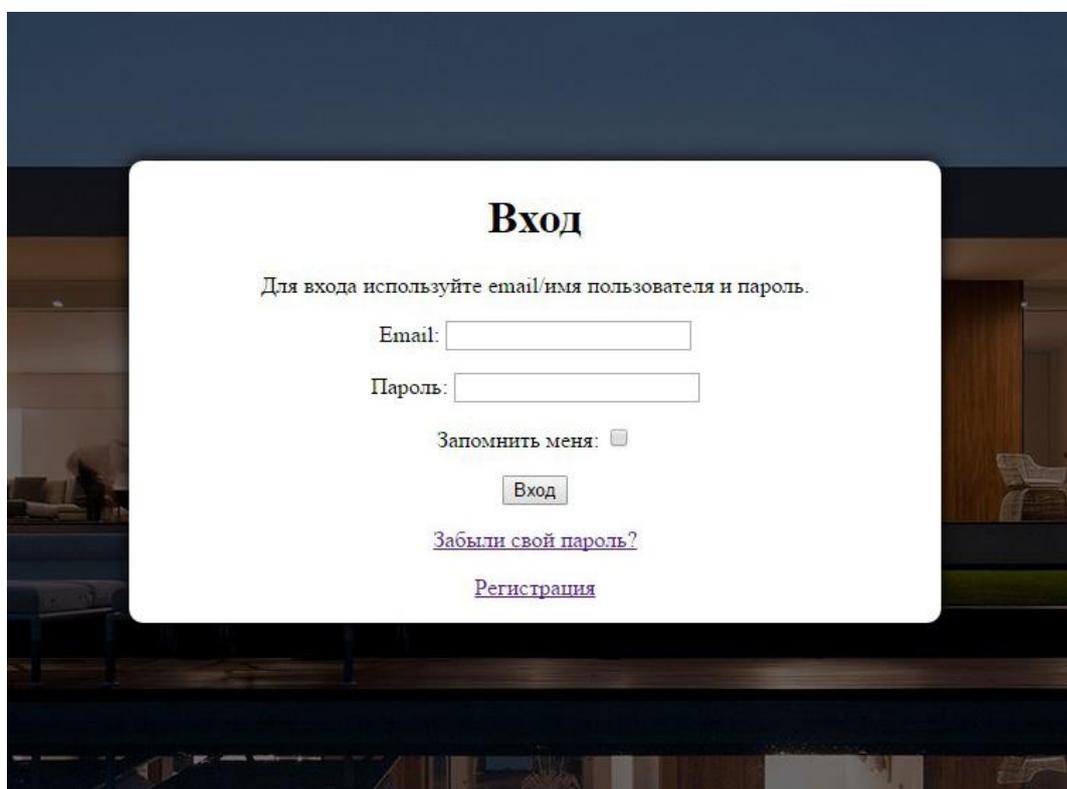


Рисунок 3.3 – Модуль авторизации пользователя

Программный код данной реализации представлен на рисунке 3.4.

```
<h1><?php echo lang('login_heading');?></h1>
<p><?php echo lang('login_subheading');?></p>

<div id="infoMessage"><?php echo $message;?></div>

<?php echo form_open("auth/login");?>

<p>
  <?php echo lang('login_identity_label', 'identity');?>
  <?php echo form_input($identity);?>
</p>

<p>
  <?php echo lang('login_password_label', 'password');?>
  <?php echo form_input($password);?>
</p>

<p>
  <?php echo lang('login_remember_label', 'remember');?>
  <?php echo form_checkbox('remember', '1', FALSE, 'id="remember"');?>
</p>

<p><?php echo form_submit('submit', lang('login_submit_btn'));?></p>

<?php echo form_close();?>

<p><a href="forgot_password"><?php echo lang('login_forgot_password');?></a></p> <p><a href="create_user">Регистрация</a></p>
```

Рисунок 3.4 - Программный код для авторизации пользователя

В случае если пользователь информационной системы забыл пароль от своего аккаунта, он может восстановить доступ к системе по ключевому полю электронной почты. Процесс восстановления пароля представлен на рисунке 3.5.

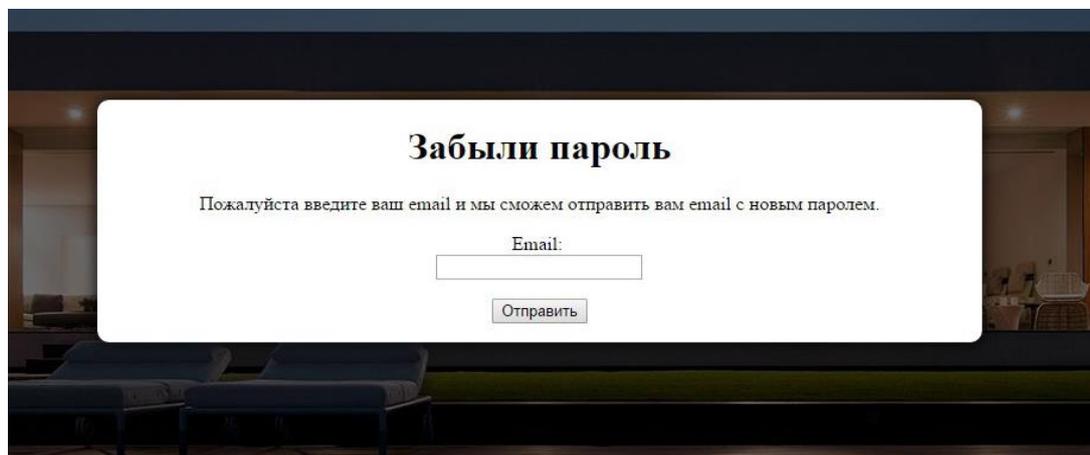


Рисунок 3.5 – Модуль восстановления пароля

Программный код данной реализации представлен на рисунке 3.6.

```

<?php echo form_open("auth/change_password");?>

<p>
  <?php echo lang('change_password_old_password_label', 'old_password');?> <br />
  <?php echo form_input($old_password);?>
</p>

<p>
  <label for="new_password"><?php echo sprintf(lang('change_password_new_password_label'), $min_password_length);?></label> <br />
  <?php echo form_input($new_password);?>
</p>

<p>
  <?php echo lang('change_password_new_password_confirm_label', 'new_password_confirm');?> <br />
  <?php echo form_input($new_password_confirm);?>
</p>

<?php echo form_input($user_id);?>
<p><?php echo form_submit('submit', lang('change_password_submit_btn'));?></p>

<?php echo form_close();?>

```

Рисунок 3.6 – Программный код модуля восстановления пароля

В созданной информационной системе, в клиентской части сотрудник ЖКХ может наблюдать общую статистику по обслуживаемым помещениям. У группы пользователей с правами сотрудников ЖКХ реализован доступ к сводной таблице с данными датчиков. Сотрудник может управлять отображаемым контентом. Данный процесс проиллюстрирован на рисунке 3.7.

Имя и Фамилия	Температура	Влажность	Статус датчика	Почта	Телефон	Адрес
Дмитрий Иванов	27.68 C	30.12 %	●	ivanov777@yandex.ru	8 (958) 324-65-81	ул. Приморский д. 37 кв. 1
Кузнецов Александр	28.11 C	30.00 %	●	kuznes1993@yandex.ru	8 (958) 694-68-88	ул. Приморский д. 44 кв. 2
Мария Веселова	28.00 C	29.18 %	●	petrov231@yandex.ru	8 (958) 963-95-14	ул. Степана Разина д. 72 кв. 5
Юлия Петренко	27.32 C	29.58 %	●	julia327@yandex.ru	8 (958) 966-12-78	ул. Юбилейная д. 56 кв. 9

Рисунок 3.7 – Раздел для сотрудников ЖКХ

Программный код данной реализации представлен на рисунке 3.8.

```
$query = $this->db->query("SELECT * FROM 'device'");
$info = $query -> result_array();
echo "<td>";
print_r($info);
echo "</td>";
```

Рисунок 3.8 – Программный код запроса показаний датчиков

Клиент же в свою очередь будет обладать лишь данными по своему конкретному устройству, или несколькими устройствами при их наличии, как это видно на рисунке 3.9.

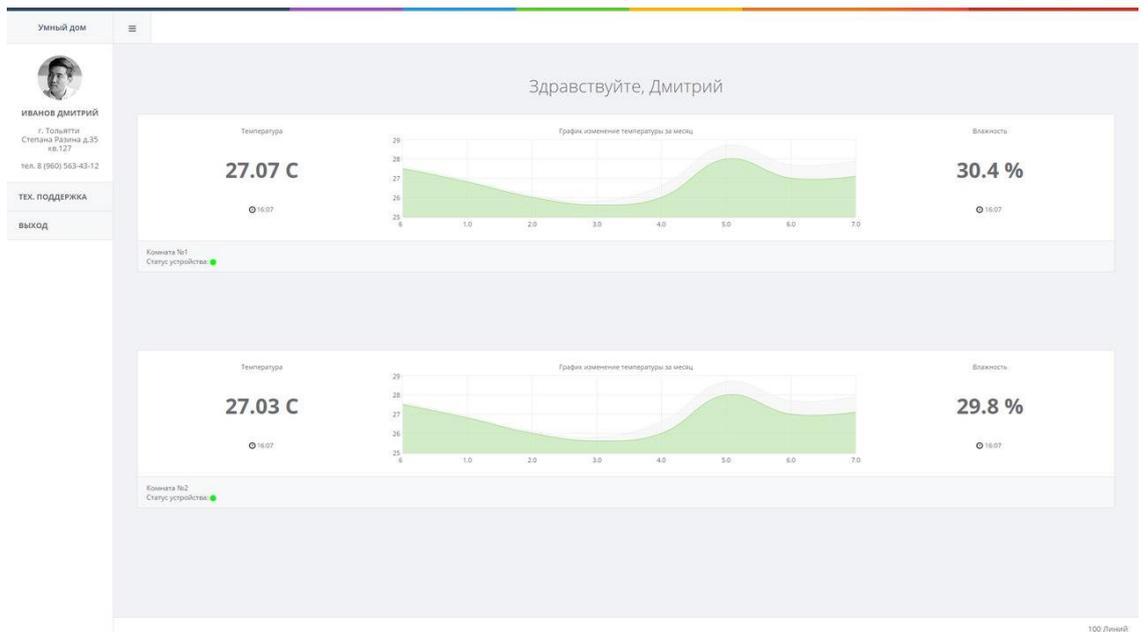


Рисунок 3.9 – Пользовательский раздел сайта

Программный код данной реализации представлен на рисунке 3.10.

```

$query = $this->db->query("SELECT time FROM `device`");
$date1 = $query->result_array();
$date2 = new DateTime( '2009-10-05 18:11:08' );

$diffInSeconds = $date2->getTimestamp() - $date1->getTimestamp();
if($diffInSeconds>'300')
    echo"<td></td>"
else
    echo"<td></td>"

```

Рисунок 3.10 – Программный код активности датчиков

Представитель любой группы пользователей будет иметь функционал обратной связи, но в случае с клиентом данный функционал позволяет отправить сообщение лишь консультанту ЖКХ, а сотрудник ЖКХ будет отправлять сообщения через форму обратной связи техническому специалисту, как это представлено на рисунке 3.11.

Рисунок 3.11 – Модуль обратной связи

Программный код данной реализации представлен на рисунке 3.12.

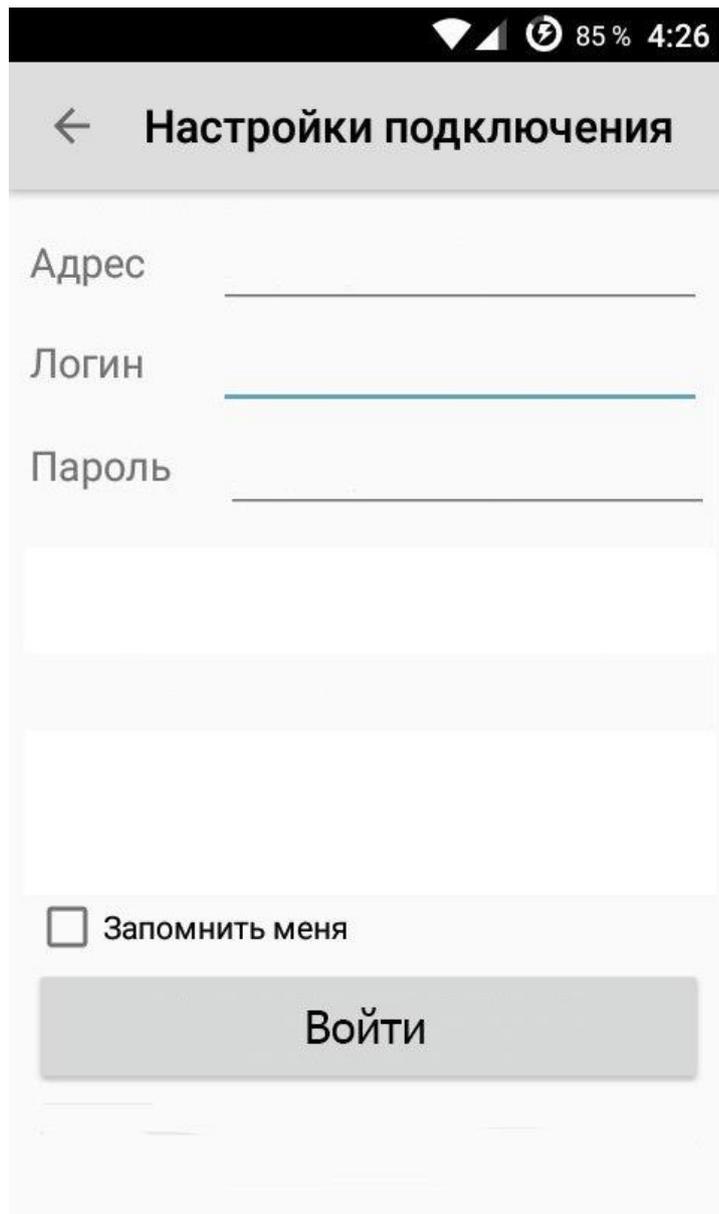
```

<meta charset="UTF-8">
<?
$desctext='Заявка от пользователя' + $name + $phone + $address";
$to = "support@zhkh.ru";
$headers = "Content-type: text/plain; charset = utf-8";
$subject = "Заявка с Вашего сайта zhkh";
$message = "С Вашего сайта shdiplom была отправлена заявка.\n" . $desctext ;
if(mail ($subject, $message, $headers)){
?>
<script type="text/javascript">
alert('Спасибо за заявку. Мы свяжемся с вами в ближайшее время.')
location.replace ("http://shdiplom.ru/");
</script>
<?>>

```

Рисунок 3.12 – Программный код отправки сообщения сотруднику ЖКХ

Мобильный клиент в свою очередь является удобным методом получения текущей информации по состоянию системы без выхода в браузер. Мобильный клиент в свою очередь обеспечивает большую степень интеграцию в аппаратную часть клиентского устройства, в данном случае смартфон [17]. На рисунке 3.13 виден функционал форма авторизации в мобильном приложении.



Настройки подключения

Адрес _____

Логин _____

Пароль _____

Запомнить меня

Войти

Рисунок 3.13 – Модуль авторизации

Программный код данной реализации представлен на рисунке 3.14.

```
URL url = new URL("https://www.shdiplom.com/tasks/users/lists?key=" + your_api_key);
URLConnection conn = (URLConnection) url.openConnection();
conn.addRequestProperty("client_id", your client id);
conn.addRequestProperty("client_secret", your client secret);
conn.setRequestProperty("Authorization", "OAuth " + token);
```

Рисунок 3.14 – Программный код авторизации пользователя мобильного приложения

Главное же меню программы представлено модулем вывода текущей информации по датчикам устройства. Тут же возможен, как выход из мобильного приложения, так и его детальная настройка. Эти данные можно наблюдать на рисунке 3.15.



Рисунок 3.15 – Модуль вывода данных

Программный код данной реализации представлен на рисунке 3.16.

```

try {
$db = new PDO (HOST_DBNAME, USERNAME, PASSWORD);

$stmt = $db->prepare ("SELECT * FROM `device`");

if($stmt->execute()) {
...
$arrToJSON = Array();

while($row = $stmt->fetch()) {
    $arrToJSON[JSON_OBJECTS][ ] = $row;
}

print json_encode($arrToJSON);

$db = null;

} catch (PDOException $e) {
    print "ERROR WITH PDO >> \n ".$e->getMessage();
}
?>

```

Рисунок 3.15 – Программный код запроса показаний датчика

В данном параграфе были разработаны и продемонстрированы небольшие куски кода мобильного и web клиентов для информационной системы «Умный дом».

3.4 Нагрузочное тестирования клиента информационной системы «Умный дом»

Нагрузочное тестирование (англ. loadtesting) — подвид тестирования производительности, сбор показателей и определение производительности и времени отклика программно-технической системы или устройства в ответ на внешний запрос с целью установления соответствия требованиям, предъявляемым к данной системе (устройству).

В ходе выполнения бакалаврской работы будет проверена клиентская часть системы «Умный дом» представленная вебсайтом. Тестирование будет производиться посредством нагрузочного теста. В свою очередь нагрузочный тест будет производиться по технологии JMETTER.

JMETER - это инструмент для проведения подобных тестов разработанный компанией ApacheSoftwareFoundation. Хотя изначально JMeter разрабатывался как средство тестирования web-сайтов, в настоящее время он способен проводить нагрузочные тесты для JDBC-соединений, FTP, LDAP, SOAP, JMS, POP3, IMAP, HTTP и TCP [15].

Интересна возможность создания большого количества запросов с помощью нескольких компьютеров при управлении этим процессом с одного из них. Архитектура, поддерживающая плагины сторонних разработчиков, позволяет дополнять инструмент новыми функциями.

В программе реализованы механизмы авторизации виртуальных пользователей, поддерживаются пользовательские сеансы. Организовано запись статистики результатов теста и разнообразная визуализация результатов в виде диаграмм, таблиц.

Результат тестирования web сайта программой Jmeter под нагрузкой в 500 запросов показан на рисунке 3.17.

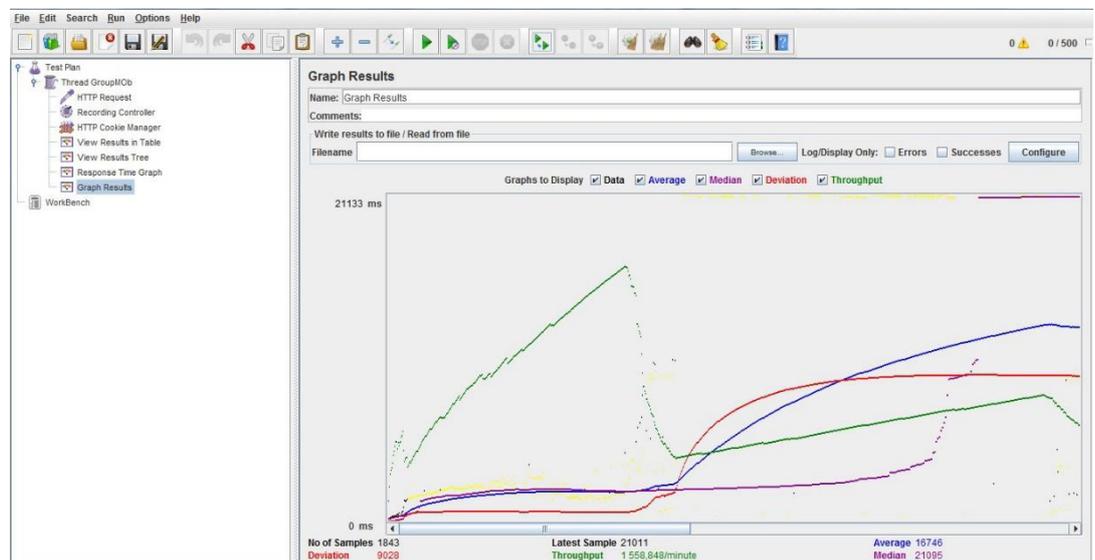


Рисунок 3.17 – Результат тестирования вебсайта программой Jmeter

Тестирование программой Jmeter под нагрузкой в 500 запросов на вебсайт, показал хороший результат подтверждает возможность комфортной работы с сайтом не менее 500 пользователей одновременно.

Заключение

В ходе выполнения выпускной квалификационной работы, был создан программный продукт для аппаратной части информационной системы умный дом. Данный программный продукт, предназначен, для организации сбора данных о жилом помещении с целью передачи этих данных защищённым шифровкой пакетом данных в типовую организацию ЖКХ. Данные предоставляемые созданным программным продуктом, могут позволить улучшить жилищные условия пользователей, а также оптимизировать выделения ресурсов со стороны ЖКХ.

Подобная система в дальнейшем может расширять свой функционал, расширяя тем самым актуальность применения данной системы в реальных условиях.

Даже в условиях выполнения бакалаврской работы удалось достичь сбора достаточно точных конечных температурных данных о жилом помещении.

Обеспечить должный уровень защиты передаваемой информации, а также предоставить подходящую для дальнейшей разработки аппаратную платформу с необходимым функционалом для реализации программной части, практически любой сложности относительно поставленных ранее задач.

В результате выполнения бакалаврской работы удалось выполнить все поставленные в техническом задании задачи и подтвердить знания материала преподаваемого в процессе обучения специальности математическое обеспечение и администрирование информационных систем.

Список используемой литературы

Учебники и учебные пособия

1. Авдеев, А. С. Основные проблемы программирования систем «Умного Дома» [Текст] / А. С. Авдеев, А. И. Герасимова // Перспективы науки. – 2014. – С. 62–65.
2. Веллинг, Л. Разработка Web–приложений с помощью PHP и MySQL [Текст] / Л. Веллинг, Л. Томпсон. – 3–е изд. – М.: «Вильямс». – 2013. – 880 с.
3. Колисниченко, Д.Н. PHP и MySQL. Разработка веб-приложений / Д.Н. Колисниченко. – СПб.: Изд-во «БХВ-Петербург», 2015. – 592с.
4. МакГрат, М. Программирование на Java для начинающих [Текст] / М. МакГрат. – М.: Эксмо. – 2016. – 192 с.
5. Маклафлин, Б. PHP и MySQL Исчерпывающее руководство [Текст] / перевод О. Сивченко – СПб: Веб–технологии, – 2014 – С. 114 – 158.
6. Никсон, Р. Создаем динамические веб-сайты с помощью PHP, MySQL, JavaScript / Р. Никсон; пер. с англ. Н. Вильчинский. – 2-е изд. – СПб.: Изд-во «Питер», 2013. – 560с.
7. Олейник, П.П. Корпоративные информационные системы: учеб. пособие / П. П. Олейник. – СПб.: Питер, 2012. – 176 с.
8. Ошероув, Р. Искусство автономного тестирования с примерами на C++, 2-е издание: учеб. пособие / Р. Ошероув. – ДМК Пресс, 2014. – 360 с.
9. Прохоренок, Н.А. HTML, JavaScript, PHP и MySQL. Джентльменский набор Web-мастера / Н.А. Прохоренок, В.А. Дронов. – 4-е изд. – СПб.: Изд-во «БХВ-Петербург», 2015. – 766с.
10. Сильвен, Р. Android NDK. Разработка приложений под Android на C/C++: учеб. пособие / Р. Сильвен. – ДМК Пресс, 2012. – 496 с.

Электронные ресурсы

11. Руководство по PHP [Электронный ресурс]: PHP: Руководство по PHP – Manual / группа документирования PHP. – Электрон. дан. – [2016]. – Режим доступа: <http://php.net/manual/ru/index.php>

12. Руководство по CodeIgniter [Электронный ресурс]: CodeInteger: Руководство по CodeInteger. – Электрон. Дан. – Режим доступа: <http://code-igniter.ru/>

13. Справочное руководство по MySQL [Электронный ресурс]: MySQLManual. – Электрон. дан. – Режим доступа: <http://www.mysql.ru/docs/man/>

14. Сафаров. Г.С. Жилищно-коммунальное хозяйство, 2016 // Википедия [Электронный ресурс]: Материал из Википедии — свободной энциклопедии: https://ru.wikipedia.org/wiki/Жилищно-коммунальное_хозяйство

15. Автор Неизвестен. Нагрузочное тестирование, 2016 // Википедия [Электронный ресурс]: Материал из Википедии — свободной энциклопедии: <https://ru.wikipedia.org/?oldid=76561075>

Литература на иностранном языке

16. Baron Schwartz, Peter Zaitsev, VadimTkachenko High Performance MySQL, 3rd Edition Optimization, Backups, and Replication. – California, USA, 2012 – pp. 88 – 96.

17. Williams, N. S. Professional Java for Web Applications [Text] / N. S. Williams – Indianapolis, Indiana, : John Wiley & Sons, Inc. – 2014. – 865 с.

18. Pro Spring MVC: With Web Flow (Expert's Voice in Spring) [Text] / M. Deinum, K. Serneels, C. Yates and other, – New York :Apress. – 2012. – 596 p.

19. Sattari, H. Spring Web Services 2 Cookbook Paperback [Text] / H. Sattari, .Kunjumohamed. – UK :Packt Publishing. – 2012.–322 p.

20. Schwartz, B. High Performance MySQL [Text] / B. Schwartz, V. Tkachenko V. – 3rd edition optimization. – California (USA) : Backups, and Replication. – 2012 – pp. 88 – 96.

Листинг кода файла create_user.php

```

1 <div style="background-image: url(/img/bg.png);">
2
3 <h1><?php echo lang('create_user_heading');?></h1>
4 <p><?php echo lang('create_user_subheading');?></p>
5
6 <div id="infoMessage"><?php echo $message;?></div>
7
8 <?php echo form_open("auth/create_user");?>
9
10 <p>
11     <?php echo lang('create_user_fname_label', 'first_name');?> <br />
12     <?php echo form_input($first_name);?>
13 </p>
14 <p>
15     <?php echo lang('create_user_lname_label', 'last_name');?> <br />
16     <?php echo form_input($last_name);?>
17 </p>
18 <?php
19 if($identity_column!='email') {
20     echo '<p>';
21     echo lang('create_user_identity_label', 'identity');
22     echo '<br />';
23     echo form_error('identity');
24     echo form_input($identity);
25     echo '</p>';
26 }
27 ?>
28 <p>
29     <?php echo lang('create_user_email_label', 'email');?> <br />
30     <?php echo form_input($email);?>
31 </p>
32 <p>
33     <?php echo lang('create_user_phone_label', 'phone');?> <br />
34     <?php echo form_input($phone);?>
35 </p>
36 <p>
37     <?php echo lang('create_user_password_label', 'password');?> <br />
38     <?php echo form_input($password);?>
39 </p>
40 <p>
41     <?php echo lang('create_user_password_confirm_label', 'password_confirm');?> <br />
42     <?php echo form_input($password_confirm);?>
43 </p>
44 <p><?php echo form_submit('submit', lang('create_user_submit_btn'));?></p>
45
46 <?php echo form_close();?>
47 </div>
48 </div>

```

Рисунок А.1 – Листинг сайта create_user.php

Листинг кода файла login.php

```
1 <h1><?php echo lang('login_heading');?></h1>
2 <p><?php echo lang('login_subheading');?></p>
3
4 <div id="infoMessage"><?php echo $message;?></div>
5
6 <?php echo form_open("auth/login");?>
7 <p>
8     <?php echo lang('login_identity_label', 'identity');?>
9     <?php echo form_input($identity);?>
10 </p>
11 <p>
12     <?php echo lang('login_password_label', 'password');?>
13     <?php echo form_input($password);?>
14 </p>
15 <p>
16     <?php echo lang('login_remember_label', 'remember');?>
17     <?php echo form_checkbox('remember', '1', FALSE, 'id="remember"');?>
18 </p>
19 <p><?php echo form_submit('submit', lang('login_submit_btn'));?></p>
20
21 <?php echo form_close();?>
22
23 <p><a href="forgot_password"><?php echo lang('login_forgot_password');?></a></p>
24 <p><a href="create_user">Регистрация</a></p>
25 </div>
26 </div>
27 </div>
```

Рисунок Б.1 – Листинг сайта login.php

Листинг кода файла forgot_password.php

```
1 <h1><?php echo lang('forgot_password_heading');?></h1>
2 <p><?php echo sprintf(lang('forgot_password_subheading'), $identity_label);?></p>
3
4 <div id="infoMessage"><?php echo $message;?></div>
5
6 <?php echo form_open("auth/forgot_password");?>
7
8 <p>
9 <label for="identity"><?php echo (($type=='email') ? sprintf(lang('forgot_password_email_label'),
10 $identity_label) : sprintf(lang('forgot_password_identity_label'), $identity_label));?></label> <br />
11 <?php echo form_input($identity);?>
12 </p>
13
14 <p><?php echo form_submit('submit', lang('forgot_password_submit_btn'));?></p>
15
16 <?php echo form_close();?>
17 </div></div></div>
```

Рисунок В.1 – Листинг кода forgot_password.php

Листинг кода файла mail.php

```
1 <meta charset="UTF-8">
2 <?
3     $phone = $_POST['phone'];
4     $name = $_POST['name'];
5     $desctext='Заявка с сайта';
6     if (isset($_POST['name'])) {$name = $_POST['name'];}
7     if (isset($_POST['tel'])) {$phone = $_POST['phone'];}
8     $to = "helpshdiplom@mail.ru";
9     $headers = "Content-type: text/plain; charset = utf-8";
10    $subject = "Заявка с Вашего сайта shdiplom.ru";
11    $message = "С Вашего сайта shdiplom.ru была отправлена заявка.\n"
12    . $desctext . "\nИмя пославшего:" . $name . "\nТелефон:" . $phone;
13    if(mail ($to, $subject, $message, $headers)){
14    ?>
15    <script type="text/javascript">
16    alert('Спасибо за заявку. Мы свяжемся с вами в ближайшее время.')
17    location.replace("http://shdopлом.ru");
18
19    </script>
20    <?>
```

Рисунок Г.1 – листинг сайта mail.php

Листинг кода авторизации android клиента

```

1  import java.io.IOException;
2  import java.net.Socket;
3  import java.security.KeyManagementException;
4  import java.security.KeyStore;
5  import java.security.KeyStoreException;
6  import java.security.NoSuchAlgorithmException;
7  import java.security.SecureRandom;
8  import java.security.UnrecoverableKeyException;
9  import java.security.cert.CertificateException;
10 import java.security.cert.X509Certificate;
11 import java.util.ArrayList;
12 import java.util.Arrays;
13 import javax.net.ssl.KeyManagerFactory;
14 import javax.net.ssl.SSLContext;
15 import javax.net.ssl.TrustManager;
16 import javax.net.ssl.TrustManagerFactory;
17 import javax.net.ssl.X509TrustManager;
18
19 import org.apache.http.conn.ssl.SSLSocketFactory;
20
21
22 public class AdditionalKeyStoresSSLSocketFactory extends SSLSocketFactory {
23     protected SSLContext sslContext = SSLContext.getInstance("TLS");
24
25     public AdditionalKeyStoresSSLSocketFactory(KeyStore keyStore) throws NoSuchAlgorithmException, KeyManagementException, KeyStoreException, UnrecoverableKeyException {
26         super(null, null, null, null, null, null);
27         KeyManagerFactory kmf = KeyManagerFactory.getInstance(KeyManagerFactory.getDefaultAlgorithm());
28         kmf.init(keyStore, "".toCharArray());
29         sslContext.init(kmf.getKeyManagers(), new TrustManager[]{new ClientKeyStoresTrustManager(keyStore)}, new SecureRandom());
30     }
31
32     @Override
33     public Socket createSocket(Socket socket, String host, int port, boolean autoClose) throws IOException {
34         return sslContext.getSocketFactory().createSocket(socket, host, port, autoClose);
35     }
36
37     @Override
38     public Socket createSocket() throws IOException {
39         return sslContext.getSocketFactory().createSocket();
40     }
41
42     public static class ClientKeyStoresTrustManager implements X509TrustManager {
43
44         protected ArrayList<X509TrustManager> x509TrustManagers = new ArrayList<X509TrustManager>();
45
46         protected ClientKeyStoresTrustManager(KeyStore... additionalKeyStores) {
47             final ArrayList<TrustManagerFactory> factories = new ArrayList<TrustManagerFactory>();
48
49             try {
50                 final TrustManagerFactory original = TrustManagerFactory.getInstance(TrustManagerFactory.getDefaultAlgorithm());

```

Рисунок Д.1 – листинг кода авторизации android клиента, часть 1

```

51     original.init((KeyStore) null);
52     factories.add(original);
53
54     for ( KeyStore keyStore : additionalkeyStores ) {
55         final TrustManagerFactory additionalCerts = TrustManagerFactory.getInstance(TrustManagerFactory.getDefaultAlgorithm());
56         additionalCerts.init(keyStore);
57         factories.add(additionalCerts);
58     }
59
60     } catch (Exception e) {
61         throw new RuntimeException(e);
62     }
63
64
65     for (TrustManagerFactory tmf : factories)
66     for ( TrustManager tm : tmf.getTrustManagers() )
67     if (tm instanceof X509TrustManager)
68         x509TrustManagers.add( (X509TrustManager) tm );
69
70     if ( x509TrustManagers.size() == 0 )
71         throw new RuntimeException("Couldn't find any X509TrustManagers");
72
73 }
74
75 public void checkClientTrusted(X509Certificate[] chain, String authType) throws CertificateException {
76     for ( X509TrustManager tm : x509TrustManagers ) {
77         try {
78             tm.checkClientTrusted(chain, authType);
79             return;
80         } catch ( CertificateException e ) {
81
82         }
83     }
84     throw new CertificateException();
85 }
86
87
88 public void checkServerTrusted(X509Certificate[] chain, String authType) throws CertificateException {
89     for ( X509TrustManager tm : x509TrustManagers ) {
90         try {
91             tm.checkServerTrusted(chain, authType);
92             return;
93         } catch ( CertificateException e ) {
94
95         }
96     }
97     throw new CertificateException();
98 }

```

Рисунок Д.2 – листинг кода авторизации android клиента, часть 2

```
100 public X509Certificate[] getAcceptedIssuers() {  
101     final ArrayList<X509Certificate> list = new ArrayList<X509Certificate>();  
102     for ( X509TrustManager tm : x509TrustManagers )  
103         list.addAll(Arrays.asList(tm.getAcceptedIssuers()));  
104     return list.toArray(new X509Certificate[list.size()]);  
105 }  
106 }  
107 }  
108 }
```

Рисунок Д.3 – листинг кода авторизации android клиента, часть 3