

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

Институт математики, физики и информационных технологий

(наименование института полностью)

Кафедра «Прикладная математика и информатика»

(наименование)

09.03.03 Прикладная информатика

(код и наименование направления подготовки, специальности)

Бизнес-информатика

(направленность (профиль) / специализация)

ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА (БАКАЛАВРСКАЯ РАБОТА)

на тему Разработка веб-приложения для сбора показаний счетчиков (на примере СНТ «Солнечная поляна – 1»)

Студент

А.С. Бурнаев

(И.О. Фамилия)

(личная подпись)

Руководитель

к.т.н., О.В. Аникина

(ученая степень, звание, И.О. Фамилия)

Тольятти 2020

АННОТАЦИЯ

Выпускную квалификационную работу (ВКР) выполнил Бурнаев Алексей Сергеевич, с темой: Разработка веб-приложения для сбора показаний счетчиков (на примере СНТ «Солнечная поляна – 1»).

Структура ВКР представлена введением, тремя главами, заключением, списком литературы и приложением.

Во введении обозначены: актуальность работы, цели и задачи, объект и предмет исследования.

В первой главе была выяснена деятельность СНТ «Солнечная поляна – 1», проанализированы основные бизнес-процессы организации, дано обоснование автоматизации процесса сбора показаний счетчиков приборов учета.

Во второй главе проведено проектирование с помощью современных технологий моделирования веб-приложения по сбору показаний счетчиков приборов учета света, газа и воды.

В последней главе описывается реализованная база данных, и веб-приложение на его основе, которое автоматизирует сбор показаний счетчиков. Обосновывается эффективность использования веб-приложения.

Заключение подводит итоги и указывает на выводы о проделанной работе.

В ходе выполнения ВКР были использованы 6 таблиц, 23 рисунка, 2 приложения с техническими материалами. Список использованной литературы содержит 33 источника. Общий объем ВКР составляет 53 страницы.

Оглавление

Введение.....	4
Глава 1 Анализ предметной области и моделирование бизнес-процессов СНТ «Солнечная поляна - 1»	6
1.1 Техничко-экономическая характеристика СНТ «Солнечная поляна - 1»	6
1.2 Концептуальное моделирование предметной области	8
1.2.1 Выбор технологии концептуального моделирования.....	8
1.2.2 Разработка и анализ модели бизнес-процессов «Как есть».....	9
1.2.3 Обоснование необходимости автоматизации сбора показаний счетчиков и формирование требований к проектируемому решению	14
1.2.4 Анализ существующих решений по сформулированным требованиям для веб-приложения.....	16
1.3 Построение модели «Как должно быть»	17
Выводы по главе 1	19
Глава 2 Проектирование веб-приложения и выбор средств реализации ..	20
2.1 Моделирование предметной области	20
2.2 Выбор архитектуры веб-приложения и проектирование БД	25
2.3 Выбор системы управления базы данных	29
2.4 Этапы разработки веб-приложения.....	30
2.5 Выбор технологии разработки веб-приложения	31
2.6 Требования к аппаратно-программному обеспечению веб-приложения.....	34
Выводы по главе 2	35
Глава 3 Программная реализация веб-приложения	36
3.1 Реализация веб-приложения с помощью фреймворка Django	36
3.2 Описание функциональности веб-приложения	39
3.3 Оценка эффективности использования веб-приложения	45

Выводы по главе 3	47
Заключение	48
Список используемой литературы	49
Приложение А Программный код.....	52
Приложение Б Пример квитанции	56

Введение

Человеческая деятельность становится немислимой без информационных технологий. Информация является одним из важных ресурсов в жизни общества. А для обработки, хранения и использования информации в жизни общества необходимы информационные системы.

Технологии сегодняшнего дня позволяют спроектировать и реализовать необходимые для человека информационные системы, которые частично заменяют ручной труд человека. Автоматизация определенной деятельности в организациях и учреждениях актуальная в наше время. Деятельность, которая повторяется и возможна автоматизации будет рассмотрена в СНТ «Солнечная поляна – 1».

Актуальность выполняемой ВКР обусловлена необходимостью сократить трудозатраты заказчика в своей деятельности.

Целью данной ВКР является разработка веб-приложения для сбора показаний счетчиков приборов учета (на примере СНТ «Солнечная поляна – 1»).

Объектом исследования является деятельность правления СНТ «Солнечная поляна – 1».

Предметом исследования является автоматизация бизнес-процесса финансово-экономической службы по сбору показаний счетчиков приборов учета света, газа и воды.

Для достижения поставленной цели необходимо выполнить следующие задачи:

- проанализировать бизнес-процессы;
- проанализировать существующие решения;
- разработать логическую модель веб-приложения;
- выбрать технологии разработки веб-приложения;
- выполнить программную реализацию веб-приложения;
- оценить эффективность работы по использованию веб-приложения.

В ходе выполнения ВКР будет реализовано веб-приложение для сбора показаний счетчиков света, газа и воды для СНТ «Солнечная поляна – 1», заказчиком, которого он и является.

В ходе выполнения ВКР были использованы 6 таблиц, 23 рисунка, 2 приложения с техническими материалами. Список использованной литературы содержит 33 источника. Общий объем ВКР составляет 53 страницы.

Глава 1 Анализ предметной области и моделирование бизнес-процессов

СНТ «Солнечная поляна - 1»

1.1 Техничко-экономическая характеристика СНТ «Солнечная поляна - 1»

СНТ «Солнечная поляна - 1» - это садовое некоммерческое товарищество, разновидность товарищества собственников недвижимости, или точнее, это добровольное объединение граждан, чьи садовые участки расположены на совместной территории, отведенной специально для садоводства. Некоммерческая природа товарищества проявляется в том, что все виды деятельности, приносящие ему доход, должны быть закреплены уставом СНТ. Заработанные СНТ средства должны использоваться для нужд товарищества (например, для приведения в порядок дорог общего пользования, устройства коллективной артезианской скважины либо организации дополнительного вывоза мусора) и не могут распределяться между его членами. Товарищество имеет все атрибуты, присущие любому юридическому лицу: собственное наименование, зарегистрированный устав, печать, расчетный счет в банке. Регистрируют это объединение в налоговом органе по месту нахождения. Кроме того, каждое СНТ должно состоять на учете в Пенсионном фонде РФ, Фонде обязательного медицинского страхования, Фонде социального страхования, а также в статистических органах, чтобы штатные сотрудники товарищества (электрик, бухгалтер) могли получать зарплату и социальные выплаты. Товарищество ведет бухгалтерский учет в соответствии с действующим законодательством, а отчетность представляет в местную налоговую инспекцию.

Органы управления СНТ представляют из себя трехзвенную структуру:

- Общее собрание членов товарищества;
- Правление;
- Ревизионная комиссия.

Деятельность СНТ происходит в соответствии с его уставом. Ревизионной комиссией является один ревизор, выбираемый на общем

собрании, который следит за исполнением финансовых и других решений общего собрания, проверяет деятельность правления.

Главой правления является председатель, который выбирается на общем собрании. Председателю подчиняется охранная служба, хозяйственная служба, финансовая-экономическая и юридическая службы. Сам председатель принимает заявления от членов товарищества, принимает необходимые управленческие решения, заключает сделки, выдает доверенности, принимает на работу, председательствует и исполняет решения общего собрания.

Бухгалтерская деятельность осуществляется на аутсорсинге в другой организации, там происходит формирование и сдача бухгалтерской отчетности, и формальное кадровое делопроизводство.

В охранной службе участвуют два охранника, один на пункте пропуска следит за тем, чтобы посторонние не проходили без его ведома, другой периодически обходит общую территорию.

В хозяйственную службу входит несколько человек, занимающиеся хозяйственными делами и делами благоустройства в СНТ, такими как обслуживание электросети, водопровода, ремонт дороги, заборов, и другой общей собственности членов СНТ.

Финансово-экономическая служба, состоящая из трех человек, занимается планированием расходов и возможных доходов СНТ от совместной деятельности, которая тратится на общие нужды членов СНТ. Занимается составлением приходно-расходных смет, отчетов правления товарищества и представляет их на утверждение общему собранию членов товарищества. Обосновывает необходимые траты расчетами. Также передает необходимые квитанции, платежки, счета и договора в организацию, занимающаяся бухгалтерскими функциями. При этом осуществляет контроль за членскими и целевыми взносами, принимает показания счётчиков света, газа и воды, выставляет счета членам СНТ, осуществляет расчеты с другими организациями, подготавливает отчеты для ревизора и председателя. Член Правления – экономист, который занимается сбором показания счетчиков.

Юридическая служба состоит из юриста и его помощника, которые обеспечивают необходимую юридическую составляющую деятельности СНТ, такую как, составление и проверку договоров, проверку на законность действий СНТ, и осуществление консультаций всем членам товарищества.

Организационная структура СНТ «Солнечная поляна - 1» представлена на рис. 1.1.

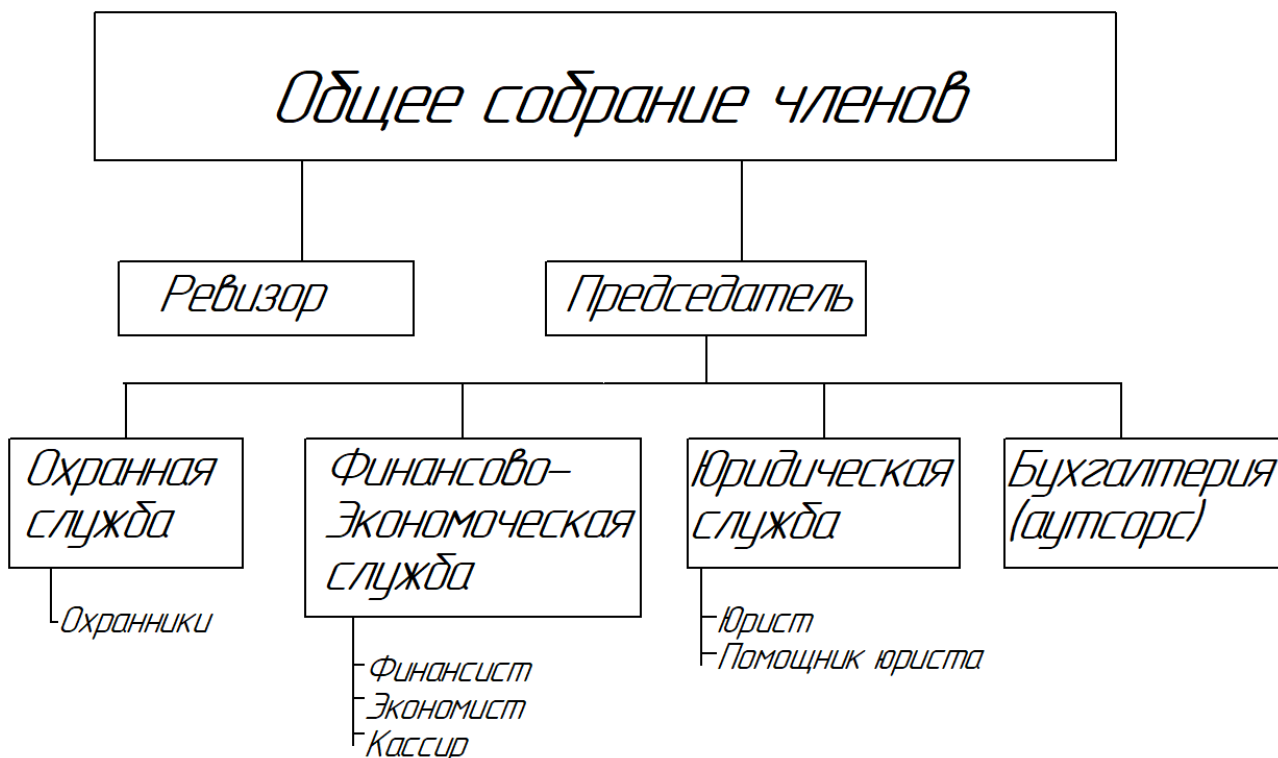


Рисунок 1.1 – Организационная структура СНТ «Солнечная поляна - 1»

Таким образом, исполнительным органом является правление СНТ, представляющее из себя линейно-функциональную структуру, имеющую в своем составе небольшое количество сотрудников.

1.2 Концептуальное моделирование предметной области

1.2.1 Выбор технологии концептуального моделирования

Бизнес-процессы СНТ «Солнечная поляна – 1» разноплановые и кратковременные, но периодические и постоянные. Анализ и моделирование

предметной области необходим для точной и грамотной постановки задачи автоматизации одного из бизнес-процессов организации, а для этого необходимо найти взаимосвязи с другими бизнес-процессами организации. Что может быть сделано с помощью различных методологий проектирования автоматизированных информационных систем, в основе которой лежит понятие бизнес-модели автоматизированной информационной системы (АИС). При этом анализируются основные бизнес-процессы организации и приводятся к описанию в терминах методологии проектирования АИС, и описываются на таких уровнях как:

- концептуальный уровень (содержательное описание в разрезе структурного подхода);
- логический уровень (объектно-ориентированный подход);
- физический уровень (программная реализация).

Наиболее известные и применяемые нотации описания бизнес-процессов на концептуальном уровне: IDEF0, IDEF3, DFD.

IDEF0 простой блочно-функциональный способ отображения бизнес-процессов. Данный способ описания сконструирован для бизнес-аналитики, и хорошо отлажен, способен моделировать процессы как в маленьких организациях, в отдельных отделах, так и в иерархически сложных структурах.

IDEF3 диаграмма предназначена для документирования в организации технологических действий.

DFD – это диаграмма потоков данных, такая нотация обеспечивает структурную визуализацию потока входных данных, которые реорганизуются в выходные данные.

В данной работе достаточно будет использовать графическую нотацию IDEF0. И, если необходимо, произвести декомпозицию функциональных блоков на более мелкие операции.

1.2.2 Разработка и анализ модели бизнес-процессов «Как есть»

Так как СНТ «Солнечная поляна – 1» выступает в роли заказчика, и из всех бизнес-процессов организации в первую очередь ее интересует

автоматизация деятельности финансово-экономической службы, которая занимается составлением смет, бизнес-планов, принимает платежи, взносы от членов СНТ, производит все необходимые платежи, составляет необходимые отчеты. Полноценная автоматизация влечет за собой создание целой АИС, а ресурсы СНТ не так велики. Поэтому заказчиком поставлена задача автоматизации процесса сбора показаний счетчиков по расходу воды, электричества, газа от каждого члена СНТ, но в данный момент эта деятельность не автоматизирована, но должна быть использована платформа, с возможностью расширения функционала в будущем.

Правление СНТ принимает оплату за расходование света, газа и воды по тарифным сеткам поставщиков услуг на расчётный счет СНТ, или оплату принимает кассир-операционист. Член Правления (экономист) в конце каждого месяца принимает показания счетчиков на личном приеме, на бумажных квитках в почтовом ящике правления СНТ, или на электронной почте. Далее заносит эти показания в электронную таблицу Excel, где рассчитывается сумма к оплате по тарифным сеткам поставщиков услуг. При этом все показания счетчиков суммируются, и сравниваются с показаниями общих счетчиков СНТ. Разница в суммарных показаниях дает дополнительную величину оплаты, которая распределяется между всеми членами равномерно, так как считается, что это общие технологические потери. При этом если разница слишком велика, есть причина проверить счетчики всех членов СНТ, или начать поиск утечек. Затем экономист рассылает вручную выставленные счета к оплате по электронной почте. Так происходит процесс сбора показаний счетчиков.

Тем самым процесс сбора показаний счетчика является одним из «узких», неавтоматизированных мест бизнес-процессов СНТ «Солнечная поляна – 1», а в частности деятельности финансово-экономической службы по работе с членами СНТ. Поэтому необходимо построить графическую нотацию IDEF0 ее деятельности, которая представляет из себя контекстную диаграмму в виде одного общего функционального блока A0, на вход которого поступает информация общего и частного характера, денежные средства, заключенные

договора. Функциональный блок ограничен уставом организации, и соответствующим законодательством. При этом используются для работы блока специалисты, компьютерная и офисная техника. И на выходе получают выставленные счета, квитанции, оплаченные счета, сметы, бизнес-планы, отчеты.



Рисунок 1.2 – Контекстная диаграмма деятельности финансово-экономической службы «Как есть»

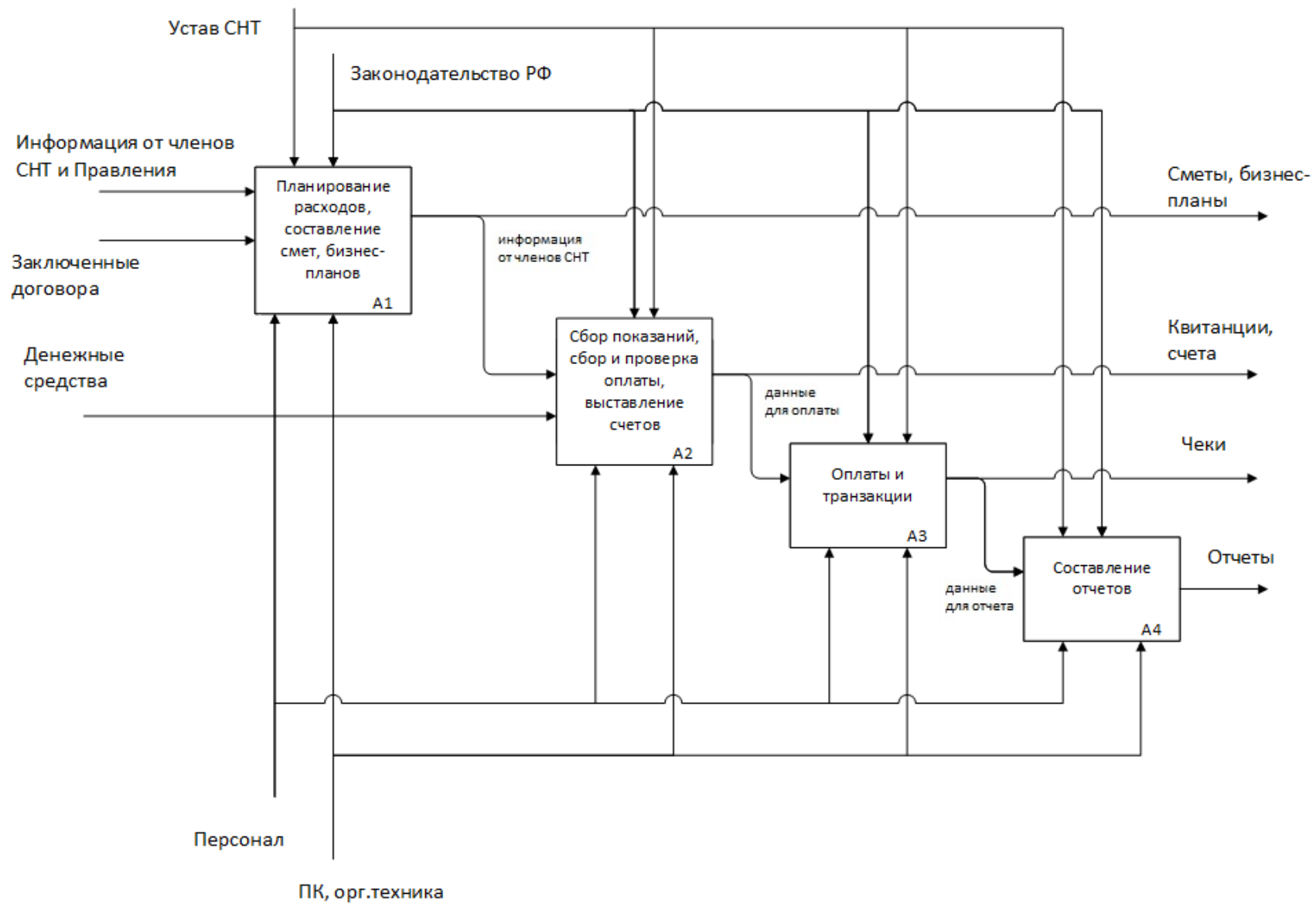


Рисунок 1.3 – Декомпозиция контекстной диаграммы «Как есть»

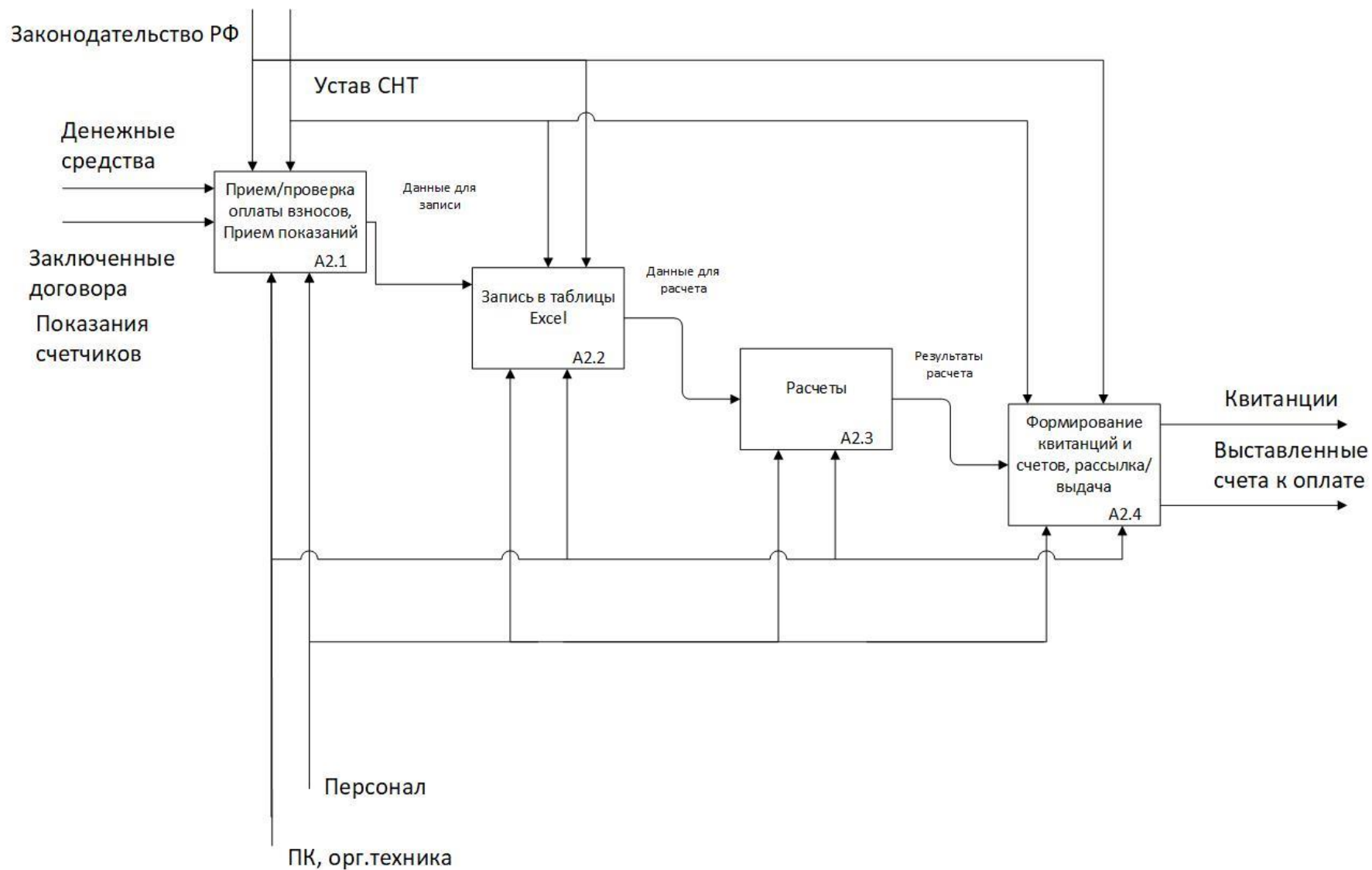


Рисунок 1.4 – Декомпозиция контекстной диаграммы «Как есть» блока А2

Для дальнейшего моделирования на рис. 1.3 выполнена декомпозиция блока А0 процесса деятельности финансово-экономической службы.

Теперь возможно проанализировать более детально каждый функциональный блок, который из себя представляет отдельный процесс:

- процесс планирования расходов, составление бизнес-планов и смет;
- процесс сбора платежей и взносов;
- процесс оплаты счетов и проведение транзакций;
- процесс составления отчетной документации.

Нас интересует процесс сбора платежей и взносов, который изображен в виде блока А2 (рис. 1.3). Декомпозиция блока А2 изображена на рис. 1.4.

Декомпозиция блока А2 произведена на подпроцессы:

- Прием взносов от членов СНТ, проверка поступления безналичных взносов;
- Прием показаний счетчиков света, газа и воды;
- Запись в соответствующие таблицы (для показаний, целевых расходов, и другие);
- Производство расчетов, проверка расчетов;
- Формирование квитанций, выставление счетов и их рассылка или уведомление.

Таким образом, построив модели «Как есть», можно будет построить модель «Как должно быть», учитывая цели автоматизации.

1.2.3 Обоснование необходимости автоматизации сбора показаний счетчиков и формирование требований к проектируемому решению

Разобравшись с декомпозицией процессов на подпроцессы, можно понять, что прием показаний счетчиков, их фиксация, расчеты стоимости

оплаты за расход света, газа и воды, формирование и рассылка квитанции достаточно трудоемки, требует выделения отдельного времени персонала для принятия информации показаний счетчиков от каждого члена СНТ, и выставление счетов к оплате каждому члену СНТ. Что можно автоматизировать с помощью отдельного веб-приложения, которое может быть интегрировано затем в полноценную АИС. А также использование баз данных внутри веб-приложения повысит надежность хранения информации и даст дистанционный, многопользовательский доступ к данным, как Правлению СНТ, так и членам СНТ.

Требуется чтобы был реализован веб-сайт с такими характеристиками и возможностями, как:

- должен иметь информационную структуру в режиме доступа 24/7;
- можно зарегистрироваться каждому члену СНТ;
- заносить туда показания счетчиков, находить необходимую информацию;
- видеть историю показаний;
- получать квитанцию к оплате;
- доступ к веб-приложению должен быть многопользовательским;
- администратору можно было проверять и вносить поправки, если это необходимо к введенной информации;
- автоматически производить расчет и формировать квитанции к оплате, отчеты, и автоматически их получать;
- программное обеспечение по умолчанию должно иметь поддержку популярных браузеров;
- возможность формировать квитанции в pdf-формате, отправлять форму на почту, или просто выводить на экран;
- реализован на такой платформе, которая позволит достаточно просто расширять функционал под новые задачи СНТ в будущем (такие как новостной блок, блок для дистанционного голосования).

В следующем разделе проанализируем существующие на рынке автоматизированные решения по сформулированным в данной работе требованиям.

1.2.4 Анализ существующих решений по сформулированным требованиям для веб-приложения

Рассмотрим далее типичные известные автоматизированные решения, предлагаемые на рынке. Широко известна автоматизация предприятий и организаций на основе прикладного решения «1С: Предпр. 8. Бухгалтерия СНТ. Электронная поставка». Но такая широкая автоматизация будет слишком избыточна, так как в ней присутствуют блоки бухгалтерского учета, а в СНТ бухгалтерия находится на аутсорсинге в другой организации, и достаточно затратна. Также для автоматизации 1 рабочего места необходимо приобрести электронную поставку.

Также известны автоматизированные решения использующие счетчики с дистанционной передачей своих показаний (например, saures.ru - Система автоматизированного учёта и контроля ресурсов). Но такие решения подразумевают покупку дорогостоящих специализированных счетчиков, и соответственно замену старых счетчиков на новые, что не подходит для членов СНТ, так как неоправданно дорого.

Существуют просто облачные сервисы, специализирующиеся на автоматизации учета в жилищно-коммунальном хозяйстве (например, smartflex.ru – Автоматизация учета в ЖКХ). Но они рассчитаны на большое количество пользователей, подразумевают платное обслуживание, в виде абонентской платы. Что опять же по затратам будет избыточным решением для членов СНТ, нежели свое простое веб-приложение. Поэтому наиболее подходящим способом решения существующей проблемы сбора показаний счетчиков будет самостоятельная разработка приложения, которое будет учитывать все особенности автоматизируемого бизнес-процесса, и отвечать существующим потребностям Правления СНТ.

В таблице 1 приведен сравнительный анализ известных решений.

Таблица 1 – Сравнительный анализ известных решений

№	Критерии	Названия			
		1С: Бухгалтерия СНТ snt1c.ru	saures.ru	smart-flex.ru	Разрабатываемое веб-приложение
1.	Не требуется лицензия	-	-	-	+
2.	Простота развертывания	-	-	+	+
3.	Не требуется покупка дополнительного оборудования	+	-	+	+
Итого		1	0	2	3

Для автоматизации сбора показаний счетчиков преимущественно использовать собственное веб-приложение, реализованное в виде веб-интерфейса к бизнес-логике сбора показаний счетчиков. Бизнес-логика будет осуществляться с использованием базы данных, расположенной на сервере, и разработанной программной логикой.

1.3 Построение модели «Как должно быть»

Для автоматизации бизнес-процесса сбора показаний счетчиков света, газа и воды необходимо построить модель того, как должен быть устроен новый или изменённый бизнес-процесс.

Так как автоматизируемый бизнес-процесс сбора показаний счетчиков рассмотрен в контекстной диаграмме блока А2 (рис. 1.3), необходимо смоделировать новую диаграмму «Как должно быть» блока А2. Глубоко проанализировав бизнес-процесс сбора показаний счетчиков, приведем контекстную диаграмму «Как должно быть» блока А2 на рисунке 1.5.

Контекстная диаграмма «Как должно быть» блока А2 реализована с учетом требований к разрабатываемому веб-приложению в рамках измененного бизнес-процесса по поставленной задаче заказчика сократить трудоёмкость и время сбора показаний счетчиков света, газа и воды.

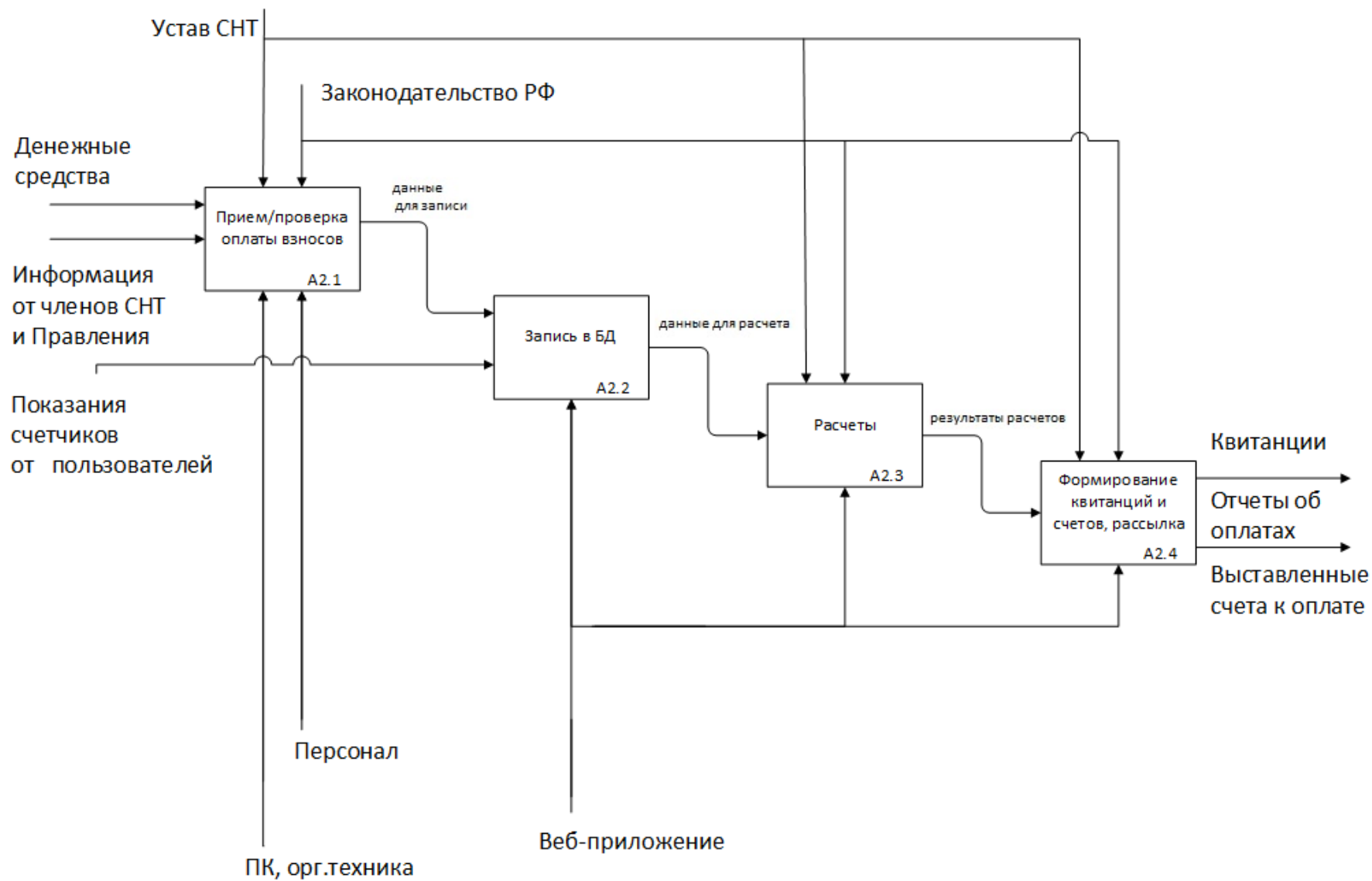


Рисунок 1.5 – Контекстная диаграмма «Как должно быть» блока А2

Теперь блок А2.1 принимает входящую информацию, денежные средства от членов СНТ, а веб-приложение, персонал и офисная техника выступают инструментом этого блока. Затем поступившие данные записываются в базу данных с помощью веб-приложения в следующем блоке А.2.2, в следующем блоке А2.3 данные записей рассчитываются в результаты, необходимые для блока А2.4, что приводит к выходу из диаграммы с результатами выполнения всего бизнес-процесса сбора показаний счетчиков, таких как квитанции, отчеты, выставленные счета. Теперь с помощью веб-приложения происходит запись в базу данных информации об оплате взносов, показания счетчиков, расчет необходимой суммы к оплате, формирование квитанций и счетов, отправка их на электронную почту членам СНТ или просто получение их в электронном виде. Доработанная модель автоматизирует сбор показаний счетчиков, выставления квитанций и счетов, что должно сократить затраты рабочего времени финансово-экономической службы, а также повысить надежность хранения информации.

Выводы по главе 1

В первой главе был проведен детальный анализ некоторых бизнес-процессов СНТ «Солнечная поляна – 1», дана краткая характеристика организации.

Проведена декомпозиция бизнес-процессов, построены контекстные диаграммы «Как есть», анализ которых показал возможность автоматизации бизнес-процесса сбора показаний счетчиков. Была построена новая контекстная диаграмма «Как должно быть».

Определена целесообразность автоматизации бизнес-процессов, которая сократит трудозатраты финансово-экономической службы, а также повысит надежность хранения информации о деятельности СНТ, о ее внутренних процессах. Что позволит выполнить поставленные цели и задачи.

Глава 2 Проектирование веб-приложения и выбор средств реализации

2.1 Моделирование предметной области

Правлением СНТ «Солнечная поляна - 1» была поставлена задача автоматизации сбора показаний приборов учета света, газа и воды, решением которой будет разработка веб-приложения. Следующим шагом в разработке веб-приложения является логическое проектирование, оно поможет в физической разработке и детальном анализе взаимосвязей объектов, которые будут необходимы для данной работы. Для этого нужно будет построить диаграмму вариантов использования, где может применяться один из языков моделирования (например, UML).

Диаграмма вариантов использования отражает отношения между акторами и прецедентами, что позволяет описать будущую систему на концептуальном уровне. За ее основу возьмем контекстную диаграмму «Как должно быть». Созданная диаграмма вариантов использования показана на рисунке 2.1.



Рисунок 2.1 – Диаграмма вариантов использования

На диаграмме вариантов акторы это член СНТ, и член Правления СНТ. Член СНТ вносит показания, получает квитанцию к оплате в электронном виде, может просмотреть свои прошлые показания. Член Правления СНТ может редактировать какие-либо ошибки в записях, вводить новые тарифы для услуг, просматривать статистику показаний, просматривать отчеты, и общие суммарные показания всех членов СНТ.

Таким образом, формализуем действия акторов и опишем прецеденты в таблице 2.

Таблица 2 – Характеристика прецедентов

№	Прецедент	Характеристика
1.	Внесение показаний	Внесение показаний счетчиков на свет, газ и воду, и запись их в БД.
2.	Получение квитанции к оплате	Получение квитанции к оплате на почту или просмотр в личном кабинете информации о сумме к оплате.
3.	Просмотр Истории	Просмотр прошлых показаний
4.	Редактирование записей	Редактирование записей пользователей, если были внесены некорректные данные.
5.	Ввод тарифов	Ввод новых тарифов при их изменении в новый отчетный период.
6.	Просмотр Истории	Просмотр полной статистики по всем пользователям, просмотр всех историй.

Далее создадим диаграмму последовательности на рисунке 2.2. Там изображено каждое действие актора и, разрабатываемого веб-приложения.

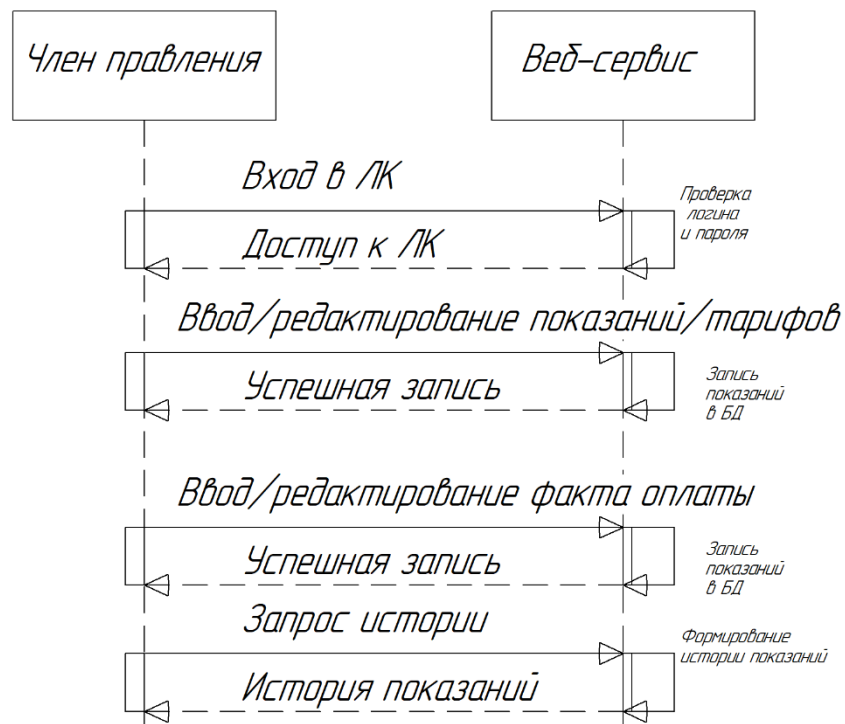
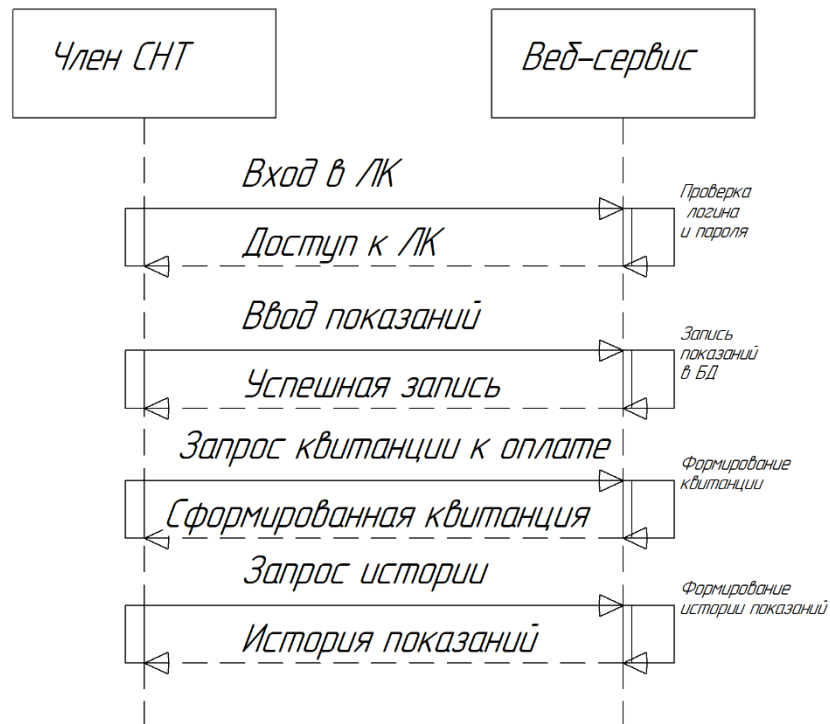


Рисунок 2.2 – Диаграмма последовательности использования веб-приложения

Член СНТ заносит показания счетчиков, в расчетный период получает квитанцию к оплате в личном кабинете. Просматривает прошлые показания, смотрит прошлые оплаты, отчеты и другую информацию от Правления СНТ

если такая была опубликована на веб-сайте. Член Правления редактирует показания, если те были введены неправильно. Заносит в БД показания тарифов на новый расчетный период, если те изменились. Также член Правления смотрит статистику по всем членам СНТ, общие отчеты и вносит информацию о факте оплаты.

Логическая модель данных описывает взаимодействие объектов (сущностей) и их связи, и представлена на рисунке 2.3.

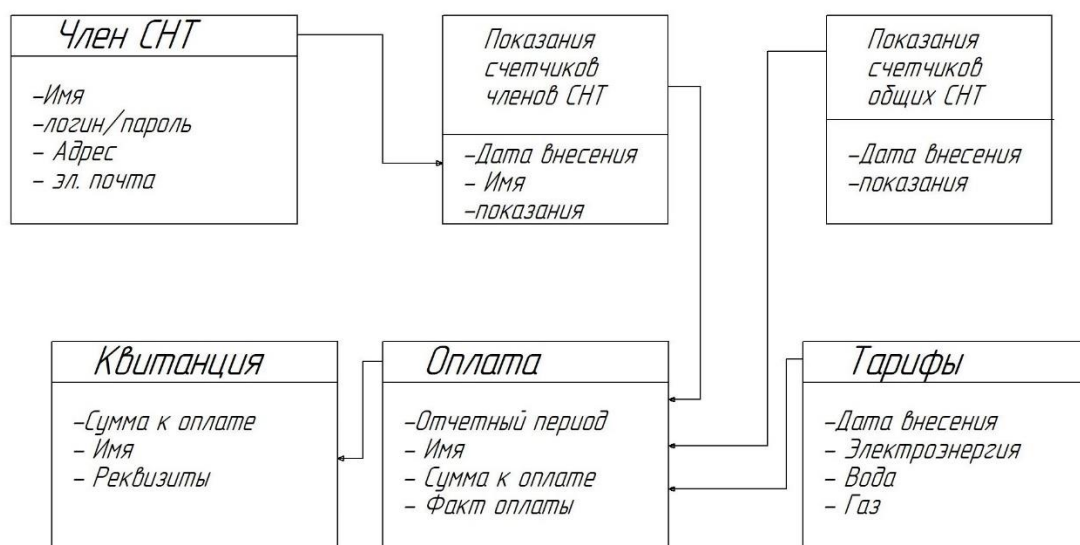


Рисунок 2.3 – Логическая модель данных

Как видно на рис. 2.3 «Член СНТ» это сущность содержащая информацию о членах СНТ, другая сущность это «Показания счетчиков членов СНТ», которая связана с первой сущностью, Так же существует сущность «Тарифы» и «Показания общих счетчиков», которые связаны как и предыдущие с сущностью «Оплата», а далее сущность «Квитанция» уже является производной от «Оплаты».

Для формирования квитанций к оплате необходимо провести автоматические операции расчета суммы, которую необходимо оплатить члену СНТ, что при этом сократит время работы членов Правления СНТ. Расчет

суммы к оплате в квитанции к оплате должен содержать все необходимые параметры для заполнения и автоматически рассчитываться.

Расчет суммы к оплате определяется по формуле:

$$S = L_1 + L_2 + L_3 + P_1 + P_2 + P_3 \quad (1.1)$$

где S – сумма к оплате,

$L_1 + L_2 + L_3$ – сумма личного потребления света, газа и воды

$P_1 + P_2 + P_3$ – сумма общих потерь, разделенная на всех членов СНТ, по каждой позиции (свет, газ, вода).

$$L = R \times (P_2 - P_1) \quad (1.2)$$

Где L – это сумма за потребление света, или газа, или воды к выражению $L_1 + L_2 + L_3$,

R – ставка для соответствующего тарифа,

$(P_2 - P_1)$ – разница между текущим показанием счетчика и прошлым члена СНТ.

$$P = R \times ((PO_2 - PO_1) - \sum (P_2 - P_1)_i) / N \quad (1.3)$$

Где P – сумма общих потерь, разделенная на всех членов СНТ,

N – количество всех членов СНТ.

$PO_2 - PO_1$ – разница между текущим показанием и нынешним общего счетчика СНТ.

Таким образом, определенные сущности и связи между ними, понадобятся для проектирования базы данных как часть АИС, в следующем разделе приведем модель базы данных и архитектуру веб- приложения.

2.2 Выбор архитектуры веб-приложения и проектирование БД

Много элементов или объектов, связанных вместе для выполнения какой-то функции, которая необходима для достижения каких-либо целей, не может существовать без какой-то внешней структуры, которая бы давала более систематическое и простое видение, более легкое в реализации решение. Поэтому необходимо выбрать архитектуру веб-приложения из уже имеющейся практики на сегодняшний день.

Архитектура «клиент-сервер» определяет организацию взаимодействия в сети, в которой взаимодействуют сервера, узлы-поставщики функций и данных, и потребители. Архитектура веб-приложения может иметь два или три звена. В данной работе нам необходима 3-х звенная архитектуры, так как будут использоваться фреймворки (приложения/программная логика), которые в свою очередь представляют из себя промежуточное звено между базой данных (БД) и веб-сервером, взаимодействующим с клиентом.

На рисунке 2.4 представлена архитектура веб-приложения.

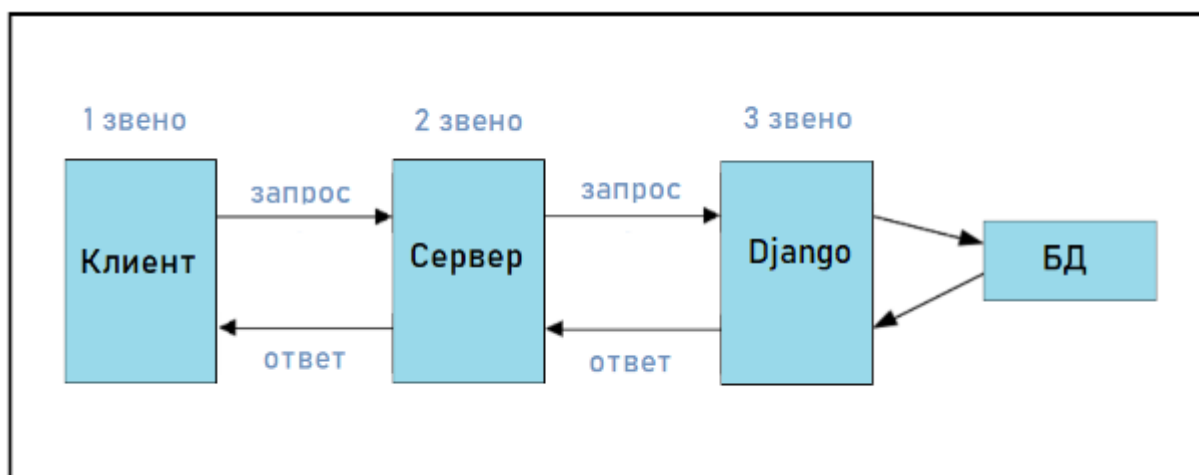


Рисунок 2.4 – Трехзвенная клиент-серверная архитектура

И так как физически клиент и сервер находятся в разных местах, то распределение компонентов веб-приложения определяет модели взаимодействия, таких как, например, сервер терминалов, файл-сервер, сервер БД, сервер приложений. В данной работе необходим Сервер – это программа,

точки зрения использования потребляемых ресурсов. Универсальности доступа для клиентов.

Недостатками же является то, что нужен отдельный специалист по администрированию серверов, и если сервер не работоспособен, то доступ к данным прекращается у всей сети.

Ядром веб-приложения является БД, в которой происходит основная работа этого приложения. БД должна соответствовать требованиям заказчика, таким как:

- отказоустойчивость;
- выбор необходимых типов данных для хранения;
- высокую степень защиты;
- хорошую пропускную способность;
- возможность миграции данных;
- простоту администрирования.

Для начала необходимо разработать структуру таблиц БД. То есть таблицы должны содержать необходимые поля под определенные типы данных, или, другими словами, в эти поля, объединенные в некие таблицы, мы должны поместить наши сущности, необходимые для реализации бизнес-логики заказчика. В разрабатываемой структуре записей (таблицы) в БД не должно быть лишних полей, но и должно быть учтено все необходимое, иначе в процессе эксплуатации изменение или внесение новых типов полей будет осуществить затруднительно.

Для нашего ядра веб-приложения необходимо иметь информацию о данных самого пользователя, о его вносимых показаниях счетчиков на свет, газ и воду, дату внесения показаний, а также информацию о текущих тарифах на услуги в данный период, а также данные об оплаченных счетах. Для этого составим необходимые таблицы данных в БД. В таблице 2.1 описаны поля логической модели базы данных.

Таблица 2.1 – Поля и типы данных логической модели БД

№	Имя таблицы	Название атрибута	Описание
1.	Payment	NameUser	ForeignerKey Ключ один ко многим
2.		Period	Отчетный период использования, тип поля данных дата
3.		Total	Сумма, вещественный тип поля
4.		Check	Факт оплаты, логический тип поля
5.		View_list	Лист квитанции, тип Текс
6.	UserSNT	Id	Первичный ключ для каждой записи данных, которые вносит пользователь, тип данных текстовый
7.		NameUser	Имя члена СНТ, который вносит показания счетчиков, тип поля данных текстовый
8.		Adress	Номер участка члена СНТ8
9.		WhenRec	Дата внесения показаний, тип поля данных дата
10.	Rate	r_id	Первичный ключ для каждой записи тарифов, текстовый тип поля
11.		r_gas	Тариф на потребление газа, Руб/м ³ , вещественный тип поля
12.		r_light	Тариф на потребление электричества, Руб/кВт.ч, вещественный тип поля
13.		r_water	Тариф на потребление воды, Руб/м ³ , вещественный тип поля
15.		r_when	Дата действия тарифа, тип поля дата
16.	UnitValue	u_id	Первичный ключ для каждой записи тарифов, текстовый тип поля
17.		u_gas	Общие показания потребления газа, вещественный тип поля
18.		u_light	Общие показания потребления электричества, вещественный тип поля
19.		u_water	Общие показания потребления воды, вещественный тип поля
20.		u_when	Дата, тип поля дата
21.	Meter	Createat	Дата, тип поля дата
22.		NameUser	ForeignerKey
23.		Water	Показания счетчиков воды, вещественный тип поля
24.		Light	Показания счетчиков света, вещественный тип поля
25.		Gas	Показания счетчиков газа, вещественный тип поля

На рисунке 2.6 представлена физическая модель базы данных веб-приложения на основе логической модели данных, при этом использованы типы полей, распространенные у всех реляционных БД.

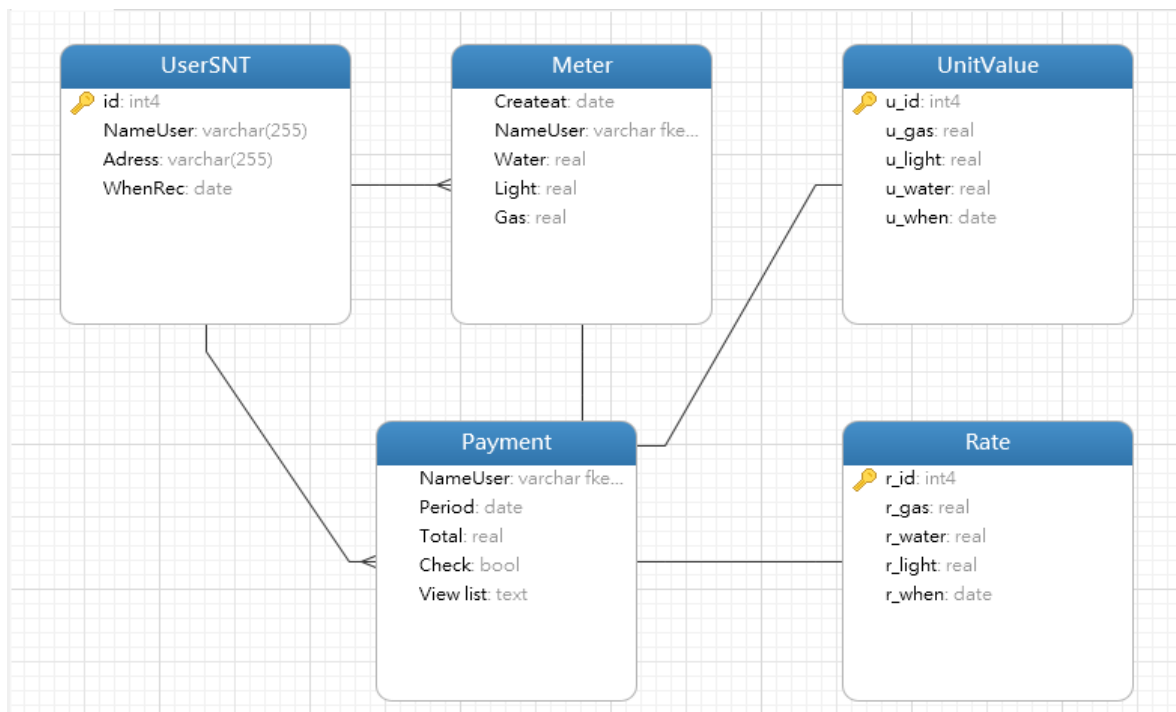


Рисунок 2.6 – Физическая модель базы данных

Таким образом, разработанная физическая модель БД, являющаяся ядром веб-приложения, позволит осуществить дальнейшие шаги в программной разработке веб-приложения.

2.3 Выбор системы управления базы данных

База данных – это хранилище данных, при этом данные могут быть разнотипные, или по-другому это некий файл со своей определенной кодировкой информации, записанной определенным образом, для надежного и эффективного ее хранения. Для удобства использования базы данных используются системы управления базой данных (СУБД). А для обслуживания клиентов, которые обращаются к хранилищу данных за необходимой информацией, используется сервер БД, при этом он может быть в составе самой СУБД. Далее нам необходимо выбрать СУБД для нашего веб-приложения.

Ниже рассмотрены наиболее известные и популярные реляционные БД, которые можно использовать для наших задач.

MySQL – популярная и полнофункциональная СУБД. Может хранить множество типов данных и огромное количество информации с этими типами данных. Имеет также свой собственный интерфейс.

PostgreSQL – считается самой продуктивной и поддерживает популярные объектно-ориентированные модели данных. Имеет свой собственный интерфейс, и собственный клиент.

SQLite – система основана на файлах. При работе вызовы отправляются сразу в файлы. Эта СУБД быстрая и простая. Но она не рекомендуется для использования в промышленной производстве программных продуктов. Не имеет своего интерфейса.

Сравним в таблице 2.2 СУБД.

Таблица 2.2 – Сравнение СУБД

№	Критерий	Название		
		MySQL	PostgreSQL	SQL
1.	Удобство эксплуатации	+	+	-
2.	Гибкость настройки	+	+	-
3.	Высокая производительность	+	+	+
4.	Широкий функционал	+	+	-
5.	Опыт работы	-	+	+
6.	Простота реализации	+	+	+
Итого:		5	6	3

Проведя анализ путем попарного сравнения, приходим к выводу, что целесообразно использовать СУБД PostgreSQL, одна из актуальных версий – это версии 12.

2.4 Этапы разработки веб-приложения

Одной из задач разработки веб-приложения было разработать структуру веб-сайта для использования через сеть интернет. В котором должны были быть макет страниц для отображения, такие как:

- Главная страница;
- Вход в личный кабинет;

- Страница личного кабинета для подачи показаний счетчиков;
- Страница личного кабинета администратора;
- Страница для просмотра всех поданных показаний;
- Страница для просмотра квитанций к оплате.

Другая задача это с помощью программной логики будет осуществить отображение данных, которые будут браться из БД.

И последняя из задач это проверить, протестировать и развернуть разрабатываемое приложение на реальном хостинге.

При этом веб-приложение должно обладать следующими особенностями:

- гибкостью;
 - удобной для администратора системой управления;
 - удобным юзабилити-дизайном, то есть удобным в использовании и интуитивно понятным для пользователей;
- для пользователя также была возможность распечатать квитанцию к оплате.

Однако, конечно, главной и самой сложной задачей проектирования является разработка системы управления содержимым веб-приложения, которая бы позволяла вносить изменения и правки в случае необходимости, с возможностью разграничения прав доступа содержимому и независимостью от технических специалистов или разработчиков веб-приложения. В программной реализации веб-приложения будут осуществлены все основные поставленные задачи.

2.5 Выбор технологии разработки веб-приложения

Успешность проекта зависит от выбора технологии разработки, которая должна удовлетворять всем необходимым требованиям заказчика, но при этом должна быть не слишком трудозатратной, долго поддерживаемой сообществом специалистов. Поэтому необходимо рассмотреть варианты технологий разработки нашего веб-приложения.

Для реализации данного веб-приложения на минимальном уровне необходимо, чтобы с веб-интерфейса, который будет отображать веб-браузер, пользователю давался доступ к БД, для получения или внесения необходимых данных. Такое решение, на прямую давать доступ к базе, тоже реализуемо. Но обладает недостатками, такими как, отсутствие выполнения требований безопасности, потому что данные может получить любой человек имеющий знания по написанию SQL-запросов. А также такое решение нельзя будет интегрировать на этом же уровне в более сложную информационную систему, если такое потребуется в будущем, и соответственно пользователь должен обладать более продвинутыми знаниями веб-технологий. Поэтому нам необходимо такое решение, которое позволит сделать безопасным и удобным получение и отправку данных для пользователя. Такому решению соответствуют современные языки программирования с удобными, автоматизированными библиотеками-фреймворками, которые позволят реализовать логику работы веб-приложения взаимодействующую и с серверами, и с БД, применение языков программирования на стороне пользователя для нас будет избыточным. Потому что веб-браузеры современных стандартов позволяют отправить и получить данные очень быстро, и их отобразить в удобном и понятном виде с необходимым стеком протоколов и стандартов, а вся работа будет выполнена на стороне сервера-машины серверной программой и программой нашего приложения, которое будет формировать шаблон интерфейса пользователя, и работать с БД.

HTML – это прежде всего стандарт для разметки, маркирования и кодирования информации, которую будет отображать веб-браузер для пользователя. Необходимо использовать по возможности последние версии стандарта. Альтернативы ему практически нет.

CMS – система управления содержимым (контентом) – уже готовое программное решение, которое имеет свой графический веб-интерфейс и настаивается на уровне обычного пользователя, но требует знания тонкостей подключения к БД. С помощью этого решения можно реализовать веб-

интерфейс к БД, в которой будет прописана логика работы с данными пользователей. Решение будет платным, и расположено на определенном хостинге, который уже задан на первом уровне.

PHP – это распространенный язык программирования общего назначения с открытым исходным кодом. Он специально сконструирован для веб-разработок и его код может внедряться непосредственно в HTML. Но PHP интерпретатор медленно обрабатывает большие данные, и поэтому в будущем стоит под сомнением его популярность у сообщества специалистов.

JAVA – язык программирования со сложной объектно-ориентированной моделью данных, интерпретатор, по сути, имеет собственную виртуальную кроссплатформенную машину, на которой выполняется свой так называемый байт-код.

Python3 – популярный высокоуровневый объектно-ориентированный язык программирования с удобочитаемым синтаксисом для разработчика. С богатым набором инструментов для сетевого, серверного программирования приложений, как и в JAVA, но Python с открытым исходным кодом и свободной лицензией для использования.

Golang – это название эффективного компилируемого многопоточного языка программирования, который разработала компания Google. Хотя Golang и объектно-ориентированный внутри, но он не поддерживает полноценную объектно-ориентированную модель данных, а больше ориентирован на многокомпонентную модель с привязкой методов и данных друг к другу без наследования, как в объектно-ориентированных языках.

Составим сравнительную оценку в балах в таблице 2.3 выбранных технологий для разработки нашего веб-приложения.

Таблица 2.3 – Сравнение выбранных технологий разработки

№	Критерий	Вариант технологии разработки				
		CMS	PHP	Java	Python3	Golang
1.	Простота обучения	+	+	-	+	-
2.	Поддержка сообществом	-	+	+	+	+
3.	Популярность у	-	-	+/-	+	+

	сообщества					
4.	Простота использования	+	+	-	+	+/-
5.	Широкие возможности с работой с БД	-	+	+	+	+
6.	Эффективность работы со средней загруженностью по обработке данных, и обработке запросов	-	-/+	+	+/-	+
7.	Скорость разработки	+	-/+	-	+	+/-
Итого:		3	5,5	3,5	6,5	5

Проведя сравнение технологии разработки, было принято решение использовать для отображения веб-интерфейса язык разметки HTML. Для всего остального использовать язык программирования Python3 и его фреймворки для создания веб-приложений. Наиболее популярным полнофункциональным фреймворком является Django. Выбор данного фреймворка был сделан по требованию заказчика. Так как в будущем заказчик собирается расширить функционал разрабатываемого веб-приложения.

2.6 Требования к аппаратно-программному обеспечению веб-приложения

Важной составляющей Веб-приложения является его «железная» часть, без которой невозможна реализация поставленных задач автоматизации. Веб-приложение должно качественно и безопасно работать, у него должно быть свое уникальное доменное имя, как минимум второго уровня. Поэтому определим требования к аппаратно-программному обеспечению:

- операционная система – Windows10;
- центральный процессор – тактовая частота не менее 1,4 ГГц;
- оперативная память – не менее 1 Гб;
- свободное место на жестком диске – не менее 1 Гб;
- подключение к сети интернет;

- интернет-браузер.

Требования к серверной части:

- круглосуточная работа;
- процессор – не ниже Pentium IV;
- оперативная память – не менее 512 Мб;
- свободное место на SSD диске – не менее 2 Гб;
- пропускная способность интернет-канала – не менее 50 Мб/с.

Требования к серверу СУБД и АИС:

- процессор – тактовая частота не менее 1.4 ГГц;
- оперативная память – не менее 1 Гб;
- свободное место на жестком диске – не менее 2 Гб;
- пропускная способность интернет-канала – не менее 50 Мб/с.

Требования к доменному имени:

- доменное имя не ниже 3 уровня;
- отсутствие истории Whois;
- название домена на латинице.

Требования к безопасности:

- требуется разграничить доступ для членов СНТ и членов правления;
- требуется использовать такие библиотеки со стороны языка программирования, которые бы исключали «sql-инъекции»;
- разграничение доступа к данным в БД должно выполняться на уровне приложения, и также на уровне СУБД.

Выводы по главе 2

Во второй главе была описана логическая модель веб-приложения, показана диаграмма вариантов использования, диаграмма последовательности. Определены требования к аппаратно-программному обеспечению, разработана логическая модель базы данных, описаны операции расчета суммы к оплате для члена СНТ. Тем самым была проделана необходимая предварительная работа для непосредственной разработки веб-приложения.

Глава 3 Программная реализация веб-приложения

В данной главе описывается практическая часть реализации веб-приложения по сбору показаний счетчиков приборов учета от членов СНТ «Солнечная поляна – 1».

3.1 Реализация веб-приложения с помощью фреймворка Django

В ходе проектирования веб-приложения были рассмотрены некоторые технологии, посредством которых можно реализовать требуемое веб-приложение, и выбор был сделан в пользу языка программирования Python версии 3.8 и на его основе фреймворка Django (далее фреймворк) версии 3.0.

Данный фреймворк с открытым исходным кодом в свободном доступе. В нем уже реализован некий каркас будущего веб-приложения, с продуманной архитектурой, и имеет все необходимые инструменты для выполнения большинства задач для реализации веб-приложений.

На рисунке 3.1 представлена структура базового проекта с тремя приложениями (другими словами сервисами).

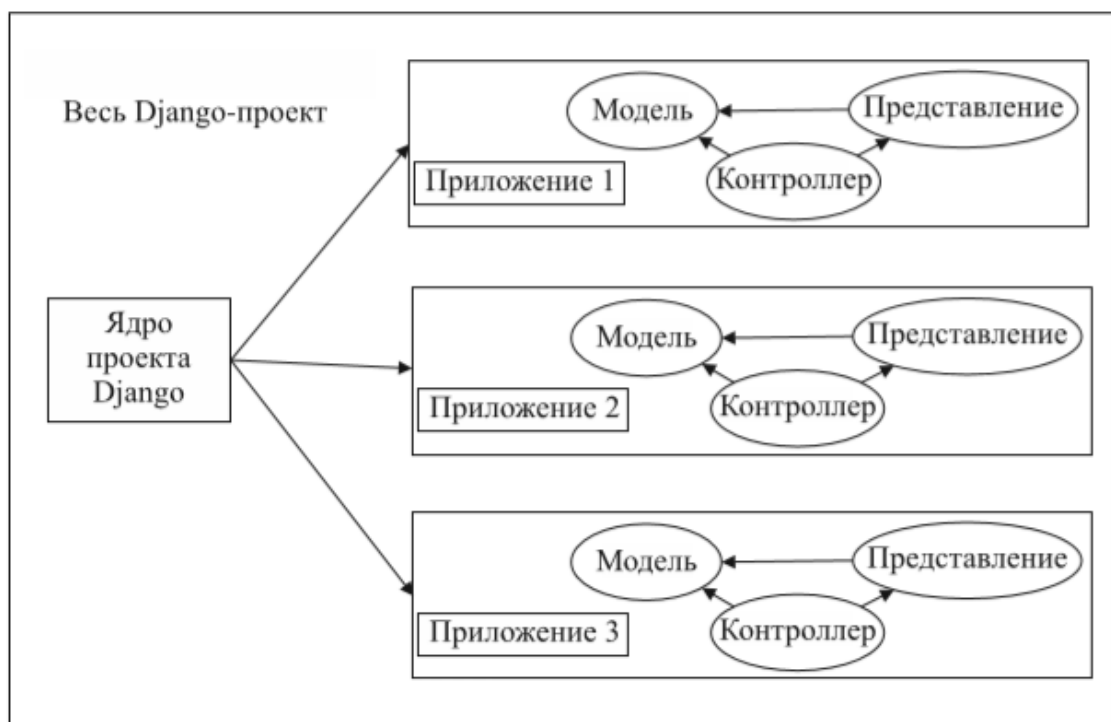


Рисунок 3.1 – Структура базового проекта с тремя приложениями

Предполагается, что в дальнейшем заказчик реализует приложение для блока «Новостей», и блока «Голосование». В реализуемом веб-приложении достаточно на данном этапе будет одного приложения, в котором осуществлен способ разработки, при котором программный код определения и доступа к данным (модель) отделен от логики взаимодействия с приложением (контроллер – необходимые функции и классы), которая в свою очередь отделена от пользовательского интерфейса (представление).

Общая же схема работы приложения на платформе фреймворка представлена на рисунке 3.2.

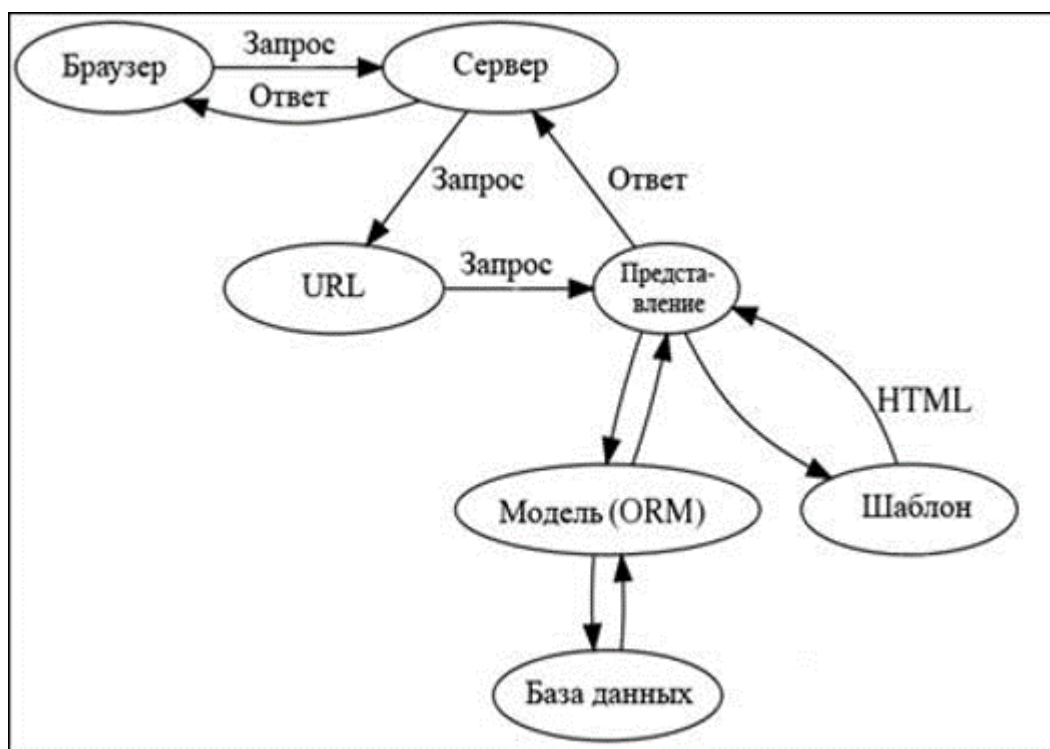


Рисунок 3.2 – Общая схема работы веб-приложения

Конкретная структура проекта разработки веб-приложения перед развертыванием на хостинге представлена на рисунке 3.3.

Как видно из рис. 3.2 архитектура получается трехзвенной, так как есть веб-сервер, фреймворк и сервер БД. Фреймворк делает работу по записи или получения данных из БД, формирует ответ для веб-сервера, который уже отправляет клиенту запрошенные данные.

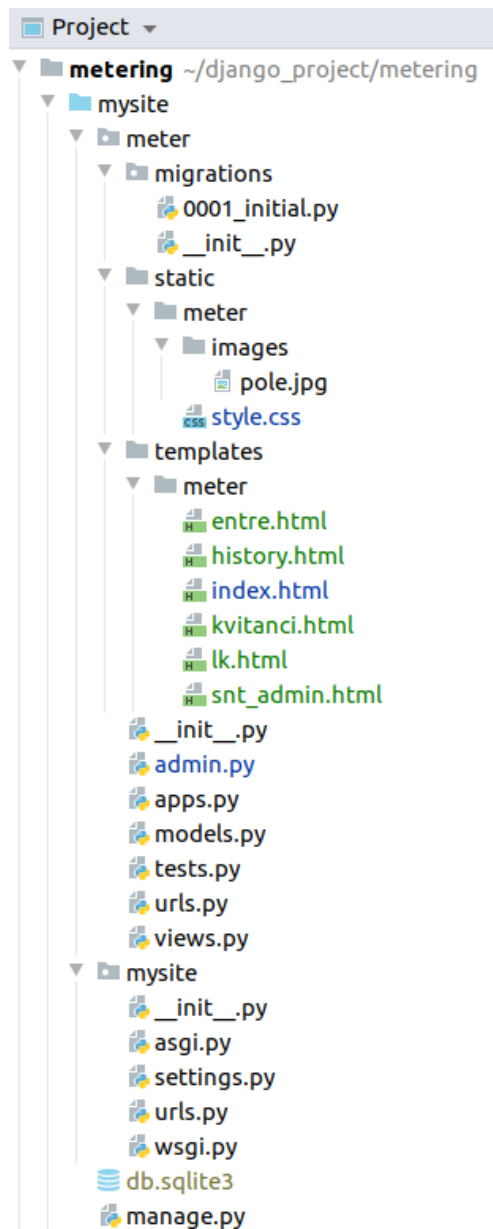


Рисунок 3.3 – Реализованная структура разрабатываемого веб-приложения перед развертыванием на сервере

Нашим приложением в терминологии фреймворка будет папка «meter», в которой и будут расположены файлы с программным кодом, разделенным на программный код, один из которых отвечает за модель данных, другой за представление данных, и нужные для реализации необходимых действий над данными. Также у нас есть папка «templates», где расположены шаблоны необходимые для отображения, папка «static» содержит статические файлы, такие как, например, картинки для визуализации дизайна, в папке «mysite» находятся файлы настроек веб-приложения необходимые для библиотек самого

фреймворка. Это основные папки. В рамках дипломного проекта не представляется возможным в полном объеме описать все файлы и папки веб-приложения и фреймворка.

Программный код основных файлов веб-приложения приведен в Приложении А. Так в «models.py» описаны классы для работы с логической моделью базой данных, описанных во 2 главе, которые фреймворк с помощью своей объектно-реляционной модели (ORM) будет записывать и извлекать данные из БД. А в «views.py» реализованы функции, которые обращаются к ORM извлекают необходимые данные и обращаются к шаблонам представления для подстановки в них нужных данных, и с помощью фреймворка отдают серверу готовое отображение, тот в свою очередь отправляет его клиенту (браузеру).

Тем самым была реализована программная логика работы веб-приложения с использованием выбранного фреймворка, который много работы берет на себя и предоставляет разработчику много готовых инструментов, таких как, например, веб-панель администратора для разработчика. Далее же опишем функциональность, которая реализована программной логикой веб-приложения.

3.2 Описание функциональности веб-приложения

Рассмотрим функциональность нашего веб-приложения. Каждый член СНТ заходя на сайт СНТ, в реализации веб-сервиса это шаблон для отображения index.html, может перейти в личный кабинет имея свой логин и пароль.

Главная страничка веб-приложения изображена на рисунке 3.4. Ввод логина и пароля изображен на рисунке 3.5.

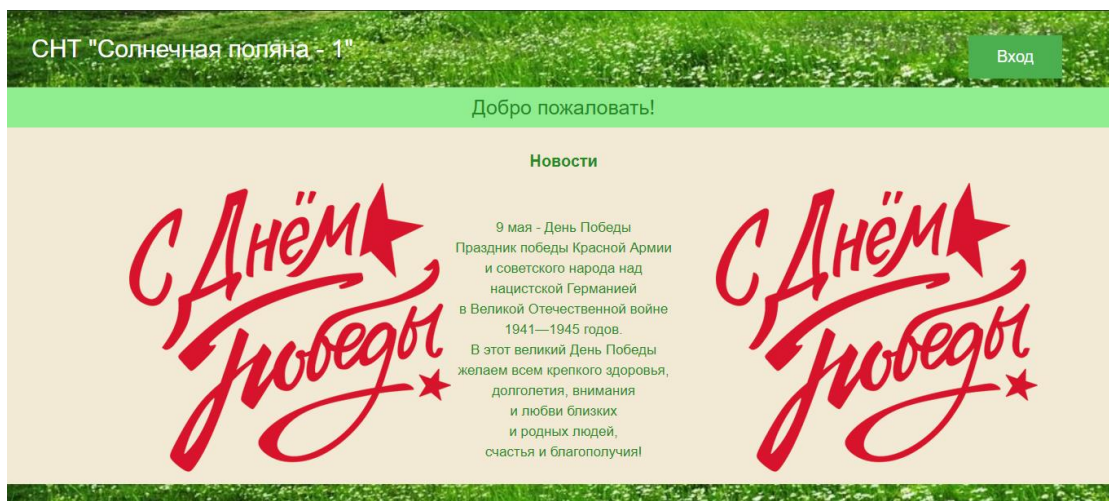


Рисунок 3.4 – Страница главная страница веб-приложения

Авторизация

Логин

Пароль

Войти

Запомнить меня

Отмена [Забыли пароль?](#)

Рисунок 3.5 – Страница ввода логина и пароля

На странице входа помимо полей для введения логина и пароля, есть возможность запомнить для браузера, введенные данные, ссылка на форму восстановления пароля в случае того, если член СНТ забыл свои данные. Логинотом является e-mail члена СНТ, пароль выдается Правлением СНТ.

Член СНТ проходит аутоинтефикацию и авторизацию, и затем получает доступ к личному кабинету, который изображен на рисунке 3.6.

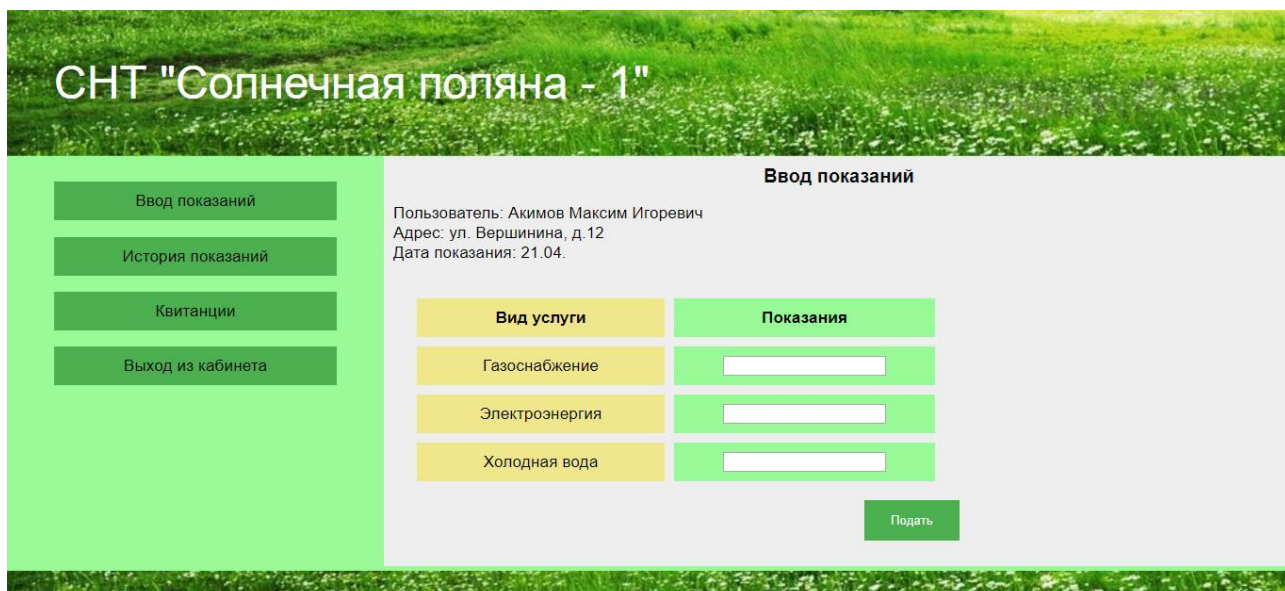


Рисунок 3.6 – Личный кабинет члена СНТ

В личном кабинете имеется четыре раздела: ввод показаний, история показаний, квитанции, выход из кабинета. В кабинете отображаются данные вошедшего в свой кабинет члена СНТ и сразу предлагается ввести показания счетчиков, для этого есть графы «Вид услуги», «Показания» для соответственно ввода показаний приборов учета электроэнергии, газоснабжения и холодной воды. В разделе «Квитанции» (рис. 3.7) можно посмотреть все сформированные квитанции за отчетные периоды по месяцам.

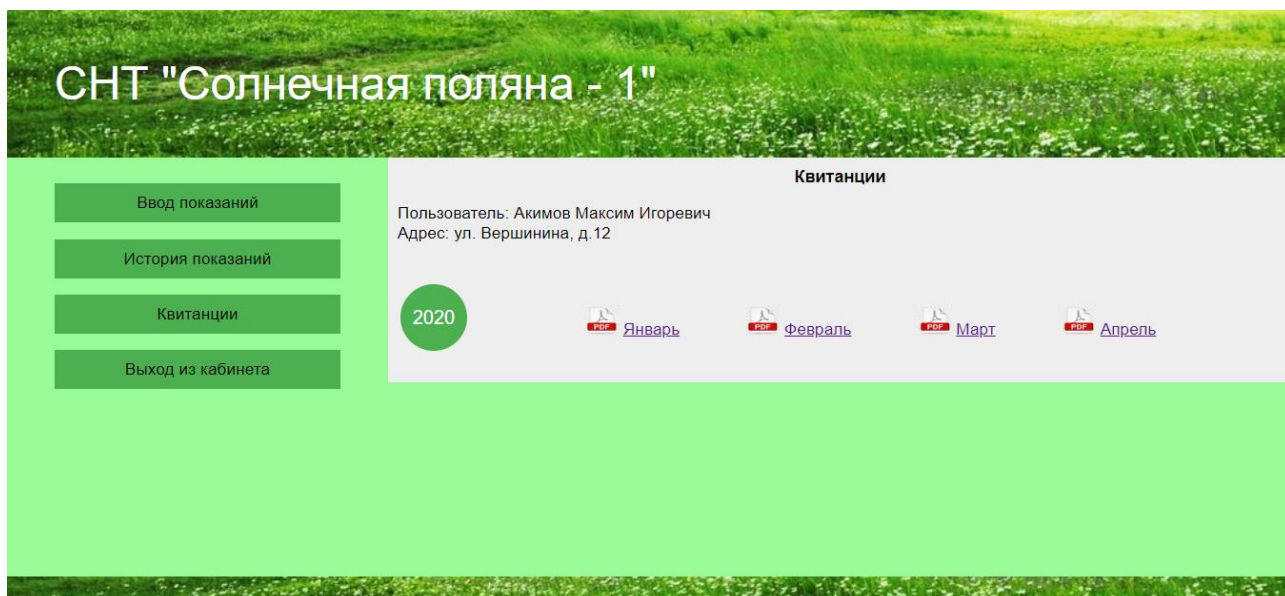


Рисунок 3.7 – страница раздела «Квитанции»

На странице сформированной квитанции (бланк квитанции приведен в Приложении Б) отображается информация о том, сколько было потреблено света, газа и воды с прошлого и текущего месяца, а также указаны общие потери за позапрошлый месяц, которые тоже включены в расчет суммы, которую необходимо оплатить члену СНТ. Квитанцию можно распечатать или загрузить на компьютер, для этого в самом браузере есть такая функция к загружаемым pdf-файлам, которая загрузит сформированный бланк квитанции с необходимой информацией и реквизитами.

На рисунке 3.8 изображена история показаний.

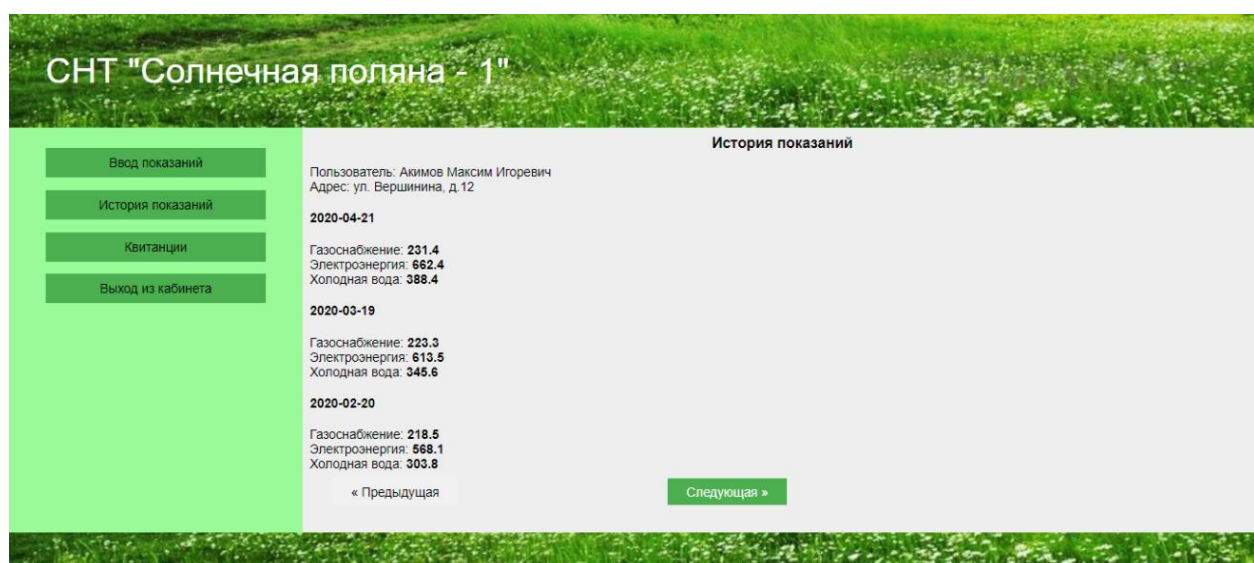
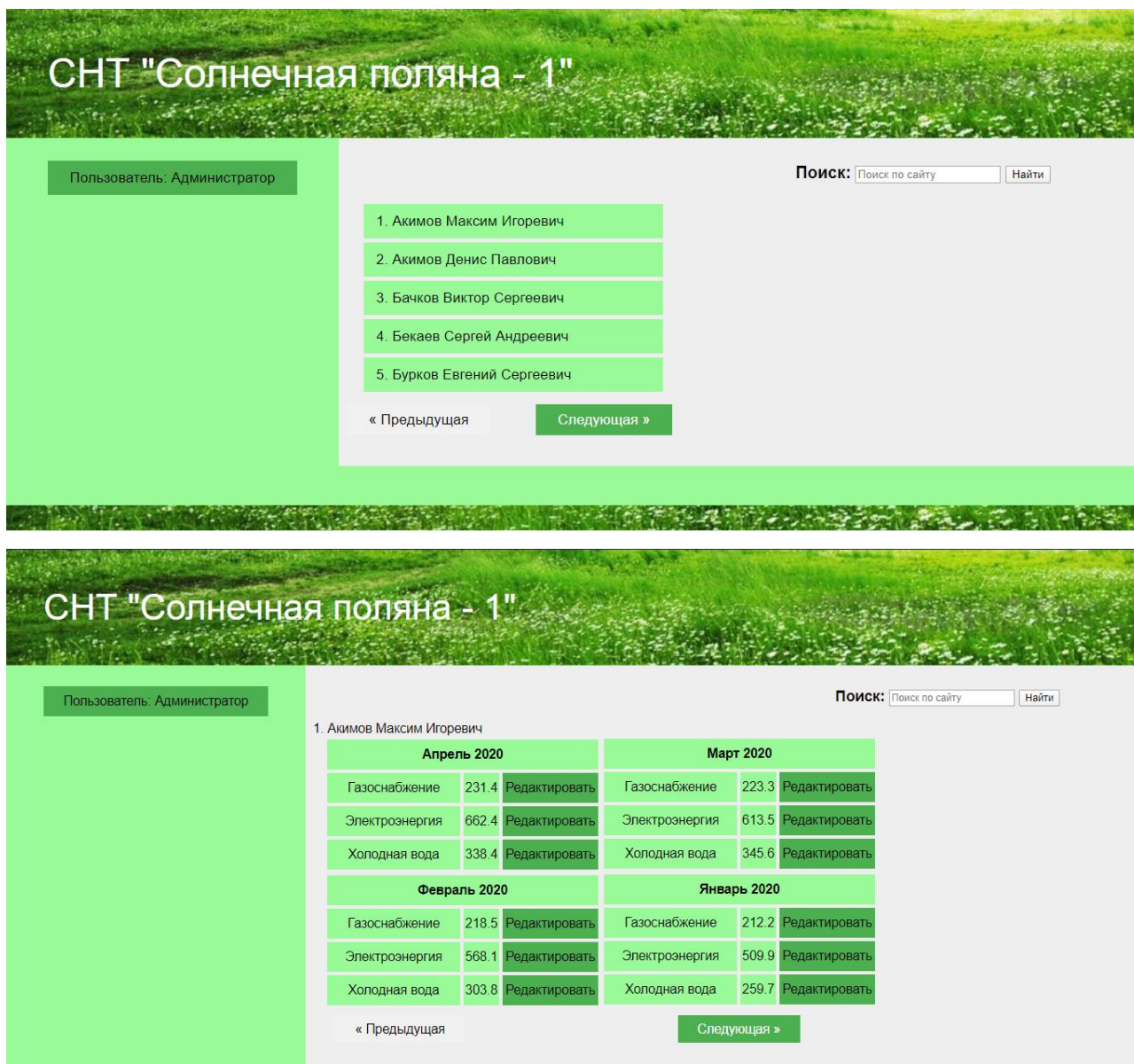


Рисунок 3.8 – История показаний

На рисунке 3.9 изображена панель администратора для редактирования ошибочно введенных данных. То есть таких данных, которые очевидно были записаны по ошибке, то есть, например, слишком большие или слишком маленькие показания. Проверка таких данных осуществляется самими членами Правления на данном этапе жизненного цикла веб-приложения. Заметив такие показания член Правления связывается с членом СНТ и уточняет показания.



Для ввода показаний общих счетчиков член Правления вводит на странице вода общих показаний, которая изображена на рисунке 3.10.

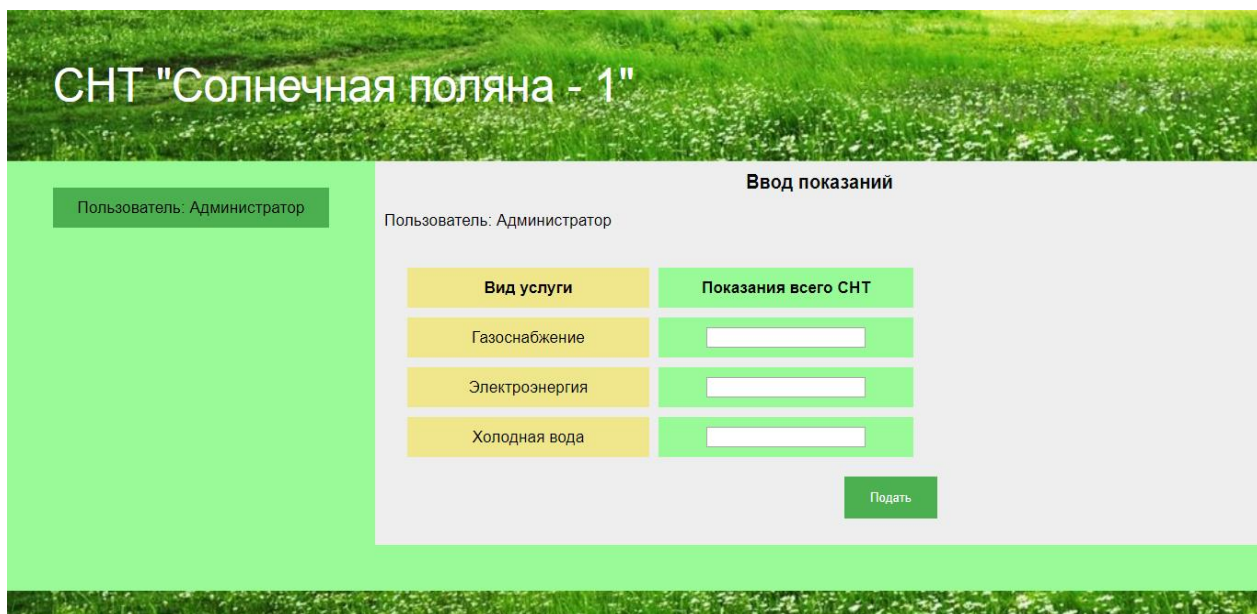


Рисунок 3.10 – Страница ввода общих показаний

На рисунке 3.11 показана страница истории показаний общих счетчиков СНТ.

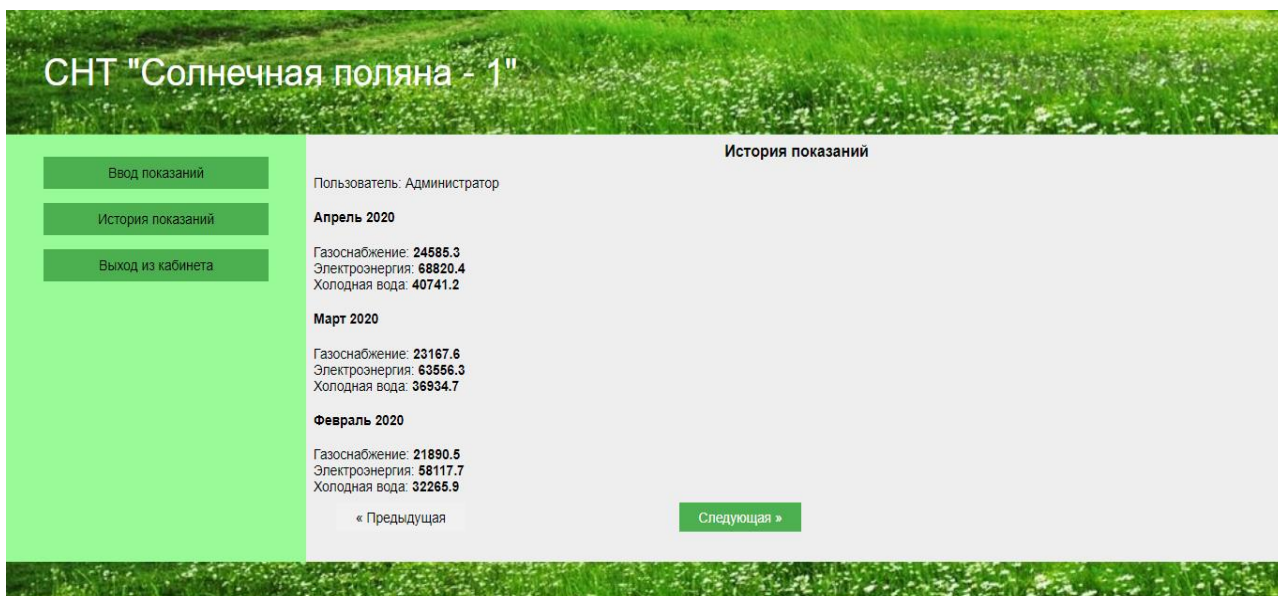


Рисунок 3.11 – Страница истории общих показаний

Так же фреймворк нам предоставляет административную панель для разработчика веб-приложения, которая изображена на рисунке 3.12.

Администрирование сайта

ПОКАЗАНИЯ		
Общие показания	+ Добавить	✎ Изменить
Оплата	+ Добавить	✎ Изменить
Показания	+ Добавить	✎ Изменить
Список пользователей	+ Добавить	✎ Изменить
Тарифы	+ Добавить	✎ Изменить

ПОЛЬЗОВАТЕЛИ И ГРУППЫ		
Группы	+ Добавить	✎ Изменить
Пользователи	+ Добавить	✎ Изменить

Последние действия

Мои действия

- ✎ Тарифы 2020
Оплата

Рисунок 3.12 – Административная панель разработчика веб-приложения

Таким образом, в нашем веб-приложении присутствует вся необходимая функциональность для заказчика, которая и будет осуществлять бизнес-логику для, использующего разработанное веб-приложение, заказчика.

3.3 Оценка эффективности использования веб-приложения

Разработка веб-приложения должна была существенно сократить затраты времени работы персонала, который занимался сбором показаний счетчиков, и выставлением счетов. Для понимания повышения эффективности работы персонала от внедрения веб-приложения рассчитаем показатели трудозатрат.

Без использования веб-приложения персонал тратил около 5 минут времени на внесение данных в электронную таблицу и проведение расчетов суммы к оплате. Еще 5 минут тратил на отправку квитанции члену СНТ. И тратил примерно 5 минут на проверку факта оплаты. После внедрения веб-

приложения персонал теперь тратит только 5 минут на проверку факта оплаты. Членов СНТ 100 человек.

Уменьшение абсолютных временных затрат рассчитывается по формуле 3.1:

$$\Delta T = T_1 - T_2 = 15 - 5 = 10 \text{ мин}, \quad (3.1)$$

где T_1 – это время на работу с показаниями члена СНТ и проверкой оплаты, а T_2 это время работы персонала после внедрения веб-приложения.

Коэффициент относительного снижения рассчитан по формуле 3.2:

$$K_T = \Delta T / T_1 \cdot 100\% = 10/15 \cdot 100\% = 66\% \quad (3.2)$$

Индекс снижения трудовых затрат посчитан по формуле 3.3:

$$Y_T = T_1 / T_2 = 15/5 = 3 \quad (3/3)$$

Таблица 3 – Показатели эффективности использования веб-сервиса

Показатель	Затраты		Абсолютное изменение, мин	Коэффициент изменения	Индекс изменения
	До использования, мин	После использования, мин			
Трудоемкость	T_1	T_2	$\Delta T = T_1 - T_2$	$K_T = \Delta T / T_1$	$Y_T = T_1 / T_2$
	15	5	10	0,66	3

Также можно сделать оценочный расчет сэкономленных средств, так как разработка осуществлялась бесплатно в рамках дипломного проекта, а затраты на хостинг были в рамках бесплатной подписки с ограниченными возможностями – нулевыми. То по формуле (3.5) посчитаем выгоду в денежном выражении. Учитывая, что рабочий час члена правления стоит 60 руб., или 1 рубль в минуту. То

$$S = \Delta T \times R = 10 \times 1 = 10 \text{ руб.} \quad (3.4)$$

$$S_{\text{общ}} = S \times N = 10 \times 100 = 1000 \text{ руб.} \quad (3.5)$$

где R – это поминутная ставка работы члена правления,

S – это выгода от работы по работе с показаниями одно члена СНТ.

N – общее число членов СНТ,

$S_{\text{общ}}$ - это выгода общая в денежном выражении по работе со всеми членами СНТ по внесению показаний за один месяц.

Таким образом видно, что разработанное веб-приложение повышает эффективность работы персонала и экономит минимум 1000 рублей в месяц. Что для СНТ «Солнечная поляна – 1» является значимой суммой, с которой стоит считаться.

Выводы по главе 3

В последней главе было реализовано веб-приложение по сбору показаний счетчиков приборов учета электроэнергии, газа, холодной воды для СНТ «Солнечная поляна – 1» с помощью фреймворка Django, приведен программный код основных компонентов, описана функциональность сервиса, и приведены страницы веб-приложения, которые отображаются для пользователя. Оценена эффективность работы члена Правления при использовании веб-приложения, и от его использования.

Заключение

Неотъемлемой частью деятельности организаций и учреждений стало использование информационных систем, которые позволяют обрабатывать, хранить и использовать информационные ресурсы.

В выпускной квалификационной работе было осуществлено выделение объекта исследования, которым является деятельность правления СНТ «Солнечная поляна – 1», предмета исследования, а именно автоматизация бизнес-процесса финансово-экономической службы по сбору показаний счетчиков приборов учета электроэнергии, газоснабжения и холодной воды. Проведен анализ и декомпозиция бизнес-процессов СНТ «Солнечная поляна – 1», дана краткая характеристика организации. Рассмотрены существующие решения. Построены контекстные диаграммы, показана диаграмма вариантов использования, диаграмма последовательности. Определены требования к аппаратно-программному обеспечению. Разработана логическая и физическая модели веб-приложения, описаны операции расчета суммы к оплате для члена СНТ, выбрана технология разработки (СУБД PostgreSQL, Python3/Django), описана функциональность приложения. Приведен программный код основных компонентов, выполнена программная реализация личного кабинета члена СНТ с внесением показаний счетчиков, просмотром истории показаний, получением квитанции с расчетом необходимой суммы к оплате.

Оценена эффективность работы по использованию веб-приложения, оценена выгода от его работы для СНТ «Солнечная поляна – 1».

Таким образом, была достигнута цель по разработке веб-приложения, которое позволило автоматизировать сбор показаний счетчиков для СНТ «Солнечная поляна – 1». Также повысило надежность хранения информации о деятельности садового некоммерческого товарищества, о его внутренних процессах и снизило временные трудозатраты финансово-экономической службы СНТ.

Список используемой литературы

1. Бейли, Л. Изучаем SQL / Л. Бейли – СПб. : Питер, 2012. – 573с.
2. Вигерс К. Разработка требований к программному обеспечению. 3-е изд., дополнительное / К. Вигерс, Д. Битти., Пер. с англ. – М.: Издательство «Русская редакция»; СПб. : БХВ-Петербург, 2014. – 736 стр.
3. Гагарина Л. Г., Киселев Д. В., Федотова Е. Л. Разработка и эксплуатация автоматизированных информационных систем М. : ИД «ФОРУМ» ; ИНФРА-М, 2012. - 384с.
4. Грофф Д. SQL: Полное руководство / Д. Грофф, П. Вайнберг. / - К. : BHV, 2001. - 816с. : ил.
5. Дейт Д. Введение в системы баз данных / Д. Дейт. / — М. : Издательский дом "Вильямс", 2005. — 1328 с. : ил
6. Диаграмма потоков данных (DFD). Графический язык диаграммы. Примеры / [Электронный ресурс]. – URL: <http://e-educ.ru/bd14.html/> (дата обращения: 08.03.2020).
7. Документация к PostgreSQL 12.2 / [Электронный ресурс]. – URL: <https://postgrespro.ru/media/docs/postgresql/12/ru/postgres-A4.pdf> (дата обращения: 11.04.2020).
8. Дудина, И.П. Рекомендации по выполнению выпускной квалификационной работы бакалавра по направлению подготовки «Прикладная информатика»: учеб. - метод. пособие / И.П. Дудина, О.М. Гущина, С.В. Мкртычев. – Тольятти: Изд-во ТГУ, 2013. – 59 с. : обл.
9. Емельянова, Н.З. Проектирование информационных систем: учебное пособие / Н.З. Емельянова, Т.Л. Партыка, И.И. Попов. - М. : Форум, 2014. - 432 с.
10. Ермаков Р.Н. Описание отдельных концепций IDEF0 [Электронный ресурс]. –URL: <http://www/idef0/ru/idef0/> (дата обращения: 19.03.2020).
11. Информатика: учеб.-метод. пособие для студентов вузов: в 2 ч. / В. Ф. Глазова, А. В. Богданова, Е. В. Панюкова; ТГУ; Ин-т математики, физики и ин-

формационных технологий; каф. «Информатика и вычислительная техника». – ТГУ; Гриф УМО. – Тольятти: ТГУ, 2013. – Ч.1. – 242 с.

12. Колесов, Ю. Б. Моделирование систем. Объектно-ориентированный подход: учебное пособие / Ю. Б. Колесов, Ю. Б. Сениченков. – СПб. : БХВ-Петербург, 2012. – 192 с.

13. Конноли Т. Базы данных. Проектирование, реализация и сопровождение. Теория и практика / Т. Конноли, К. Бегг, А. Стратчан. / – М. : Вильямс, 2000. - 1093 с.

14. Корнеев, И. К. Информационные технологии: учебное пособие / И. К. Корнеев. / - М: Проспект, 2007.- 224 с. 9.

15. Куроуз Д. Компьютерные сети. Многоуровневая архитектура Интернета: 2-е издание / Д. Куроуз, К. Росс. / – СПб. : Питер, 2004. – 765с.

16. Лабберс П. HTML5 для профессионалов: мощные инструменты для разработки современных веб-приложений / П. Лабберс, Б. Олаберс. / – М. : Вильямс, 2011. – 272с.

17. Лоусон Б. Изучаем HTML5. Библиотека специалиста. / Б. Лоусон, Р. Шарп. / – СПб. : Питер, 2012. – 286с.

18. Мкртычев, С. В. Информационные системы в социальном менеджменте: учеб. пособие / С. В. Мкртычев ; ТГУ ; Ин-т математики, физики и информационных технологий ; каф. «Информатика и вычислительная техника». – ТГУ. – Тольятти: ТГУ, 2012. – 78 с.

19. Осипов Д. Л. Технологии проектирования баз данных. – М. : ДМК Пресс, 2019. – 498 с. : ил.

20. Особенности концептуального моделирования предметной области / [Электронный ресурс]: <https://habrahabr.ru/post/256823/>, (дата обращения: 19.03.2020).

21. Продукты Visio и Project / MicrosoftCorporation [Электронный ресурс]. – URL: <https://www.microsoft.com/ru-ru/office/vip/visio.aspx/>, Режим доступа: свободный (13.04.2020).

22. Проектирование информационных систем // II Основы методологии проектирования информационных систем // п. 2.3. Содержание и организация проектирования / [Электронный ресурс]. – URL: <http://webmath.exponenta.ru/db/02.html> (дата обращения: 06.03.2020).

23. Создание логической модели / [Электронный ресурс]. – URL: <http://itteach.ru/bpwin/sozдание-logicheskoy-modeli>, (дата обращения: 06.03.2020).

24. Устав садового некоммерческого товарищества «Солнечная поляна – 1» от 19.08.2003г. №1.

25. Шелухин О. И. Моделирование информационных систем : учеб. пособие. 004 / О. И. Шелухин. - 2-е изд., перераб. и доп. - Москва : Горячая линия - Телеком, 2012. - 516 с. : ил. - ISBN 978-5- 9912-0193-3.

26. Яшин, С.Н. Анализ эффективности инновационной деятельности: учеб. Пособие / С.Н. Яшин, Е.В. Кошелев, С.А. Макаров. – СПб. : БХВ-Петербург, 2012. – 288 с.

27. PostgreSQL. Основы языка SQL: учеб. пособие / Е. П. Моргунов; под ред. Е. В. Рогова, П. В. Лузанова. — СПб. : БХВ-Петербург, 2018. — 336 с. : ил.

28. Aaron A. HTML 5 & CSS3 Genius Guide Volume 2 – Imagine Publishing Ltd 2015.

29. Cederholm D. CSS3 for Web Designers – A Book Apart 2014.

30. Eric A. Meyer: Transforms in CSS: Revamp the Way You Design – O`Reilly Media, 2015.

31. PostgreSQL Reference Manual - Volume 1: SQL Language Reference the PostgreSQL Global Development Group, 2007.

32. Riaz Ahmed, SQL – The Shortest Route for Beginners, CreateSpace Independent Publishing Platform, 2015.

33. William S. Vincent, Django for APIs: Build web APIs with Python and Django, 2018.

Приложение А

Программный код

```
models.py X
1 from django.db import models
2 class UsersSNT(models.Model):
3     login_user = models.EmailField(max_length=254, verbose_name='Логин')
4     parole_user = models.PositiveSmallIntegerField(verbose_name='Пароль')
5     created_at = models.DateTimeField(auto_now_add=True, verbose_name='Дата создания')
6     name_user = models.CharField(max_length=254, db_index=True, verbose_name='Имя члена СНТ')
7
8     def __str__(self):
9         return self.name_user
10
11     class Meta:
12         verbose_name = 'Пользователи'
13         verbose_name_plural = 'Список пользователей'
14         ordering = ['name_user']
15
16 class Meter(models.Model):
17     user = models.ForeignKey(UsersSNT, on_delete=models.PROTECT, verbose_name='Пользователь')
18     when_rec = models.DateTimeField(auto_now_add=True, verbose_name='Дата внесения')
19     updated_at = models.DateTimeField(auto_now=True, verbose_name='Обновлено')
20     gas = models.DecimalField(max_digits=5, decimal_places=2, verbose_name='Газ')
21     light = models.DecimalField(max_digits=5, decimal_places=2, verbose_name='Свет')
22     water = models.DecimalField(max_digits=5, decimal_places=2, verbose_name='Вода')
23
24     class Meta:
25         verbose_name = 'Показания'
26         verbose_name_plural = 'Показания'
27         ordering = ['-when_rec']
28
29 class Rate(models.Model):
30     created_at = models.DateTimeField(auto_now_add=True, verbose_name='Дата')
31     r_gas = models.DecimalField(max_digits=5, decimal_places=2, verbose_name='Тариф за Газ')
32     r_light = models.DecimalField(max_digits=5, decimal_places=2, verbose_name='Тариф за Свет')
33     r_water = models.DecimalField(max_digits=5, decimal_places=2, verbose_name='Тариф за Вода')
34
35     class Meta:
36         verbose_name = 'Тариф'
37         verbose_name_plural = 'Тарифы'
38         ordering = ['-created_at']
39
40 class UnitValue(models.Model):
41     when_rec = models.DateTimeField(auto_now_add=True, verbose_name='Дата внесения')
42     updated_at = models.DateTimeField(auto_now=True, verbose_name='Обновлено')
43     gas = models.DecimalField(max_digits=5, decimal_places=2, verbose_name='Газ')
44     light = models.DecimalField(max_digits=5, decimal_places=2, verbose_name='Свет')
45     water = models.DecimalField(max_digits=5, decimal_places=2, verbose_name='Вода')
46
47     class Meta:
48         verbose_name = 'Показания'
49         verbose_name_plural = 'Общие показания'
50         ordering = ['-when_rec']
51
52 class Payment(models.Model):
53     user = models.ForeignKey(UsersSNT, on_delete=models.PROTECT, verbose_name='Пользователь')
54     dated_at = models.DateTimeField(auto_now=True, verbose_name='Месяц')
55     total = models.DecimalField(max_digits=7, decimal_places=2, verbose_name='Сумма к оплате')
56     Check = models.BooleanField(default=True, verbose_name='Оплачено')
57     View_list = models.TextField(blank=True, verbose_name='Комментария')
58     def __str__(self):
59         return self.user
60     class Meta:
61         verbose_name = 'Оплата'
62         verbose_name_plural = 'Оплата'
63         ordering = ['-created_at']
```

Продолжение Приложения А

```
views.py x
1 from django.shortcuts import render
2 from django.http import HttpResponseRedirect
3 from django.template import loader
4 from .models import UnitValue, Rate, Meter, UsersSNT, News
5 def index(request):
6     listing = News.objects.all()
7     users_list = UsersSNT.objects.all()
8     template = loader.get_template('meter/index.html')
9     context = {'listing': listing,
10              'users_list': users_list,}
11     return render(request, 'meter/index.html', context)
12 def entre(request):
13     if request.method == 'POST':
14         form = UserRegisterForm(request.POST)
15         if form.is_valid():
16             user = form.save()
17             login(request, user)
18             messages.success(request, 'Вход')
19             return redirect('index')
20         else:
21             messages.error(request, 'Ошибка')
22     else:
23         form = UserRegisterForm()
24     return render(request, 'meter/entre.html', context, {'form': form})
25 def user_login(request):
26     if request.method == 'POST':
27         form = UserLoginForm(data=request.POST)
28         if form.is_valid():
29             user = form.get_user()
30             login(request, user)
31             return redirect('index')
32     else:
33         form = UserLoginForm()
34     return render(request, 'meter/entre.html', {'form': form})
35 def lk(request):
36     if request.method == 'POST':
37         form = LkForm(request.POST)
38         if form.is_valid():
39             lk = form.save()
40             return redirect(lk)
41     else:
42         form = LkForm()
43     return render(request, 'meter/snt_admin.html', {'form': form})
44 def user_logout(request):
45     logout(request)
46     return redirect('login')
47 def history(request):
48     listing = Meter.objects.all()
49     s_list = UsersSNT.objects.all()
50     template = loader.get_template('meter/history.html')
51     context = {'listing': listing,
52              's_list': s_list,}
53     return render(request, 'meter/history.html', context)
54 def kvitanci(request):
55     kvitanci = Payment.objects.all()
56     s_list = UsersSNT.objects.all()
57     template = loader.get_template('meter/kvitanci.html')
58     context = {'kvitanci': kvitanci,
59              's_list': s_list,}
60     return render(request, 'meter/kvitanci.html', context)
61 def snt_admin(request):
62     if request.method == 'POST':
63         form = LkForm(request.POST)
64         if form.is_valid():
65             snt_admin = form.save()
66             return redirect(snt_admin)
67     else:
68         form = LkForm()
69     return render(request, 'meter/snt_admin.html', {'form': form})
```

Продолжение Приложения А

```
settings.py ✕
1  import os
2
3  SECRET_KEY = 'loab#0+2it86%wstz+kt7f(cyoq_#v5w(6vbyr#p8#73g8ka@d'
4  DEBUG = True
5  ALLOWED_HOSTS = []
6
7  INSTALLED_APPS = [
8      'django.contrib.admin',
9      'django.contrib.auth',
10     'django.contrib.contenttypes',
11     'django.contrib.sessions',
12     'django.contrib.messages',
13     'django.contrib.staticfiles',
14     'meter.apps.MeterConfig'
15 ]
16
17 MIDDLEWARE = [
18     'django.middleware.security.SecurityMiddleware',
19     'django.contrib.sessions.middleware.SessionMiddleware',
20     'django.middleware.common.CommonMiddleware',
21     'django.middleware.csrf.CsrfViewMiddleware',
22     'django.contrib.auth.middleware.AuthenticationMiddleware',
23     'django.contrib.messages.middleware.MessageMiddleware',
24     'django.middleware.clickjacking.XFrameOptionsMiddleware',
25 ]
26
27 ROOT_URLCONF = 'mysite.urls'
28
29 TEMPLATES = [
30     {
31         'BACKEND': 'django.template.backends.django.DjangoTemplates',
32         'DIRS': [os.path.join(BASE_DIR, 'templates')],
33         'APP_DIRS': True,
34         'OPTIONS': {
35             'context_processors': [
36                 'django.template.context_processors.debug',
37                 'django.template.context_processors.request',
38                 'django.contrib.auth.context_processors.auth',
39                 'django.contrib.messages.context_processors.messages',
40             ],
41         },
42     ],
43 ]
```

Продолжение Приложения А

```
settings.py
4
5     WSGI_APPLICATION = 'mysite.wsgi.application'
6
7     DATABASES = {
8         'default': {
9             'ENGINE': 'django.db.backends.sqlite3',
10            'NAME': os.path.join(BASE_DIR, 'db.sqlite3'),
11        }
12    }
13
14     AUTH_PASSWORD_VALIDATORS = [
15         {
16             'NAME': 'django.contrib.auth.password_validation.UserAttributeSimilarityValidator',
17         },
18         {
19             'NAME': 'django.contrib.auth.password_validation.MinimumLengthValidator',
20         },
21         {
22             'NAME': 'django.contrib.auth.password_validation.CommonPasswordValidator',
23         },
24         {
25             'NAME': 'django.contrib.auth.password_validation.NumericPasswordValidator',
26         },
27     ]
28
29     LANGUAGE_CODE = 'ru'
30
31     TIME_ZONE = 'UTC'
32
33     USE_I18N = True
34
35     USE_L10N = True
36
37     USE_TZ = True
38
39     # Static files (CSS, JavaScript, Images)
40     # https://docs.djangoproject.com/en/3.0/howto/static-files/
41
42     STATIC_URL = '/static/'
43
44
```


Приложение Б

Пример квитанции

САДОВОЕ НЕКОММЕРЧЕСКОЕ ТОВАРИЩЕСТВО «СОЛНЕЧНАЯ ПОЛЯНА – 1» СЧЕТ-КВИТАНЦИЯ ОБ ОПЛАТЕ за Апрель 2020						КВИТАНЦИЯ	
Платательщик: Акимов Максим Игоревич Адрес: ул. Вершинина д. 12 Начислено: 418.12 руб. – к оплате						Получатель: САДОВОЕ НЕКОММЕРЧЕСКОЕ ТОВАРИЩЕСТВО «СОЛНЕЧНАЯ ПОЛЯНА – 1» Адрес: 606304, Нижегородская обл, село Тепло-Троицкое, район Дальнеконстантиновский, дом 8 ИНН 5215001606 КПП 521501001 р/с: 40503610756033000004 Банк: Волго-Вятский банк ПАО Сбербанк. БИК 042202603 Кор. счет: 30101810900000000603 ОГРН 108525003585, ОКПО 89376608, ОКТМО 22630412146, ОКВЭД 68.32.2	
Вид расчета	Тариф	Объем услуг	Ед. изм.	Общие потери за позапрошлый месяц	Начислено по тарифу	Начислено по общим потерям за позапрошлый месяц	Сумма оплаты
Газоснабжение	2.9	8.1	м ³	0.4	23.49	1.16	24.65
Электроэнергия	4.1	48.9	кВт/ч	0.6	200.49	2.46	202.95
Холодное водоснабжение	4.4	42.6	м ³	0.7	187.44	3.08	190.52
ИТОГО:					411.42	6.7	418.12
ТОГО за Апрель 2020:							418.12 руб.