**Тема 2. Ознакомление со структурой и характером деятельности предприятия**

Практика проходит на предприятии «ООО Главстрой-СПб специализированный застройщик» которая является одной из крупнейших строительных компаний Северо-Западного региона РФ. Структура компании представлена на рисунке 1.

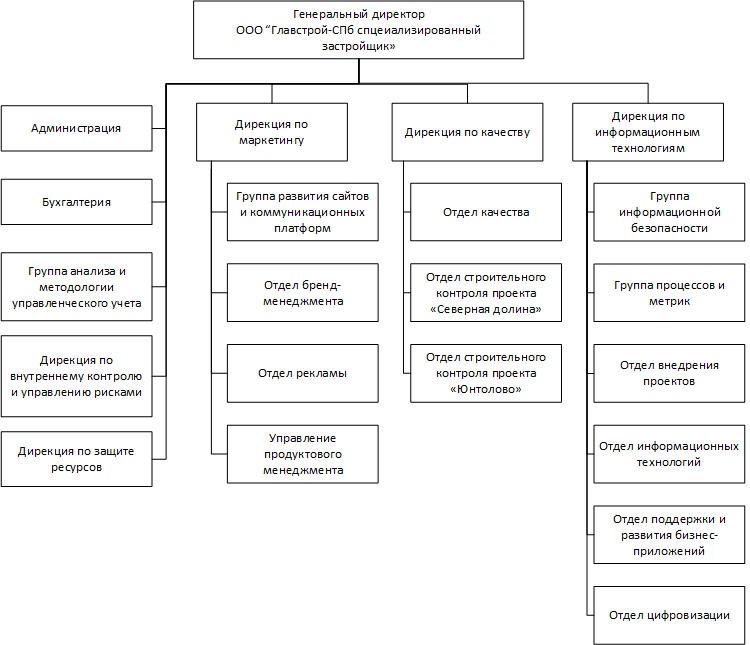


Рисунок 1 – Структура предприятия

Компания имеет линейно-функциональную структуру управления, где каждый отдел выполняет свои определённые функции. Отделы взаимодействуют друг с другом через функциональные связи. Такая структура позволяет обеспечить чёткое распределение ответственности между сотрудниками и эффективное управление процессами.

Во главе компании стоит генеральный директор в лице Сурвилло Татьяны Геннадиевны. В компании есть множество подразделений, таких как администрация, бухгалтерия, дирекции по маркетингу, качеству, защите ресурсов и т. д. Во главе каждого из подразделений стоит директор. Некоторые подразделения делятся на отделы. Например, дирекция по информационным технологиям делится на 6 отделов, в числе которых отдел цифровизации, в котором проходит данная практика.

«ООО Главстрой-СПб специализированный застройщик» по характеру деятельности является строительной компанией. Она занимается реализацией проектов жилищного строительства в сегментах комфорт- и бизнес-класса. География присутствия компании охватывает все районы Санкт-Петербурга и Ленинградскую область.

Основными направлениями деятельности компании являются:

* Строительство жилых комплексов комфорт- и бизнес-класса;
* Реализация объектов коммерческой недвижимости;
* Разработка проектной документации для строительства зданий и сооружений различного назначения.

В настоящее время компания реализует ряд крупных проектов жилищного строительства в различных районах Санкт-Петербурга и Ленинградской области. Среди наиболее значимых проектов можно выделить: ЖК «Северная долина», ЖК «Юнтолово», ЖК «Панорама 360».

**Тема 3. Сбор материалов для составления технического задания по теме дипломного проекта**

3.1 Анализ деятельности предприятия

ООО «Главстрой-СПб специализированный застройщик» – это строительная компания, специализирующаяся на разработке и реализации строительных проектов в городе Санкт-Петербурге и его окрестностях. Компания была основана в 2006 году и с тех пор успешно развивается, реализуя проекты жилищного строительства в сегментах комфорт- и бизнес-класса. Предприятие занимается возведением жилых и коммерческих объектов, а также инфраструктурных сооружений.

География деятельности компании охватывает все районы Санкт-Петербурга и Ленинградскую область. Основные направления деятельности включают строительство жилых комплексов комфорт- и бизнес-класса, реализацию объектов коммерческой недвижимости и разработку проектной документации для строительства зданий и сооружений различного назначения.

Так как компания занимается также и обслуживанием реализованных объектов, то для повышения качества взаимодействия с клиентами руководством было принято решение о разработке программного модуля по интеграции в бизнес-процессы компании сервиса голосовых рассылок. Подобное программное решение может быть полезно во многих сценариях, например:

* Уведомления о задолженности: Сервис голосовых рассылок может использоваться для автоматического уведомления жильцов о задолженности по оплате коммунальных услуг. Это поможет управляющей компании сократить количество просроченных платежей и улучшить финансовое состояние организации.
* Информирование о проведении ремонтных работ: Сервис голосовых рассылок может использоваться для информирования жильцов о проведении ремонтных работ в доме или на территории комплекса. Это позволит избежать конфликтов между управляющей компанией и жителями из-за неожиданного шума или других неудобств.
* Организация опросов и голосований: Сервис голосовых рассылок может использоваться для проведения опросов среди жителей дома или комплекса по различным вопросам (например, опрос об удовлетворенности качеством обслуживания).

Таким образом, интеграция сервиса голосовых рассылок будет являться эффективным инструментом для компании, который поможет улучшить коммуникацию с клиентами и повысить качество обслуживания.

3.2 Анализ программных средств, использованных для разработки

Для разработки проекта был выбран набор программных средств, включающий в себя интегрированную среду разработки (IDE), систему управления базами данных, систему контроля версий, инструменты тестирования.

В качестве системы управления базами данных была выбрана Microsoft SQL Management Studio. Данный инструмент разработан специально для работы с Microsoft SQL Server, используемом в качестве сервера баз данных на предприятии, что обеспечивает высокую совместимость и эффективность при работе с этой базой данных. Также, SSMS предоставляет широкий набор инструментов для администрирования баз данных, написания запросов, создания отчетов и управления безопасностью. Среда предоставляет инструменты для мониторинга производительности баз данных, выявления узких мест и оптимизации запросов, что позволяет повысить эффективность работы с базой данных.

В качестве языка разработки был выбран Python, а в качестве интегрированной среды разработки – PyCharm. Python прост в использовании и позволяет вести разработку с высокой скоростью, а так же отлично подходит для веб-разработки. У Python множество библиотечных инструментов, пакетов услуг и удобных фреймворков, в частности – flask, легкий и мощный фреймворк для разработки веб-приложений, который будет использоваться в разрабатываемом проекте. В свою очередь PyCharm, как среда разработки на Python, предлагает широкий набор мощных функций и инструментов, которые помогут увеличить производительность и улучшить качество вашего кода. Например, автодополнение кода, подсветка синтаксиса, отладчик, рефакторинг кода и многое другое. PyCharm интегрируется с различными инструментами и сервисами, такими как системы контроля версий (например, Git), виртуальные окружения, базы данных, фреймворки и т.д. Это облегчает работу с различными аспектами разработки.

В качестве системы контроля версий на сервере компании развернут Bonobo Git Server. Это простой и лёгкий git-сервер, являющийся отличным решением для управления git-репозиториями в организации. Также он является бесплатным и имеет открытый исходный код, что позволяет изменять и адаптировать его под свои потребности.

В качестве инструмента для тестирования бэкенда (в частности, для работы с API сервиса «Звонобот») был выбран Postman. Данный инструмент предлагает бесплатную версию со множеством полезных функций, которые покрывают большинство потребностей при тестировании API. Postman позволяет организовывать запросы в коллекции, что упрощает их использование и повторное использование, создавать и выполнять автотесты, проверять ответы на соответствие ожидаемым результатам и мониторить производительность API.

3.3 Описание этапов разработки

На данный момент, в компании уже начата разработка модуля для взаимодействия с сервисом «Звонобот». Доработка данного модуля будет включать следующие этапы:

1. Доработка веб-интерфейса исходя из требований заказчиков: внесение изменений в фронтенд проекта согласно требованиями. Работы на данном этапе будет включать в себя добавление/удаление полей на страницах веб-интерфейса, реализацию нового функционала, в том числе добавление новых веб-страниц в проект.
2. Рефакторинг кода бэкенда имеющегося сервиса и исправление багов в нем: улучшение структуры и читаемости кода, оптимизация производительности, устранение возможных проблем и улучшение архитектуры приложения.
3. Изучение и предоставление описания части API функционала голосового сервиса «Звонобот», отвечающую за создание и управление роботизированными диалоговыми прозвонами: изучение соответствующих эндпоинтов API сервиса «Звонобот», предоставление описания их параметров, возвращаемых результатов и других особенностей взаимодействия.
4. Анализ решений в области синтеза речи с целью дальнейшего внедрения наиболее зарекомендовавших себя в работу сервиса: изучение сервисов, предоставляющих возможность синтеза речи из текста, для создания с их помощью аудиороликов, которые в дальнейшем будут использоваться в роботизированных звонках.
5. Интеграция функции роботизированного диалогового прозвона в имеющийся сервис перезвонов: добавление требуемых заказчиком функций и сценариев диалоговых звонков на основании изученного API-функционала, обработка результатов прозвонов.

**Тема 4. Разработка программного обеспечения на основе технического задания по теме дипломного проекта**

4.1 Доработка веб-интерфейса

Структура и логика работы веб-интерфейса, который имеется на данный момент, представлены на wireframe-диаграмме (приложение А). Как понятно из неё, интерфейс имеет проблемы с навигацией (нет возможности возвращаться на предыдущие страницы), в нем отсутствуют некоторые страницы (например, нет страницы с результатами прозвона), а также требуется внесение некоторых изменений в соответствии с требованиями заказчика.

4.1.1 Страницы выбора типа обзвона

Для обеспечения навигации по интерфейсу, на страницу выбора типа обзвона добавлена кнопка «выйти», по нажатию на которую пользователь может вернуться на страницу авторизации. Итоговый вариант страницы представлен на рисунке 3.

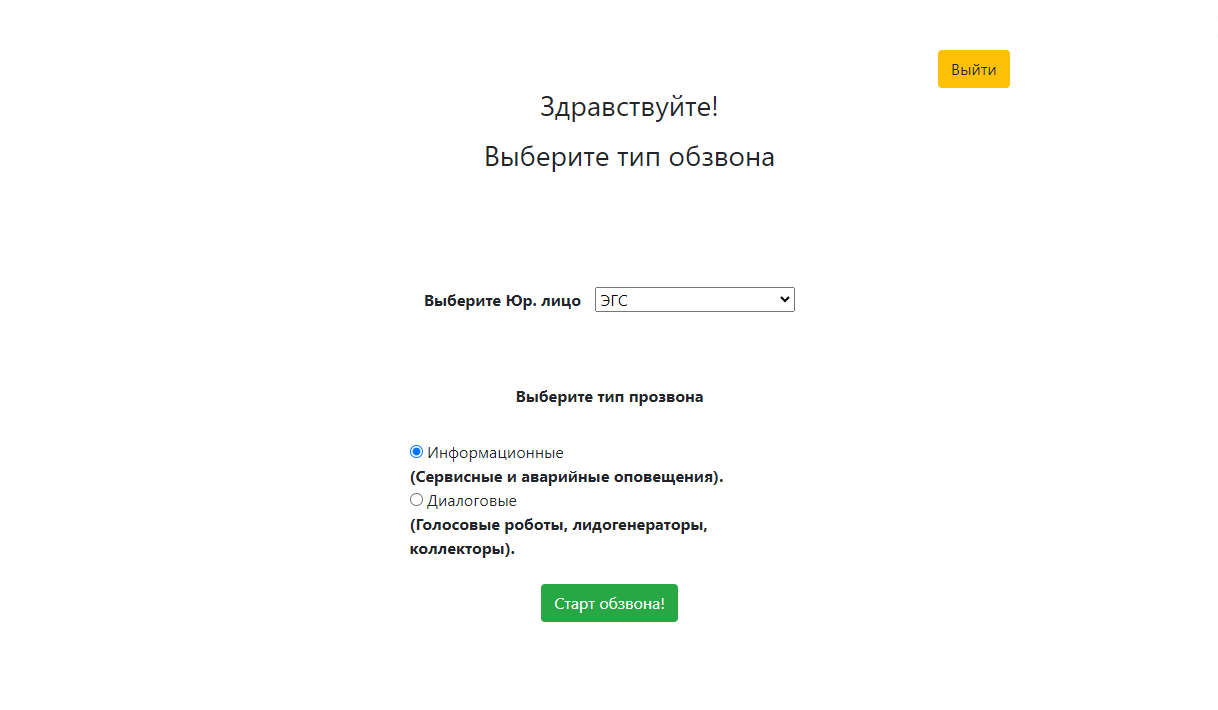


Рисунок 3 – Страница выбора типа обзвона

4.1.2 Страница информационных прозвонов

Сервис «звонобот» позволяет для голосовых рассылок использовать либо уже созданный и сохраненный в личном кабинете аудиоролик, либо текст, который будет преобразован сервисом в голос. Таким образом, на странице информационных прозвонов в выпадающем списке пользователю необходимо выбрать тип рингтона («Текст» / «ID из звонобота»). Был добавлен скрипт, реализующий логику отображения полей ввода ID рингтона или текста в зависимости от выбранной опции (изначально оба поля отображались всегда). Код скрипта представлен на рисунке 4.

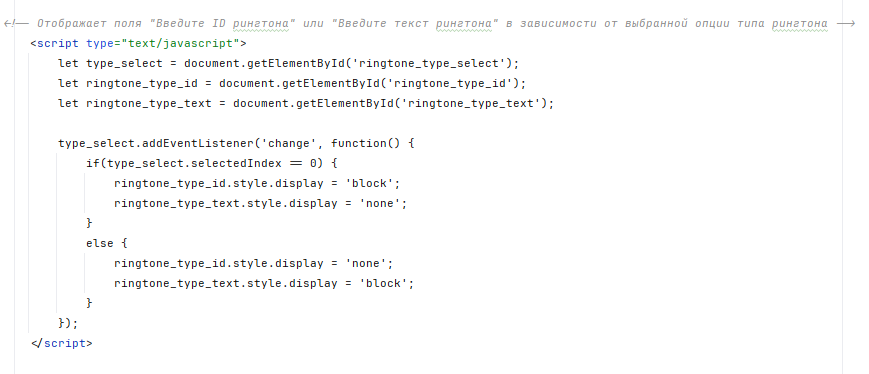


Рисунок 4 – Скрипт для отображения полей «Текст» / «ID из звонобота»

В соответствии с требованиями заказчика, в секции «что считать дозвоном» был оставлен только пункт «прослушал более N секунд ролика», и убран чекбокс «автоответчик». На страницу добавлено отображение баланса на счету личного кабинета в сервисе «звонобот».

Было замечено, что загрузка страницы занимает продолжительное время – около 10 секунд. Это происходит из-за получения исходящих номеров телефонов из «звонобота». До тех пор, пока все номера не будут получены, страница не будет отображена. Для исправления этой проблемы логика работы сайта была изменена следующим образом: страница теперь возвращается мгновенно с пустым списком, и посылает Ajax-запрос на получение номеров (код скрипта, реализующего данную логику на стороне клиента представлен на рисунке 5).

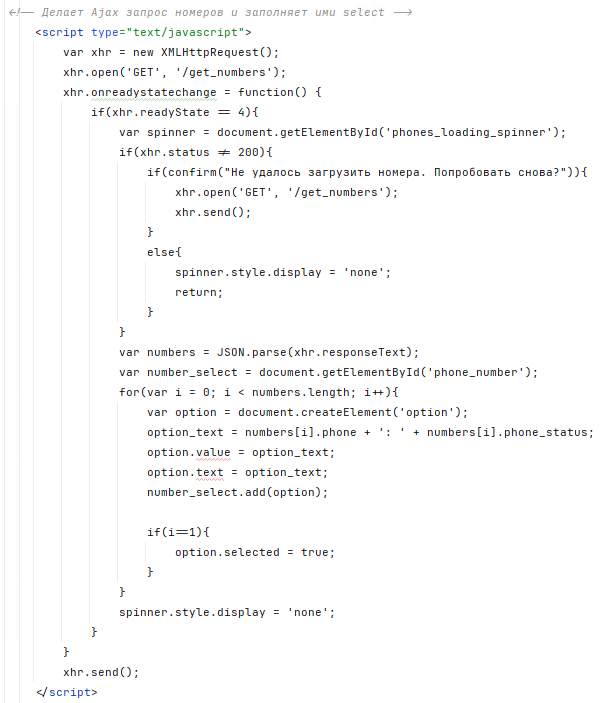


Рисунок 5 – Скрипт для получения исходящих номеров

Пока ожидается ответ от сервера, рядом с выпадающим списком вращается спиннер. В это время пользователь может начать заполнять другие поля. Как только ответ будет получен, выпадающий список будет заполен, и пользователь сможет выбрать исходящий номер.

Для удобства пользователя было добавлено отображение количества номеров в загруженном Excel-файле. Для работы с файлами .xlsx/.xls с помощью языка JavaScript была использована библиотека SheetJS. Код скрипта для подсчета количества номеров в файле представлен на рисунке 6.



Рисунок 6 – Скрипт для подсчета номеров в Excel-файле

В случае, если выбран 1 круг прозвона, поле для ввода расстояния между перезвонами в минутах не должно отображаться. Код скрипта, реализующего эту логику, представлен на рисунке 7.

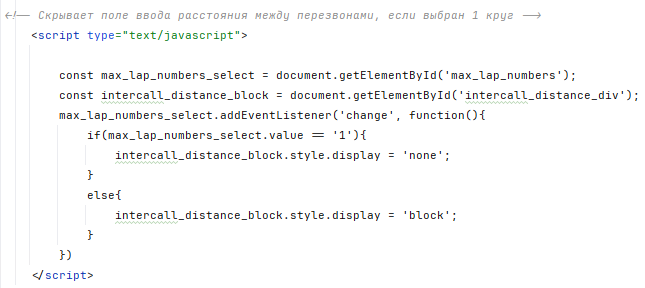


Рисунок 7 – Скрипт для скрытия поля ввода расстояния между перезвонами

Также, на страницу информационных прозвонов была добавлена кнопка «назад», нажав на которую пользователь может попасть обратно на страницу выбора типа прозвона. Итоговый вариант страница информационных прозвонов представлена на рисунке 8.

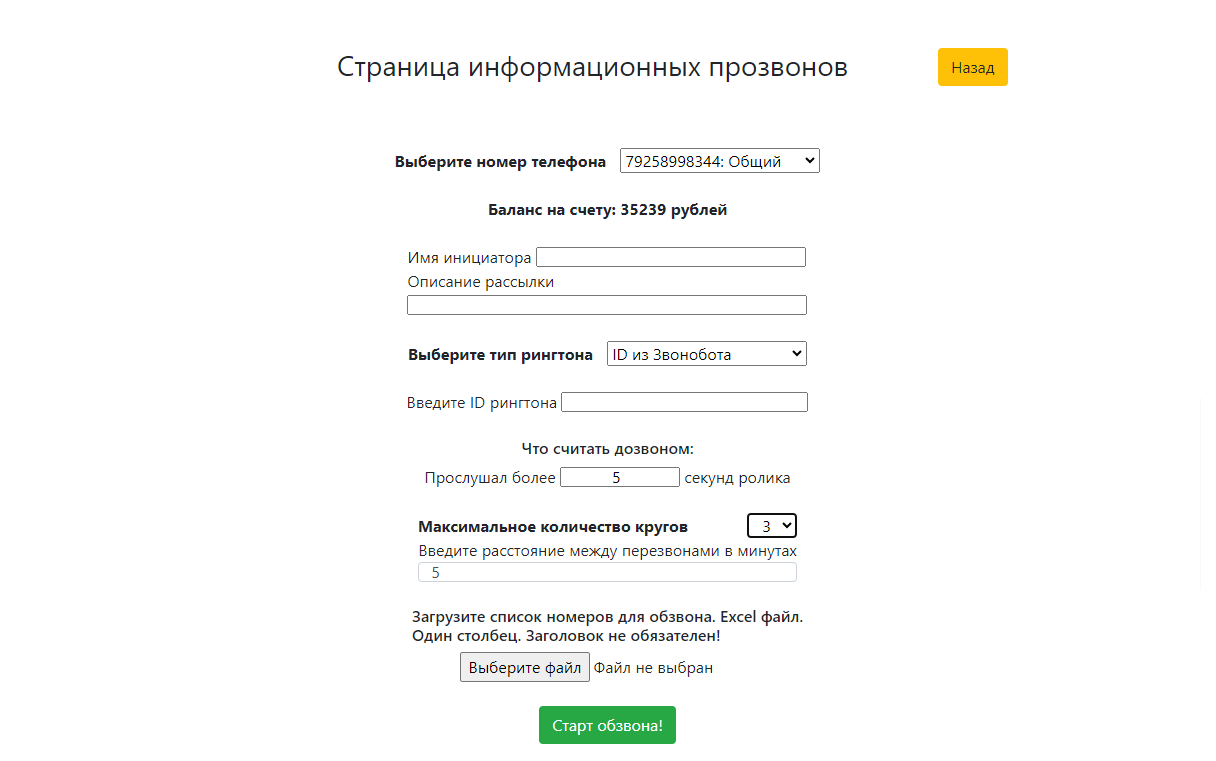


Рисунок 8 – Страница информационных прозвонов

4.1.3 Страница уведомления о незавершенном прозвоне

В интерфейс добавлена страница, оповещающая пользователя о том, что последний начатый им прозвон не был завершен (рисунок 9).

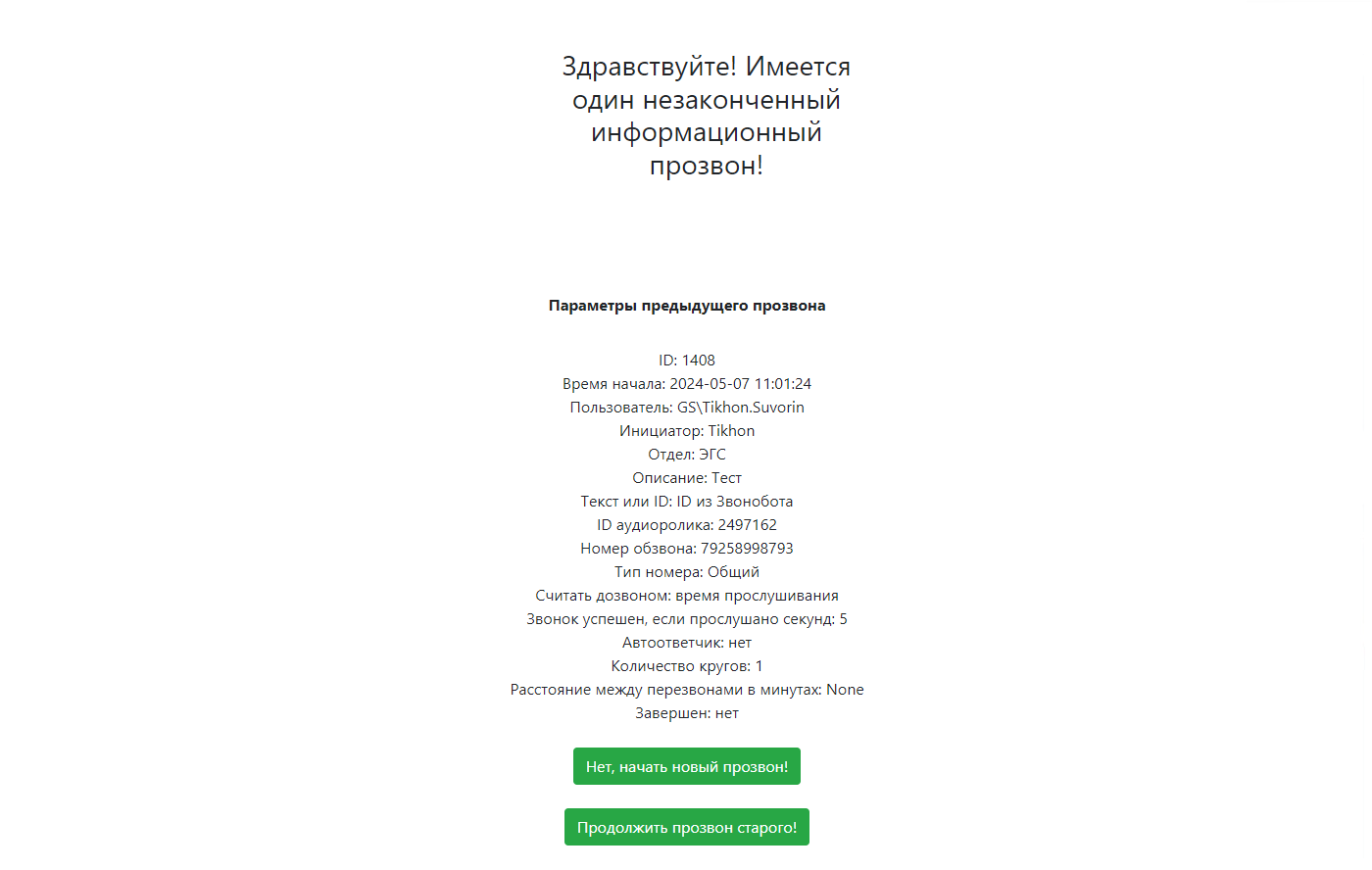


Рисунок 9 – Страница с информацией о незаконченном прозвоне

На данной странице отображаются параметры незавершенного прозвона, а также предлагается продолжить его, либо начать новый.

4.1.4 Страница вывода ошибок

В интерфейс добавлена страница, которая будет служить для отображения различного рода ошибок (рисунок 10). На ней содержится описание произошедшей ошибки, а также кнопка «назад», возвращающая пользователя на страницу выбора типа прозвона.

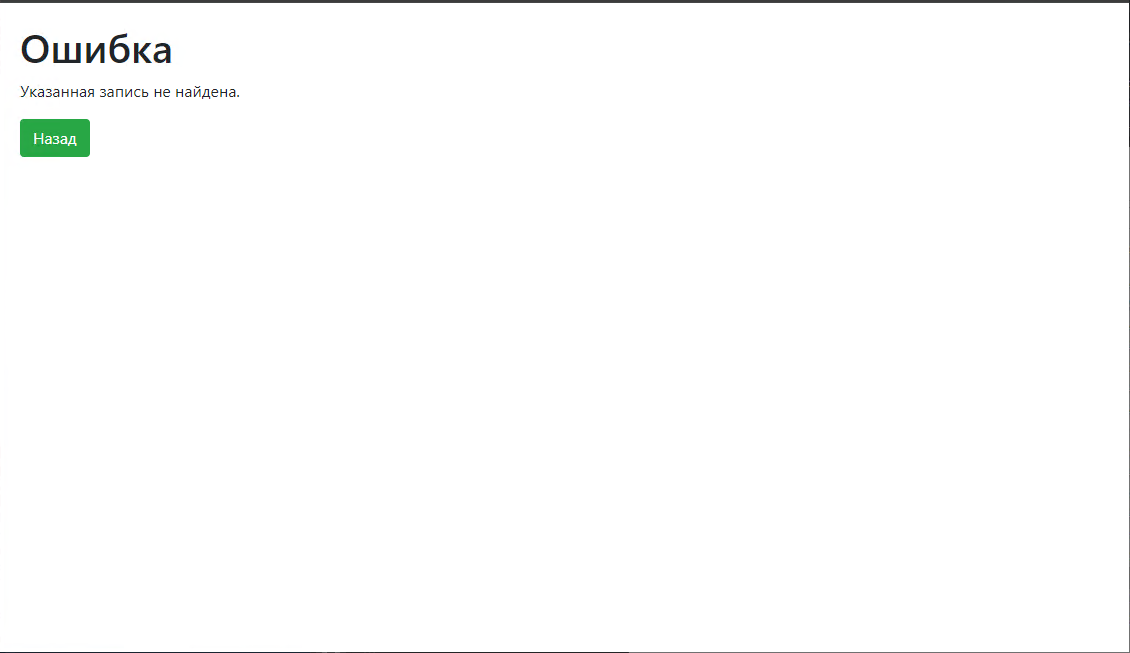


Рисунок 10 – Страница отображения ошибок

В частности, на данной странице отображается уведомление о некорректном ID аудиоролика.

4.1.5 Страница результатов прозвона

В интерфейс добавлена страница с информацией о результатах прозвона, которая открывается после его завершения (рисунок 11).

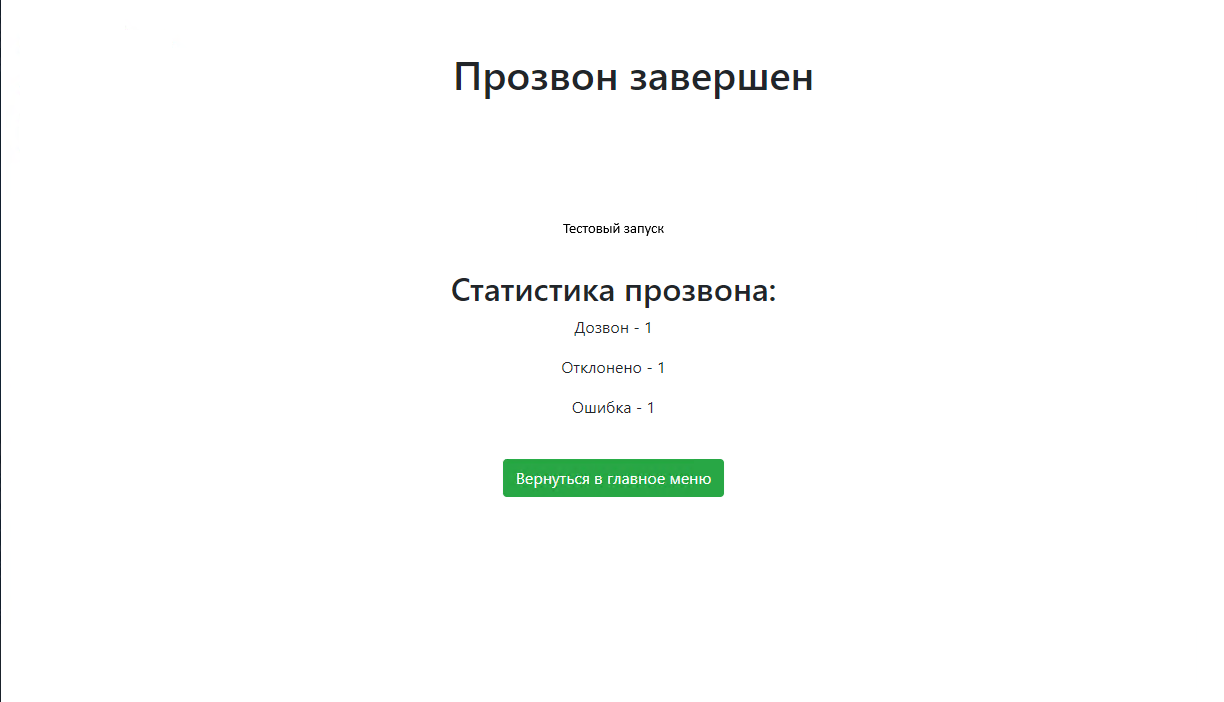


Рисунок 11 – Страница результатов прозвона

На данной странице отображается описание прозвона, а также его статистика: количество успешных, отклоненных и завершившихся с ошибкой звонков. Пользователь может вернуться на страницу выбора типа прозвона, нажав на соответствующую кнопку.

4.1.6 Страница диалоговых прозвонов

Так как на данный момент вне зависимости от выбора типа прозвона пользователь попадает на страницу информационных прозвонов, в интерфейс добавлена страница, предназначенная для создания диалоговых прозвонов (рисунок 12).

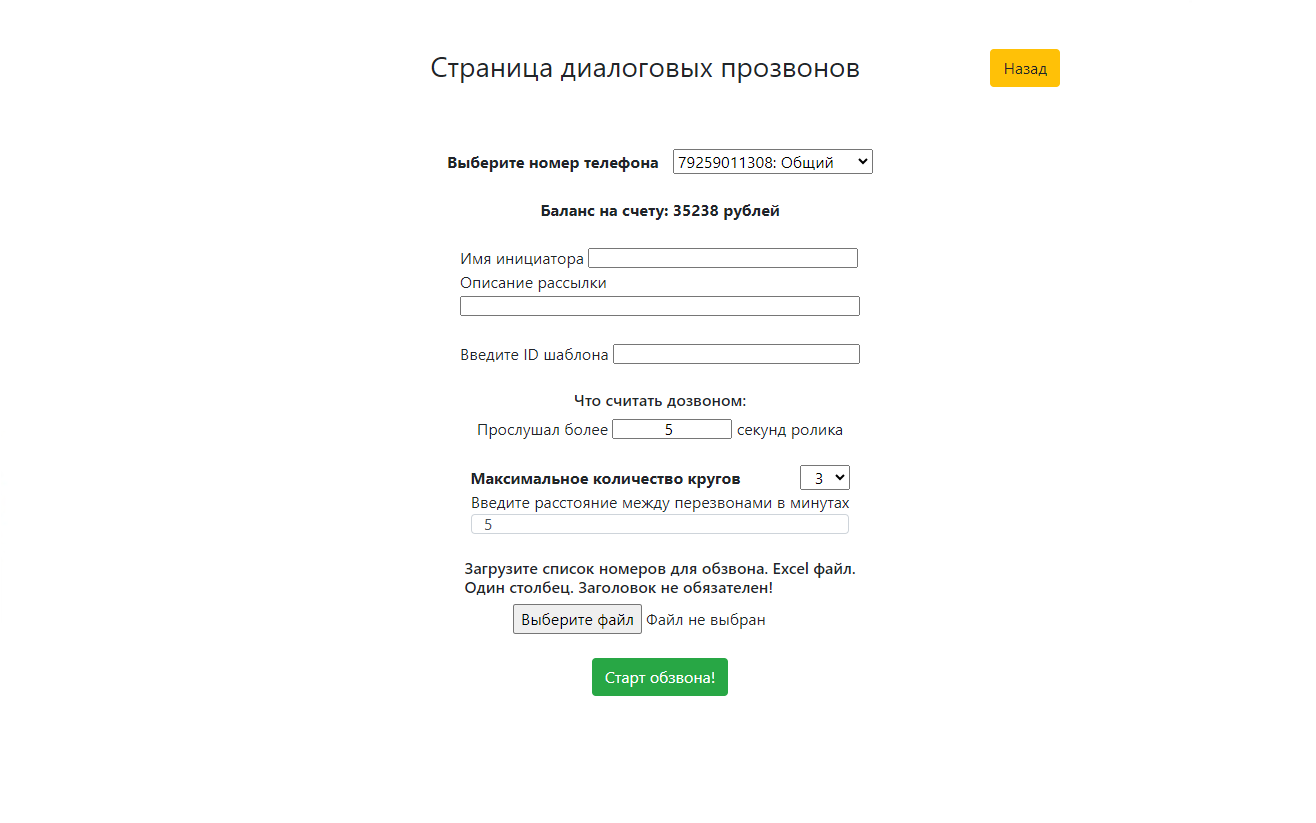


Рисунок 12 – Страница диалоговых прозвонов

Данная страница во многом использует тот же функционал, что и страница информационных прозвонов, но, вместо выбора ввода текста или ID аудиоролика, предлагается вести ID шаблона рассылки в сервисе «звонобот».

Wireframe-диаграмма интерфейса, получившегося в результате всех доработок, представлена в приложении Б.

4.2 Рефакторинг кода бэкенда

В ходе рефакторинга кода серверной части в первую очередь были исправлены ошибки, нарушающие стандарт PEP 8, убраны закомментированные строки кода, а параметры конфигурации flask-приложения из кода вынесены в конфигурационный файл (.yaml). Затем, были внесены изменения в структуру проекта. Итоговая структура представлена на рисунке 13.

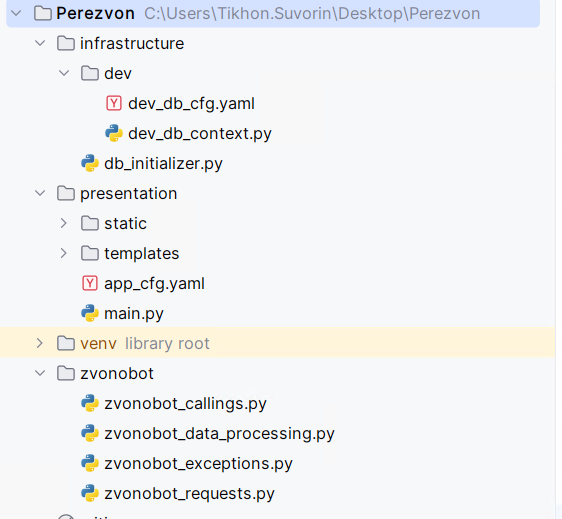


Рисунок 13 – Структура проекта после рефакторинга

Функционал по работе с базой данных был вынесен в уровень infrastructure. Модуль db\_initializer предназначен для получения объектов pyodbc connection и sqlachemy engine. В папке «dev» располагается модуль контекста, содержащий функции для работы с базой данных «Dev», а также конфигурационный файл с параметрами этой базы данных.

В папке zvonobot находятся файлы, относящиеся к уровню взаимодействия с внешним API сервиса «звонобот». Здесь содержаться модули, содержащие специфичные для работы с сервисом искючения, позволяющие осуществлять запросы к сервису, обрабатывать полученные данные. Модуль zvonobot\_callings.py содержит основную функцию launch\_calling(), позволяющую запустить рассылку, передав в неё словарь с параметрами и датафрейм с номерами телефонов.

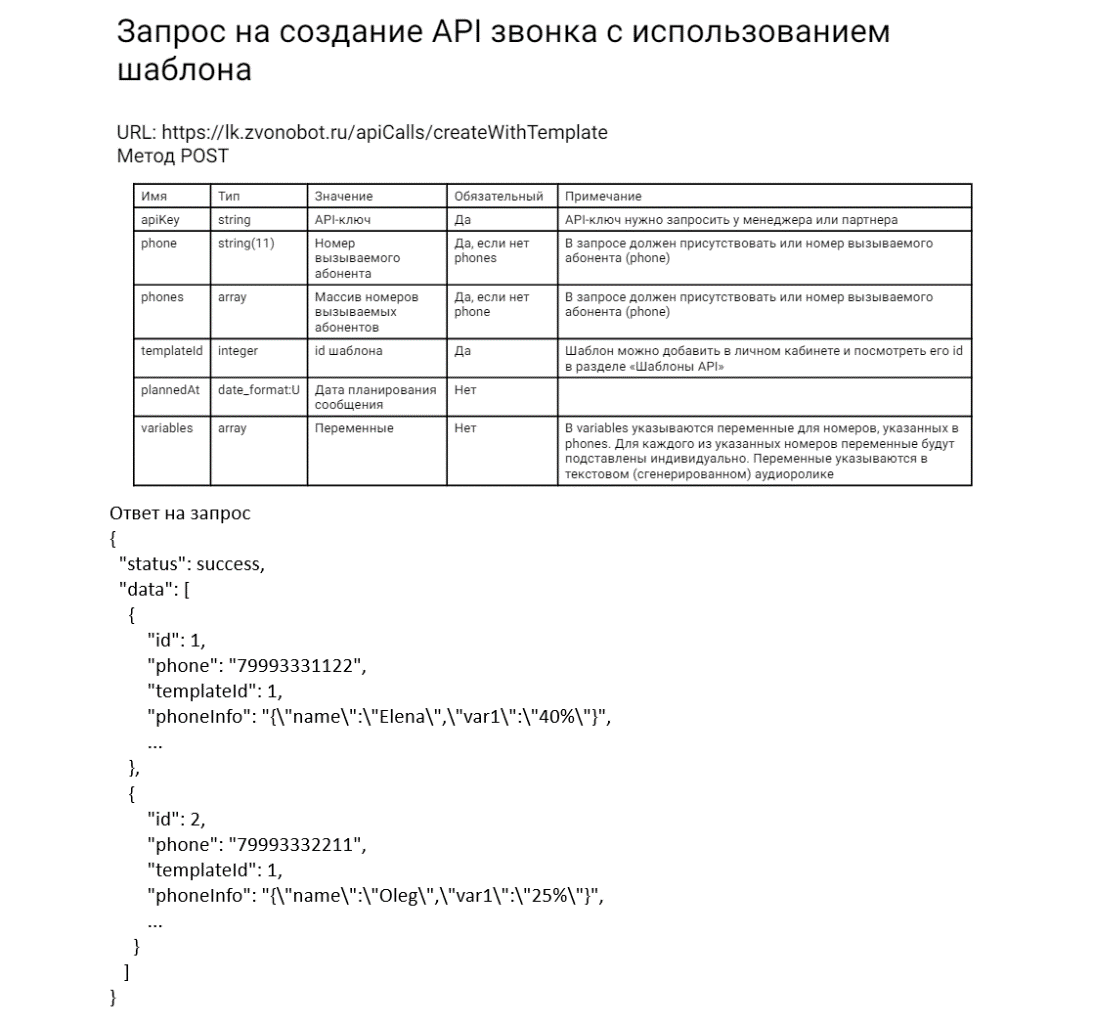
Уровень presentation – содержит само веб-приложение на flask. В папке templates хранятся html-страницы, в static – файлы стилей (.css) и скприптов (.js). В main.py содержаться контроллеры, принимающие и обрабатывающие запросы.

В результате рефакторинга код проекта стал более структурирован, что в будущем позволит удобнее вносить в него изменения и доработки.

4.3 Изучение части API-функционала сервиса «звонобот», отвечающей за диалоговые прозвоны

В ходе изучения официальной документации API сервиса «звонобот», с которой можно ознакомится в личном кабинете пользователя, было выяснено, что для управления диалоговыми прозвонами понадобится использовать 3 конечных точки.

1. https://lk.zvonobot.ru/apiCalls/createWithTemplate – используется для создания звонка с использованием шаблона (описание запроса на рисунке 14).

Рисунок 14 – Описание запроса на создание звонка с использованием шаблона

1. https://lk.zvonobot.ru/apiCalls/getWithTemplates – используется для получения информации о звонке (описание запроса на рисунке 15).

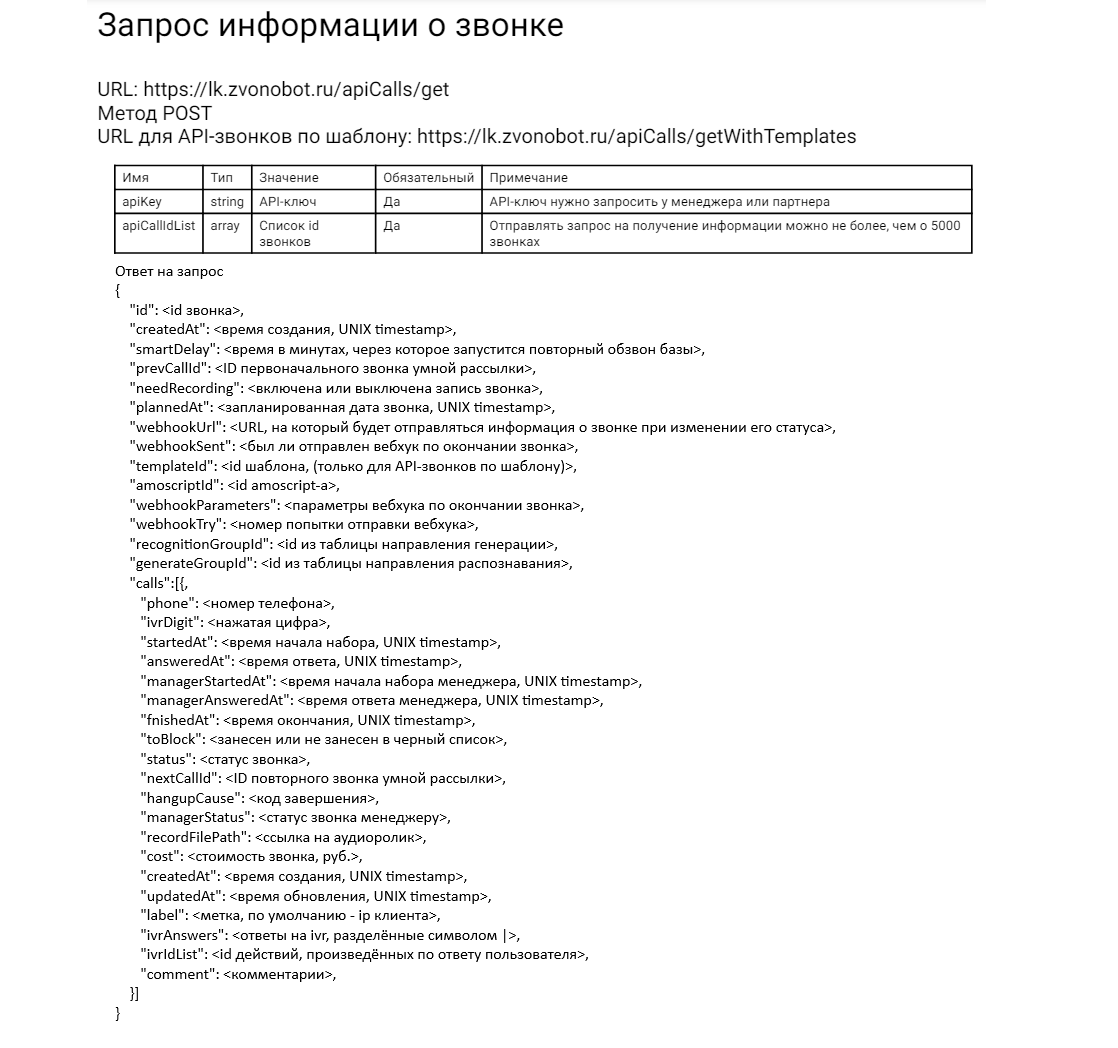


Рисунок 15 – Описание запроса на информацию о звонке с использованием шаблона

1. https://lk.zvonobot.ru/apiCalls/getTemplates – используется для получения списка шаблонов, созданных в личном кабинете (описание запроса на рисунке 16).

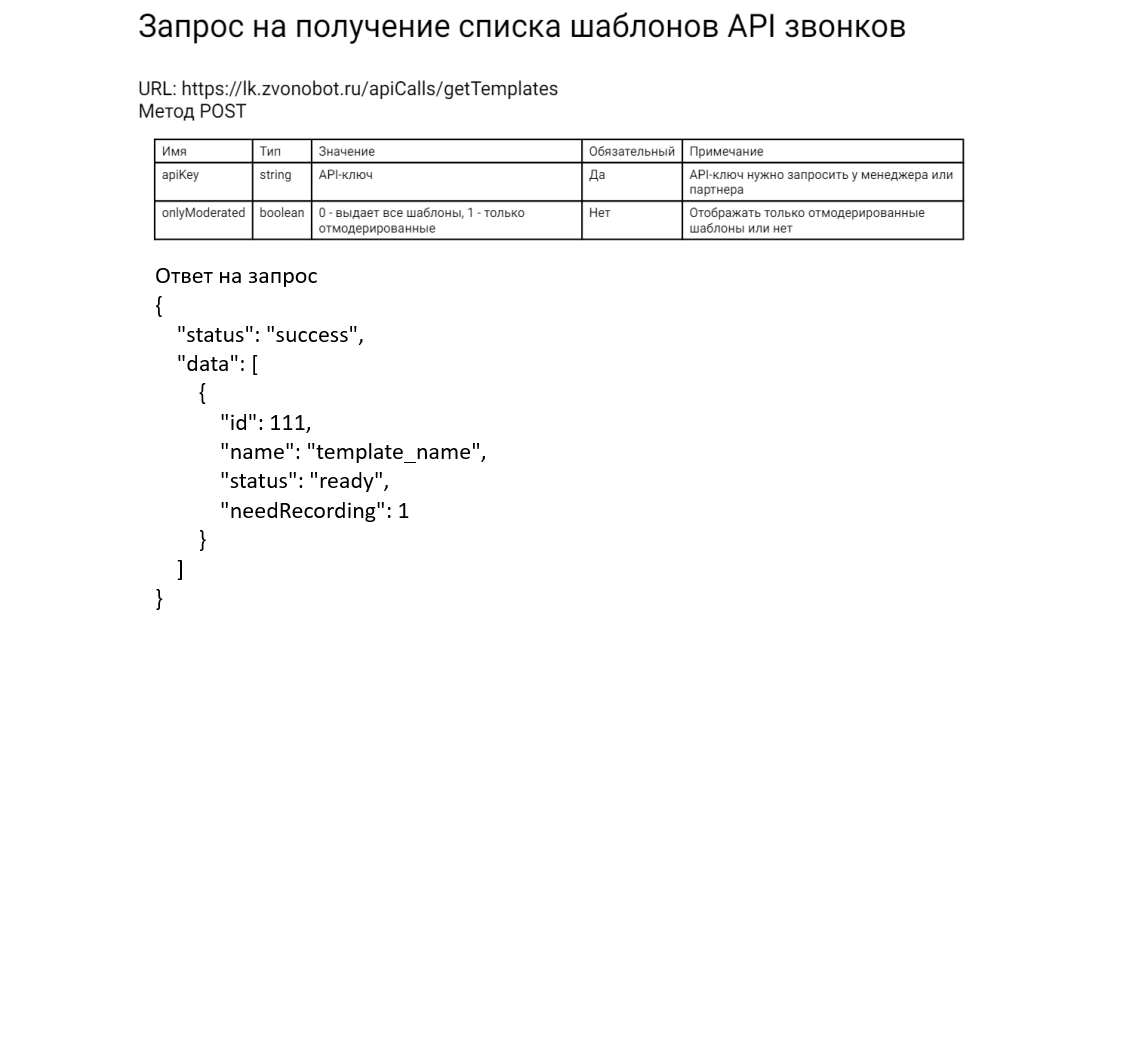


Рисунок 16 – Описание запроса на список шаблонов

Полученной информации о функционале API сервиса будет достаточно, чтобы внедрить возможность создания диалоговых прозвонов в разрабатываемый программный модуль.

4.4 Анализ решений в области синтеза речи

Синтез речи — это процесс преобразования текста в речь с использованием компьютерных программ. Существует несколько подходов к синтезу речи, таких как использование естественного языка (Natural Language Processing, NLP) и глубокого обучения (Deep Learning), или использование конкатенативного синтеза, который основан на записи отдельных фонем или звуков и их последующем объединении в слова и фразы.

На сегодняшний день существует множество сервисов в области синтеза речи, предлагающих различные функции и возможности. Например, сервис «звонобот» предлагает собственные возможности по синтезу речи. Однако, для этой цели было решено выбрать сервис Jay Copilot, из-за более низкой стоимости озвучки.

Jay Copilot – это умный помощник для бизнеса, сотрудников компаний и индивидуальных пользователей. В интерфейсе все сервисы представлены в виде удобных приложений, позволяющих решать конкретные бизнес-задачи. В основе приложений лежат как лучшие мировые нейросетевые модели, так и собственная разработка Just AI – большая языковая модель JustGPT. Интерфейс сервиса «диктор» в Jay Copilot представлен на рисунке 17, он понятен и прост в использовании. Для озвучки текста его достаточно напечатать и отправить в виде сообщения, при этом предоставляется возможность выбрать подходящий голос для озвучки из множества доступных.

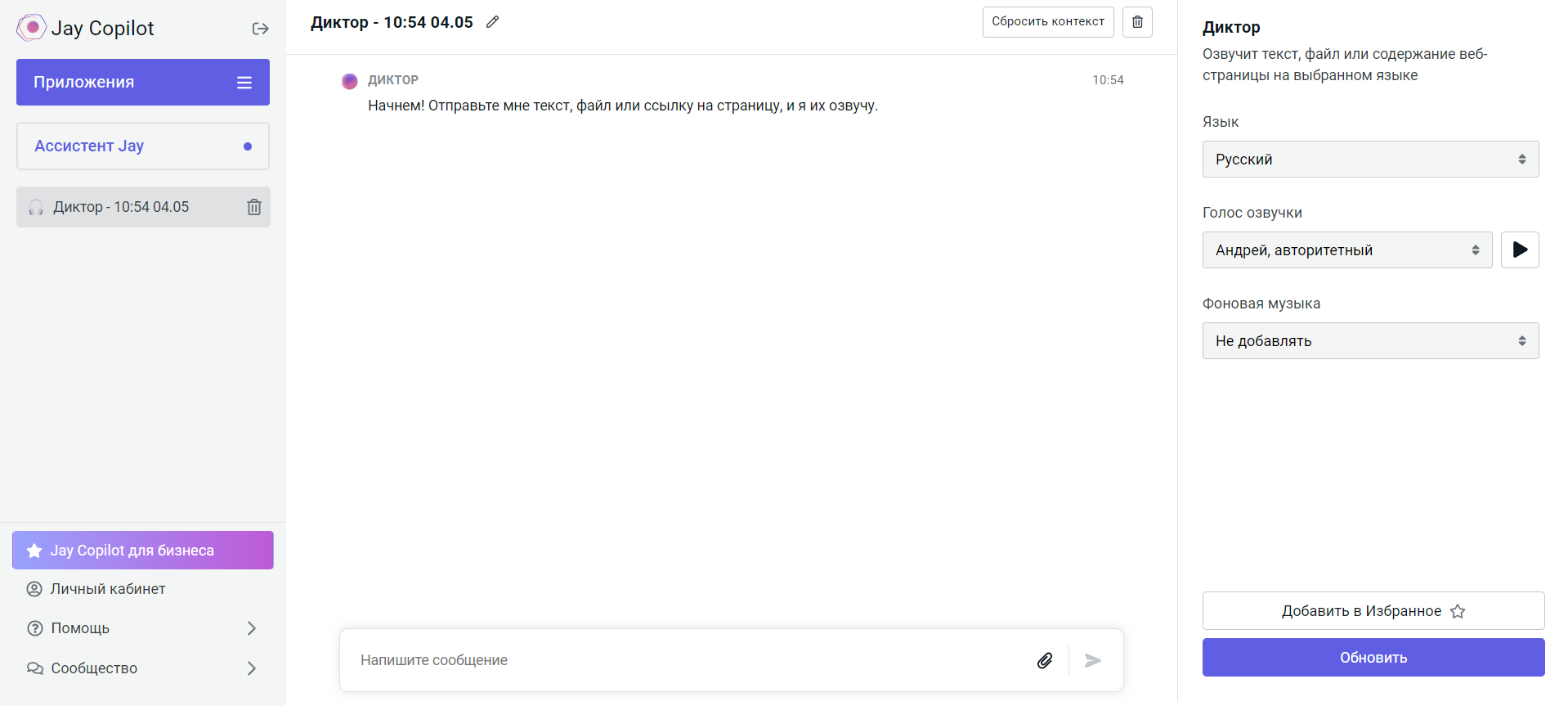


Рисунок 17 – Интерфейс сервиса «диктор» Jay Copilot

4.5 Интеграция диалоговых прозвонов в программный модуль

Согласно требованиям заказчика (управляющей компании «Эксплуатация Главстрой СПб»), в разрабатываемом сервисе необходимо реализовать 3 сценария диалоговых прозвонов, которые будут представлены в качестве отдельных модулей для последующего запуска на сервере по расписанию, с установленной дискретностью:

* Сценарий изучения клиентской лояльности (NPS): необходим для изучения отношения клиентов к компании.
* Сценарий изучения клиентской удовлетворенности (CSAT): предназначен для изучения удовлетворенности клиентов, обращавшихся ранее в диспетчерскую службу компании.
* Сценарий изучения удовлетворенности жильцов администратором дома (ADMIN): служит для оценки качества работы администраторов домов.

Реализация данных сценариев в сервисе будет проводится в 3 этапа: в первую очередь необходимо создать шаблоны рассылок в сервисе «звонобот», затем реализовать структуру хранения новых данных, получаемых по результатам выполнения рассылок, и разработать сами скрипты для запуска рассылок, интегрировав их в разрабатываемый сервис.

4.5.1 Создание шаблонов рассылок в сервисе «звонобот»

Первый – сценарий изучения клиентской лояльности (NPS – Net Promoter Score). Для этого сценария был создан шаблон рассылки в сервисе «Звонобот» (рисунок 18). После того, как клиент отвечает на звонок, ему проигрывается основной аудиоролик: «Добрый день. Оцените, пожалуйста, работу управляющей компании Эксплуатация Главстрой Санкт-Петербург по шкале от 0 до 10. Насколько вероятно, что Вы порекомендуете нашу компанию друзьям и коллегам, где 10 – это максимум, обязательно порекомендую. Благодарим за ответ.». Клиент ставит оценку голосом, после чего проигрывается завершающий аудиоролик «благодарим за ответ».

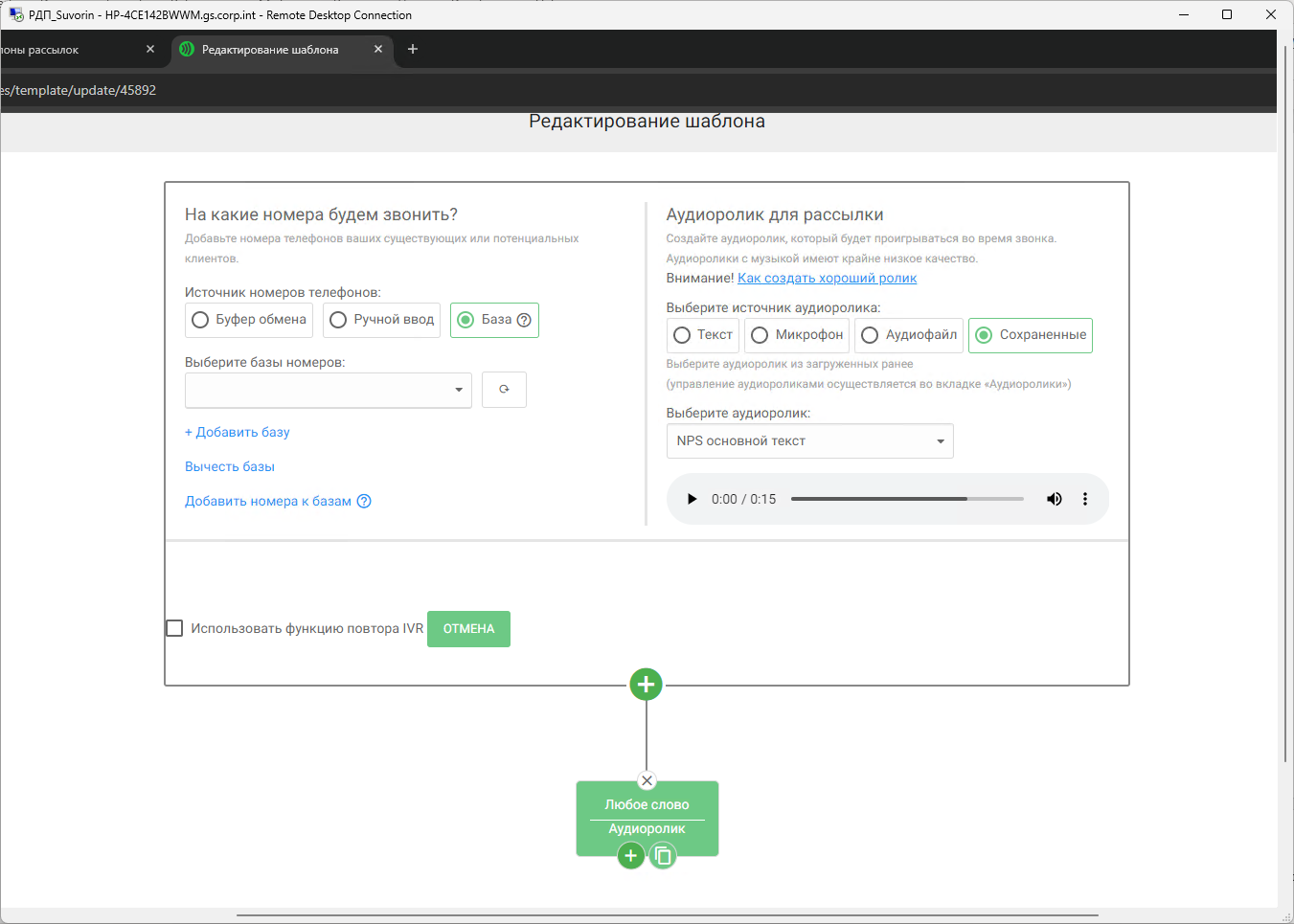


Рисунок 18 – Шаблон сценария NPS

Логика двух следующих сценариев (изучения клиентской удовлетворенности и удовлетворенности администратором дома), шаблон которых представлен на рисунке 19, в целом схожа: после ответа на звонок клиент в первую очередь слышит вопрос. В случае CSAT это: «Добрый день, Вы обращались в диспетчерскую службу управляющей компании Эксплуатация Главстрой Санкт-Петербург. Оцените, пожалуйста, качество работы диспетчера по шкале от 0 до 10, где 10 – это максимальная оценка его работы», а в случае ADMIN – «Добрый день. Оцените, пожалуйста, качество работы администратора вашего дома по шкале от 0 до 10, где 10 – это максимальная оценка его работы».

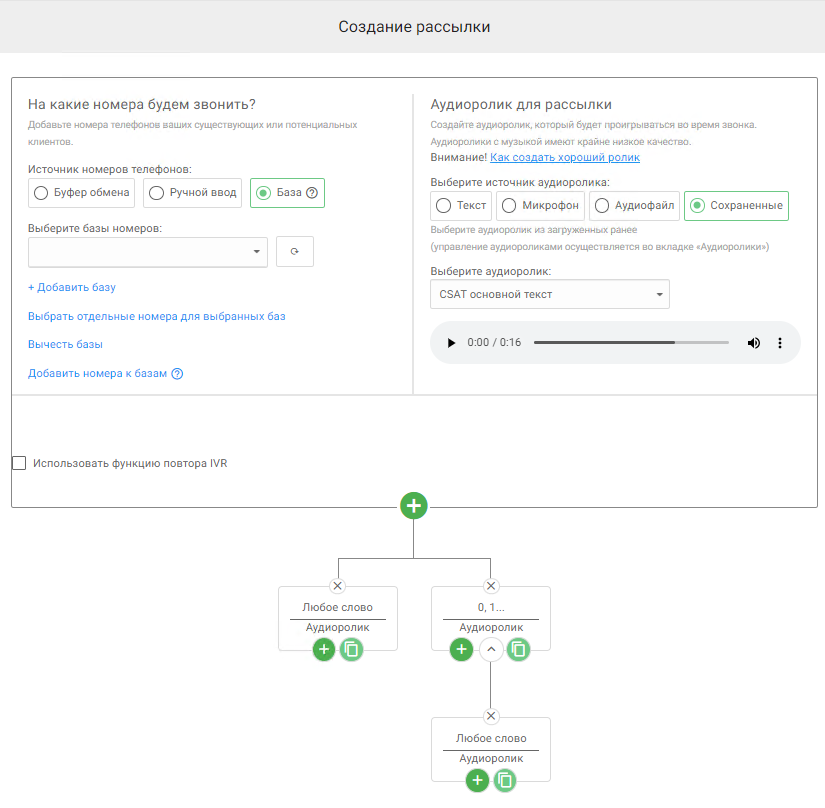


Рисунок 19 – Шаблон сценариев CSAT и ADMIN

После того, как клиент ставит оценку голосом, в зависимости от ней проигрывается либо аудиоролик «благодарим за оценку», в случае если оценка от 5 до 10, либо задаётся открытый вопрос «что повлияло на вашу оценку», если оценка от 0 до 4. В этом случае пользователю даётся 10 секунд для изложения своего мнения, после чего проигрывается аудиоролик «благодарим за ответ». Так как в данных сценариях ожидается получение развернутой обратной связи клиентов, включена функция сохранения записей разговоров для последующего их анализа.

4.5.2 Проектирование и реализация структуры хранения новых данных

Для хранения новых данных в базу данных было добавлено несколько новых таблиц (рисунок 20). Для результатов опроса по каждому из сценариев созданы таблицы Zvonobot\_NPS, Zvonobot\_CSAT и Zvonobot\_ADMIN. В каждой из них присутствуют атрибуты GlobalCallingID (глобальный индекс рассылки), PhoneNumber (телефон клиента), Score (выставленная оценка) и InsDate (время вставки). При этом первые три атрибута объединены в составной первичный ключ.

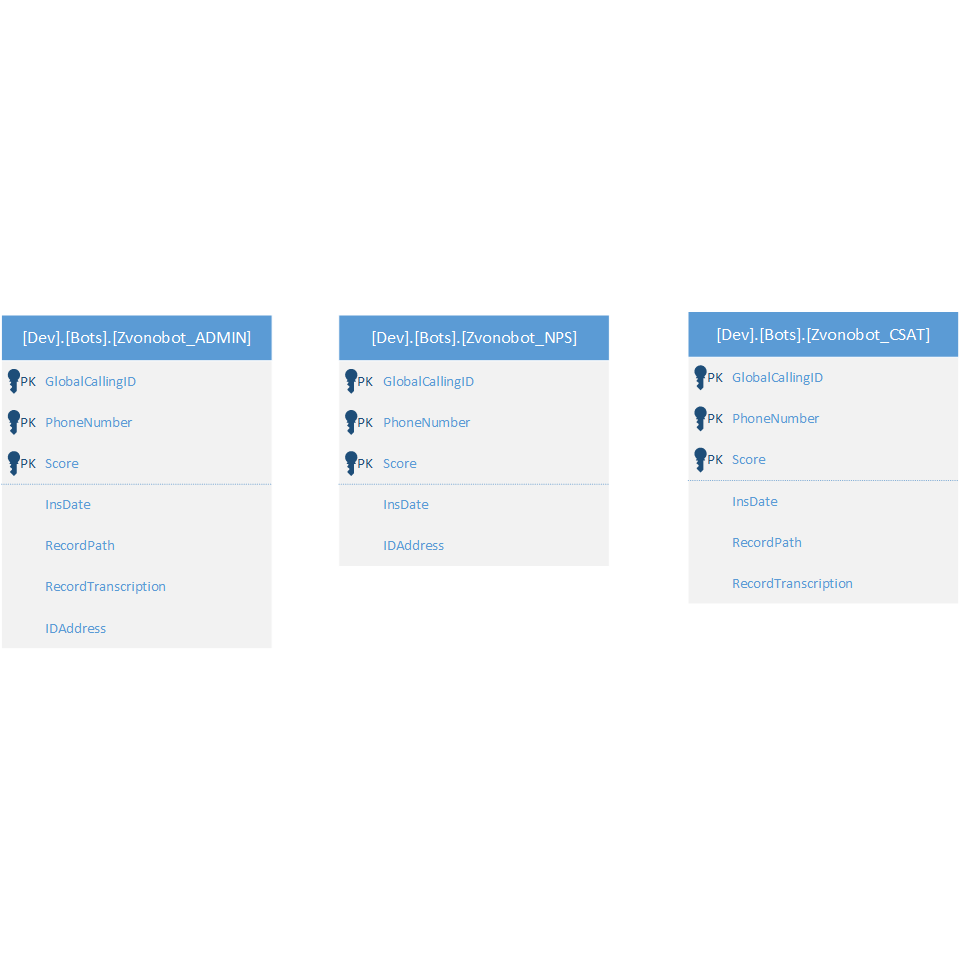


Рисунок 20 – Структура добавленных таблиц

Так как в сценариях ADMIN и CSAT ожидается получение развернутой обратной связи от клиентов, в соответствующие таблицы добавлены также атрибуты RecordTranscription (транскрибация ответа на открытый вопрос), и RecordPath (путь, по которому сохранена запись разговора).

Помимо этого, в сценариях ADMIN и NPS требуется знать, с какого адреса поступила оценка, поэтому в соответствующие таблицы также добавлен столбец IDAddress.

4.5.3 Реализация сценариев в проекте

Для реализации сценариев в проекте в первую очередь в модуль zvonobot\_requests были интегрированы функции по взаимодействию с ранее изученными конечными точками API «звонобот». Это функции для создания рассылки с шаблоном (run\_calling\_with\_template()) и получения информации о состоянии рассылки с шаблоном (check\_calling\_state\_with\_template()).

Далее, в структуру проекта добавлена папка scripts, в которой заранее подготовлены папка common (с модулями для общего использования всеми сценариями) и папки для трех разрабатываемых сценариев. Структура директории scripts представлена на рисунке 21.

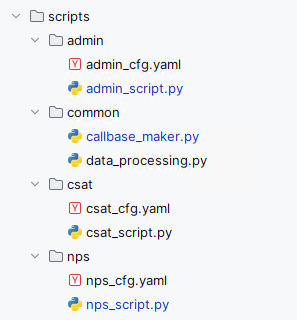


Рисунок 21 – Структура директории scripts

В каждой из папок сценариев были созданы конфигурационные файлы (.yaml), которые позволяют настраивать параметры прозвона. На рисунке 22 представлена конфигурация сценария CSAT. Конфигурации остальных сценариев схожи, за исключением параметров answer\_start\_time и record\_dir, которые отсутствуют в конфигурации NPS.

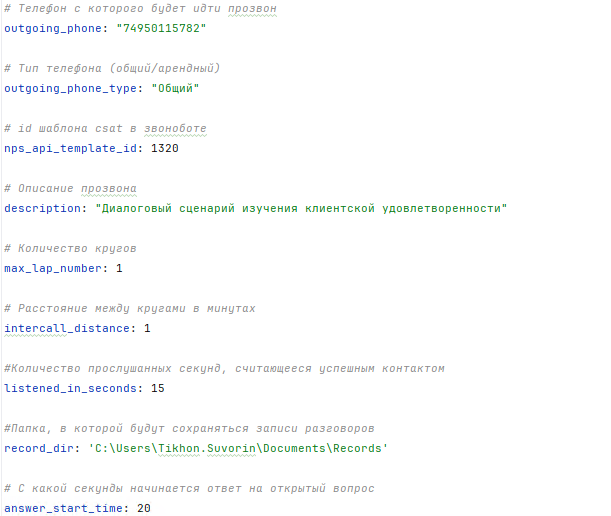


Рисунок 22 – Конфигурация сценария CSAT

В папку common были добавлены модули для получения базы номеров для прозвона (callbase\_maker) и обработки данных (callbase\_maker).

Завершающим этапом была разработка самих сценариев. Они работают по схожей логике: получают базу номеров для прозвона, затем формируют датафреймы с параметрами рассылки и номерами телефонов, после чего запускают прозвон. Когда тот завершается, оценки клиентов собираются в датафрейм и вносятся в соответствующую таблицу. Сценарии CSAT и ADMIN также записывают в таблицы транскрибации развернутых ответов и сохраняют аудиозаписи звонков по указанному пути.

**Тема 5. Разработка эксплуатационной документации**

Текст

**Тема 6. Проведение испытаний, отладка и внедрение программного продукта на предприятии**

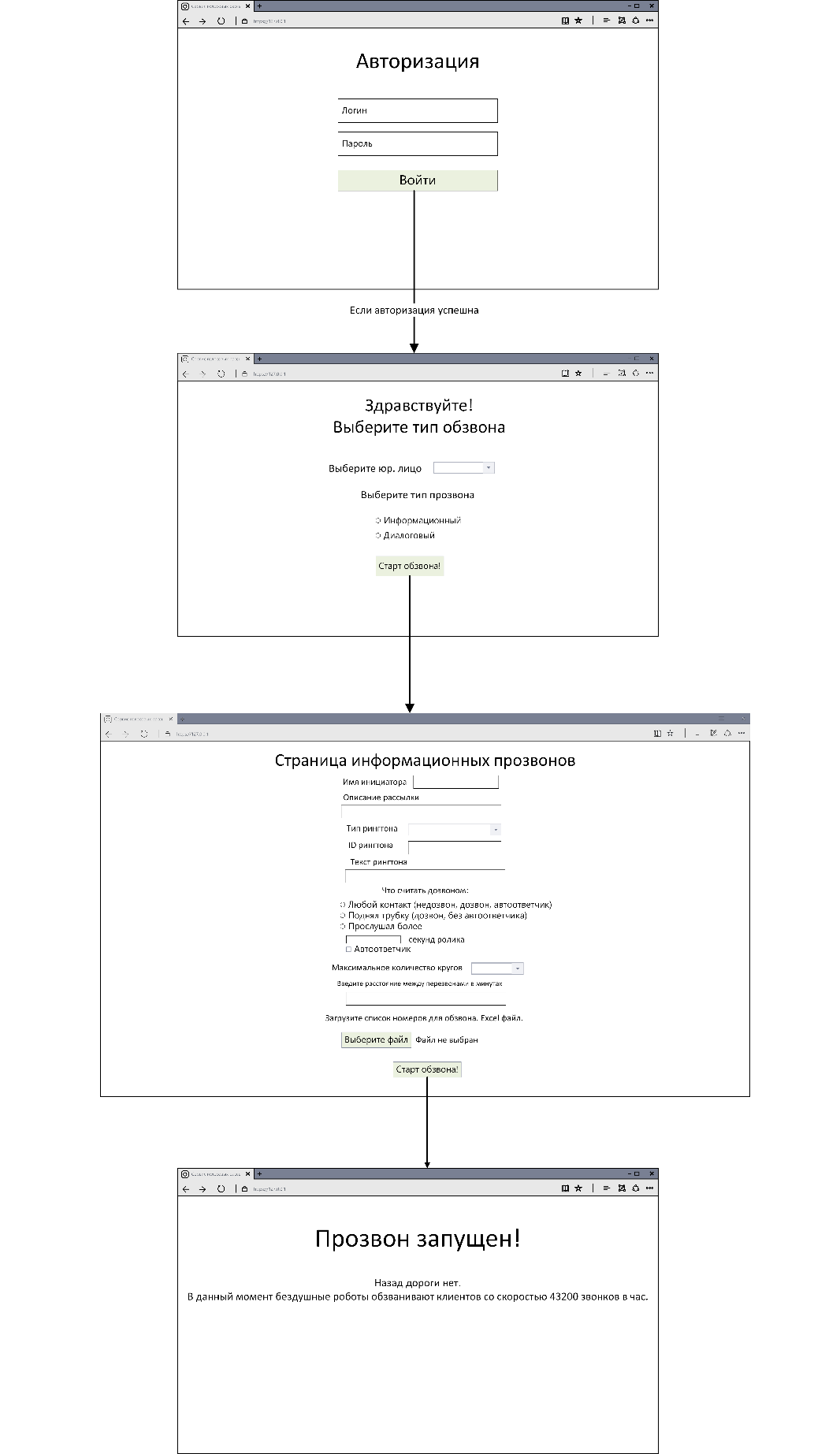
Текст

**СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ**

1. Чистый код: создание, анализ и рефакторинг, / Роберт Мартин – 2020 г. - 150 с.
2. Приёмы объектно-ориентированного проектирования, / Эрих Гамма, Ральф Джонсон – 2019 г. -213с.
3. Грокаем алгоритмы / Адитья Бхаргава – 2019 г. - 378с.
4. Изучаем SQL / Алан Болье – 2020 г. - 145 с.
5. SQL. Сборник рецептов / Энтони Молинаро – 2019 г. - 145 с.
6. Совершенный алгоритм. Основы / Тим Рафгарден – 2020 г. - 240 с.
7. Автоматное программирование / Надежда Поликарпова – 2019 г. - 140 с.
8. SQL: Руководство по изучению языка / Крис Фиайли – 2020 г. - 190 с.
9. Чистая архитектура / Роберт Мартин – 2020 г. - 290 с.
10. Разработка интеллектуальных систем / Джефф Хултен – 2021 г. - 303 с.
11. Структура компьютерных программ / Сергей Васильев – 2019 г. - 120с.
12. Искусство гибкой разработки ПО / Джефф Паттон – 2019 г. - 210 с.
13. Алгоритмы. Построение и анализ / Томас Кармен – 2021 г. - 291с.
14. Git для профессионального программиста / Бен Штрауб – 2022 г. - 156 с.
15. Изучаем Python, 5-е издание / Лутц Марк – 2019 г. – 1280 с.
16. Программирование на языке высокого уровня Python: учеб. пособие для прикладного бакалавриата / Д. Ю. Федоров. — 2019 г. — 161 с.

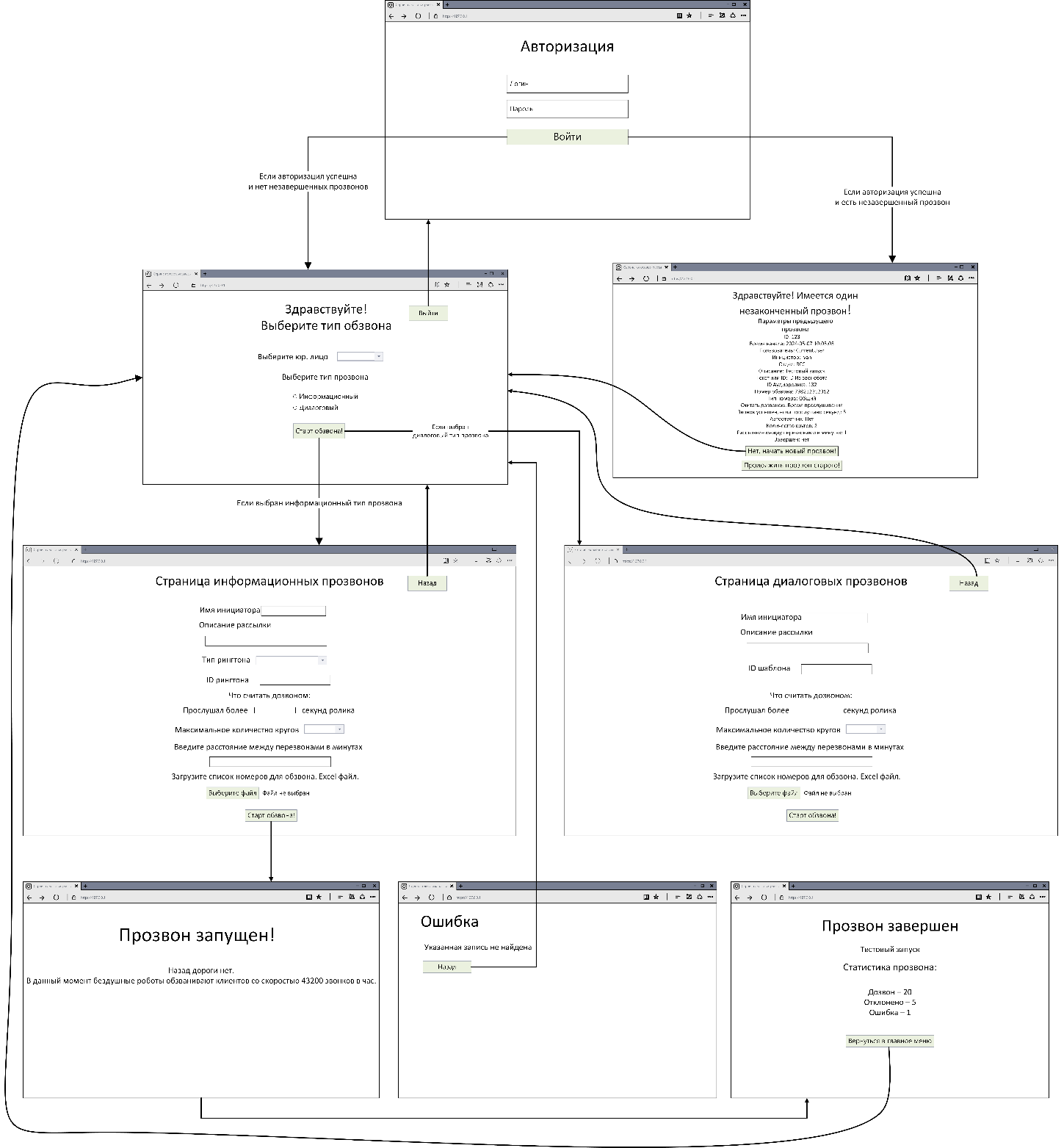
**ПРИЛОЖЕНИЕ А**

**Wireframe-диаграмма веб-интерфейса до доработок**

****

**ПРИЛОЖЕНИЕ Б**

**Wireframe-диаграмма интерфейса после доработок**

****