

Министерство образования и науки Российской Федерации
ФГБОУ ВО РОССИЙСКИЙ ГОСУДАРСТВЕННЫЙ
ГИДРОМЕТЕОРОЛОГИЧЕСКИЙ УНИВЕРСИТЕТ
(РГГМУ)

Институт Информационных систем и геотехнологий
КАФЕДРА ПРИКЛАДНОЙ ИНФОРМАТИКИ

БАКАЛАВРСКАЯ РАБОТА

На тему Проектирование и разработка информационной системы поддержки
режима дистанционного голосования

Исполнитель Воробьев Денис Игоревич
(фамилия, имя, отчество)

Руководитель доктор технических наук, профессор
(ученая степень, ученое звание)
Истомин Евгений Петрович
(фамилия, имя, отчество)

«К защите допускаю»

Заведующий кафедрой _____
(подпись)

доктор технических наук, профессор
(ученая степень, ученое звание)
Истомин Евгений Петрович
(фамилия, имя, отчество)
«__» _____ 20__ г.

Санкт-Петербург
2022

Оглавление

ВВЕДЕНИЕ	3
1. АНАЛИЗ МЕТОДОВ И СРЕДСТВ ОНЛАЙН ГОЛОСОВАНИЙ	6
1.1 Проблемы очных голосований	6
1.2 Сравнение с программными аналогами.....	7
1.3 Анализ объекта исследования	12
1.4. Сроки реализации проекта	16
2. ПРОЕКТИРОВАНИЕ ИНФОРМАЦИОННОЙ СИСТЕМЫ	18
2.1 UML моделирование.....	18
2.2 Моделирование бизнес-процессов в нотации BPMN.....	23
2.3 Схема базы данных	26
2.4. Схема восстановления и сохранения информации.	27
3. РАЗРАБОТКА ИНФОРМАЦИОННОЙ СИСТЕМЫ	29
3.1. Серверная сборка	29
3.2. Взаимодействие с интерфейсом информационной системы.....	37
3.3. Надёжность системы поддержки дистанционного голосования	46
ЗАКЛЮЧЕНИЕ	51
СПИСОК ЛИТЕРАТУРЫ.....	52
ПРИЛОЖЕНИЕ 1	57

ВВЕДЕНИЕ

Еще год назад невозможно было представить себе, как надвигающаяся пандемия COVID-19 перевернет всю нашу жизнь, поставив под сомнение прошлые предположения и пошатнет уверенность в будущем. COVID-19 обладает одновременно тремя характеристиками, которые позволили ему нанести исторический удар по человечеству, запустив глобальный виртуальный «локдаун», как единственное оружие против его неконтролируемого распространения. COVID-19 сочетает в себе характеристики вируса, который, насколько нам известно, никогда раньше не заражал людей на постоянной основе, вместе с его чрезвычайной эффективностью в отношении передачи от человека к человеку и относительно высоким уровнем заболеваемости и смертности, особенно среди пожилых людей и лиц с сопутствующими заболеваниями. Это самый настоящий «идеальный шторм» надвигающейся инфекционной болезни.^[1]

В последствии этого события спрос на удаленную работу в России начал расти еще в феврале, а число таких вакансий в марте увеличилось на 71%. При этом соискатели в период пандемии коронавируса также стали чаще искать работу рядом с домом, а список доступных для дистанционной работы профессий значительно расширился.^[2]

Последствием случившегося стало то, что в настоящее время многие заседания, советы и обсуждения перешли в дистанционный формат.

В результате того, что большинство советов, заседаний, конференций перешли на удаленный, т.е. дистанционный формат проведения мероприятий возможностей и ресурсов для рассмотрения, или обсуждений касаемых вопросов до недавнего времени имела значительную потребность, но на данном этапе она имеет необходимость.

Также для сохранения приватности личных данных возникла потребность проводить голосование анонимно.

Объектом исследования^[3] является информационная система для проведения анонимного голосования без сохранения персональных данных голосующих.

Предметом исследования^[4] является проблема проведения голосования, обсуждений и дискуссий в удалённом формате с гарантией защиты персональных данных.

Целью работы является разработка информационной системы с возможностью создания анонимного сессионного голосования.

Для достижения цели необходимо решить следующие задачи:

- Произвести теоретический и практический анализ на тему платформ для создания опросов и голосования;
- Спроектировать систему;
- Разработать функционирующее приложение на выбранной платформе.

Данная информационная система позволяет:

- Создавать голосование с индивидуальным кодом доступа;
- Отслеживать результаты в процессе голосования;
- Отслеживать количество голосующих;
- Печатать отчёты по итогам голосования;
- Печатать анонимные бюллетени участников;
- Голосовать в сессии идущего голосования.

Методы, стандарты и технологии, примененные в ВКР:

- Сравнительный анализ имеющихся на рынке аналогичных платформ для создания опросов и голосования;
- Анализ и моделирование бизнес-процессов в нотациях BPMN;
- Подходы планирования разработки систем SWOT, VCM, BPR и ISA;
- Технологии баз данных и управления базами данных;
- Моделирование системы с помощью UML диаграмм.

Используемый инструментарий:

В ходе выполнения практической части ВКР, был использован следующий стек технологий:

- Java^[5] – строго типизированный язык программирования, выполняемый на виртуальной машине;
- Spring Boot framework^[6] – фреймворк для создания веб-приложений;
- HTML/CSS^{[7][8]} – средства для создания веб-страниц;
- Thymeleaf^[9] – движок шаблонов HTML/CSS/JavaScript .

1. АНАЛИЗ МЕТОДОВ И СРЕДСТВ ОНЛАЙН ГОЛОСОВАНИЙ

1.1 Проблемы очных голосований

Возможность проведения, участия и подведения результатов голосования удалённо и ранее была востребована, однако после пандемии многие собрания и обсуждения не имели возможности проводится очно. Это вынудило многие компании, учреждения и обычные мероприятия сменить очный формат на дистанционный. Во время самоизоляции популярность удалённых сервисов организации различного рода мероприятий возросло как никогда ранее. Со временем ограничения на проведение очных мероприятий будут спадать до тех пор, пока не оставят от себя и следа.

Тем не менее факт того, что человечество рано или поздно перейдёт на дистанционные формы проведения различного рода мероприятий всегда был очевиден и являлся лишь вопросом времени. Пандемия, а точнее режим самоизоляции, на который большинство было вынуждено перейти, побудил людей столкнуться с удобствами, которые дистанционный формат за собой влечёт. Это событие сильно поспособствовало возрастанию востребованности сервисов организации такого рода мероприятий, которые ранее производились преимущественно очно.

Также всегда существовала проблема сокрытия приватной информации при голосовании. Очень часто в ходе проведения голосований особо остро стояла проблема сохранения конфиденциальности процесса обсуждения вопроса. Многие участники не желали участвовать в таких мероприятиях из-за возможности обнародования их вклада в тот или иной результат голосования. История знает большое число случаев, при которых происходила утечка информации приватных данных голосующих и в целом процесса голосования.

Таким образом, проблема возможности проведения именно такого голосования, которое являлось безопасным, т.е. без риска раскрытия приватной информации голосующих, всегда была актуальна.

Решением всех этих проблем является не ужесточение правил проведения очных дискуссий, а перевод голосования в дистанционный формат без возможности записи и хранения личных данных голосующих.

1.2 Сравнение с программными аналогами

Произведение сравнения с другими подобными информационными системами необходимо, так как оно представляет собой метод сравнительного анализа, т.е. эмпирический метод исследования, который позволяет познать особенности рассматриваемых процессов, объектов, методологий с помощью логики в изучаемой сфере.

Таблица 1.1. Анализ функций сервисов

Название ИС	Функции ИС	Формат использования	Подведение итогов голосования
Google Forms	<ul style="list-style-type: none"> — Создание опросов. — Возможность настраивать варианты ответа. — Просмотр сводки в процессе и по результатам голосования. — Возможность сохранения ответов в формате .csv и печати. 	<ol style="list-style-type: none"> 1. Администратор создаёт форму с вопросом и передаёт ссылку на голосование участником, либо вставляет голосование в сайт в форме html-тега. 2. Участники переходят по ссылке и голосуют 	<ul style="list-style-type: none"> — Администратор устанавливает переключатель «Принимать ответы» в состоянии «ВЫКЛ». — Администратор может видеть итоги голосования в форме диаграммы. — Администратор может сохранить результаты и распечатать.

<p>Polys – система онлайн голосований</p>	<ul style="list-style-type: none"> — Создание опросов. — Возможность настраивать варианты ответа. — Просмотр сводки в процессе и по результатам голосования. — Возможность выбора фона голосования. — Печать результатов по окончании голосования. 	<ol style="list-style-type: none"> 1. Администратор создаёт форму с вопросом, определяет сроки и даёт доступ либо через email, либо ссылку на голосование, либо передаёт уникальный код. 2. Участники переходят на голосование и выбирают варианты ответов. 	<ul style="list-style-type: none"> — По окончании срока голосования оно закрывается. — Администратор может видеть итоги голосования в форме диаграммы. — Администратор может распечатать результаты.
<p>Pollservice - онлайн- сервис для проведения опросов и голосований в интернете</p>	<ul style="list-style-type: none"> — Создание опросов. — Возможность настраивать варианты ответа. — Просмотр сводки в процессе и по результатам 	<ol style="list-style-type: none"> 1. Администратор создаёт опрос и получает на него ссылку. 2. Голосующий переходит по ссылке и выбирает вариант ответа. 	<ul style="list-style-type: none"> — Администратор может видеть итоги голосования в форме диаграммы.

	ГОЛОСОВАНИЕ.		
--	--------------	--	--

По результатам исследования функций программных аналогов можно выделить необходимый функционал.

Таблица 1.2 - Анализ аналогичных систем

Функционал	Google Forms	Polys – система онлайн голосований	Pollservice - онлайн-сервис для проведения опросов и голосований в интернете	ИС дистанционное голосование
Создание голосования	+	+	+	+
Возможность настраивать варианты ответа	+	+	+	+
Просмотр сводки в процессе голосования	+	+	-	+
Просмотр сводки по результатам голосования	+	+	+	+
Возможность сохранения/печати результатов голосования	+	+	-	+

Получение доступа к голосованию через приглашение	-	+	-	+
Уведомление о приглашении на голосование по почте	+	+	+	+

По данным составленной таблицы можно выявить основные необходимые функции.

Функции администратора голосования

1. Возможность входить в систему и иметь к себе привязку определённого голосования.
2. Возможность создавать голосование, настраивать варианты ответа и форму бюллетени.
3. Возможность видеть сводку по голосованию в виде диаграммы с указанием количества проголосовавших и процентным соотношением в процессе приёма голосов и по закрытию обсуждения.
4. Возможность сохранения и печати результатов голосования.
5. Возможность настраивать шаблоны для печати результатов.
6. Возможность приглашать участников на голосование.
7. Возможность отправлять приглашение на голосование по почте.

Функции голосующего

1. Возможность присоединиться к голосованию по приглашению.
2. Возможность заходить в личный профиль.
3. Возможность получить приглашение на голосование по почте.
4. Возможность выбрать один или несколько вариантов ответа.

5. Невозможность проголосовать в одном и том же голосовании повторно.

Таблица 1.3 – Все функции ИС дистанционное голосование

Необходимые	Желательные	Возможные	Отсутствующие
Создание голосования	Настройка выборов ответа	Настройка количества выбора ответов	-
Авторизация голосующих	Получение доступа к голосованию по приглашению	Получение доступа к голосованию по приглашению на почте	-
Просмотр сводки по результатам голосования	Просмотр сводки в процессе голосования	Выбор формата диаграммы для просмотра сводки	Возможность пользователя просматривать сводку в процессе голосования
Сохранение результатов голосования	Печать результатов голосования	Сохранение результатов голосования в разных форматах по выбору администратора	-

1.3 Анализ объекта исследования

Системный анализ^[10] – наука, занимающаяся проблемой принятия решений в условиях анализа большого количества информации различной природы. Системный анализ - в узком смысле представляет собой методологию принятия решений, - в широком смысле – синтез методологии общей теории систем, системного. подхода и системных методов обоснования и принятия решений.

С начала следует провести анализ по методологии ISA (Таблица 1.4). Благодаря нему мы выясним роли и составляющие систему объекты.

ISA^[11] – представляет собой выявление функциональных требований и требований пользователей, отображение архитектуры системы. Архитектуры информационных систем разрабатываются для каждого контура управления и процесса, а также архитектура интегрированных информационных систем. ISA – часть наиболее обширной области архитектур и моделей, имеющих значение для организации. Суть метода в том, что разным респондентам задают одни и те же вопросы, на которые в дальнейшем получают ответы с разных точек зрения.

Таблица 1.4 – ISA-анализ.

	А (ИС данные)	В (Бизнес-процессы)	С (Расположение обр. компонентов)	Д (Пользователи)	Е (Распределение событий и состояний)	Ф (Мотивация)
Планировщик (определяет границы системы)	Результаты анализа ИТ сферы	Создание голосований	Сервер	Пользователи, администратор	В момент работы с программой	Создание дистанционной формы голосования
Владелец (определяет концептуальную)	Кол-во прибыли	Создание голосований	Сервер компании или сторонний	Голосующие, администратор	Дата запуска, точка окупаемости	Получение прибыли

модель предприятия)						
Проектировщик (задаёт физическую модель системы)	Внутренние данные системы	Создание голосовых	Сервер, СУБД	Голосующие, администратор	Программирование, отладка, тестирование, сопровождение	Создание новой технологии
Конструктор (обеспечивает детализированное технологическое решение)	Документация системы	Создание документов	Сервер	Голосующие, администратор	Подготовка ИС к вводу в эксплуатацию	Юр. грамотно вывести ИС на рынок.
Субподрядчик (поставляет компоненты системы)	Данные о покупателе	Создание отчётов	Сервис	Покупатель	Физическая реализация ИС	Клиентская база

Далее стоит провести VCM-анализ^[12]. Метод ценностных цепочек (VCM) позволяет оценить конкурентные преимущества с помощью анализа всей цепочки видов деятельности в организации, начиная от получения сырья до конечной продукции, продаваемой и доставляемой потребителям.

Для проведения данного анализа надо ответить на следующие 5 пунктов:

1. Оценить информационную емкость продуктов и процессов.

Информационную ёмкость можно представить в виде результатов голосования.

2. Оценить роль ИТ в отраслевой структуре.

Информационная.

3. Выявить и ранжировать способы, с помощью которых ИТ создает конкурентное преимущество.

Простота использования, востребованность данного формата.

4. Рассмотреть, каким образом ИТ может создать новое направление в бизнесе.

Перевод многие ранее возможных только в очном формате мероприятий в дистанционный формат.

5. Разработать план, направленный на извлечение выгод от использования ИТ.

Можно продавать продукт, монетизировать посредством контекстной рекламы, создать платный функционал.

После следует сделать BPR анализ^[13]. BPR позволяет увидеть бизнес-процессы в понятной большинству людей форме.

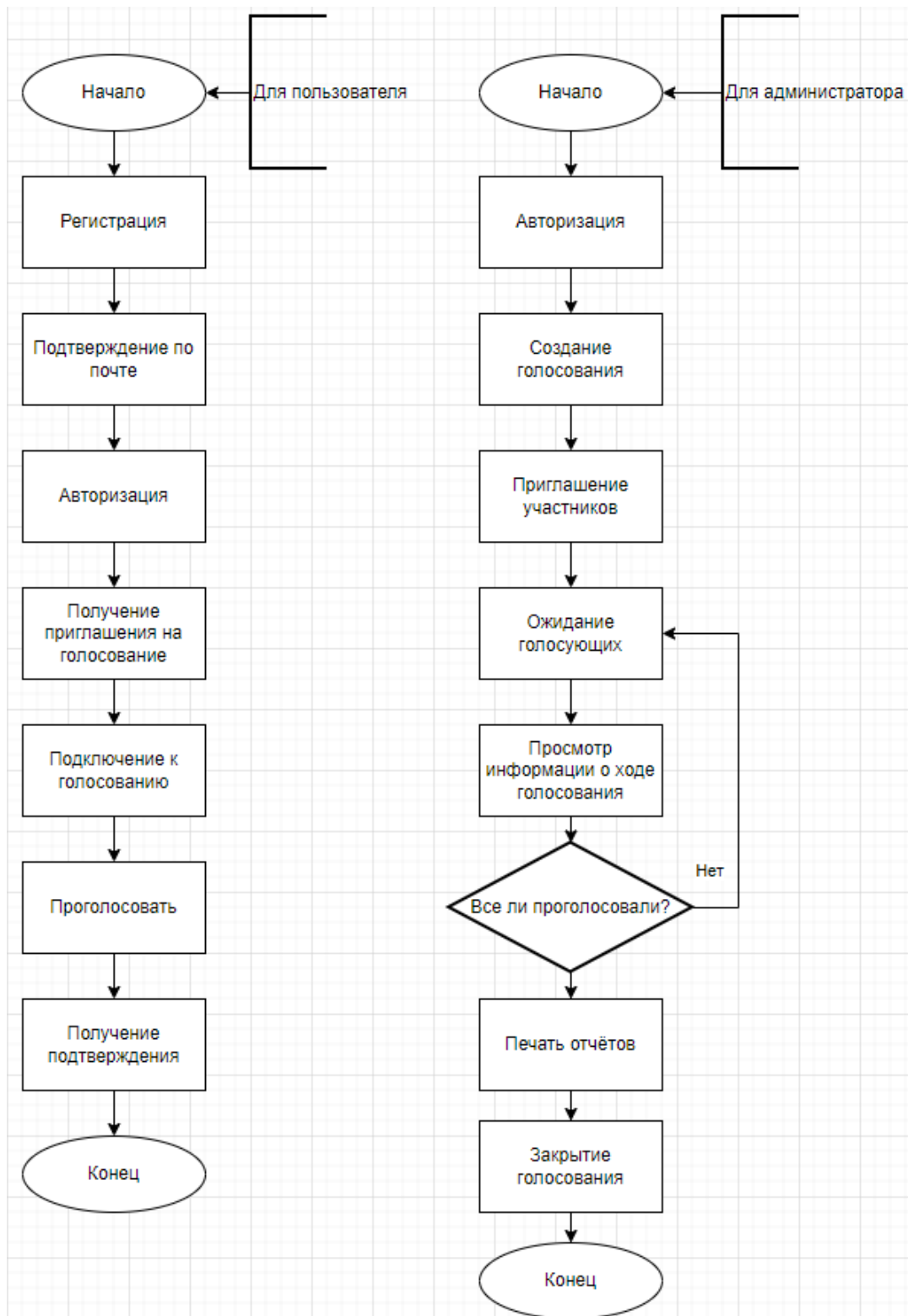


Рисунок 1.1. BRP-схема

SWOT-анализ^[14] представляет собой метод стратегического планирования, который заключается в выявлении сильных и уступающих аспектов системы (внешней и внутренней среды организации) посредством разделения их на четыре категории. Этот метод позволяет проанализировать

сильные и слабые стороны ИС для того, чтобы знать, в каких аспектах разрабатываемое ПО имеет преимущество и с какими угрозами оно может столкнуться.

Таблица 1.5. SWOT-анализ

	Положительное влияние	Отрицательное влияние
Внутренняя среда	Сильные стороны <ul style="list-style-type: none"> • Актуальность и востребованность решения • Упрощение деятельности • Экономия времени 	Слабые стороны <ul style="list-style-type: none"> • Отсутствие возможности пользовательской кастомизации • Не именитый разработчик
Внешняя среда	Возможности <ul style="list-style-type: none"> • Востребованность • Возможность модификации в интернет-площадку 	Угрозы <ul style="list-style-type: none"> • Большое количество конкурентов • Неисправности на серверном компьютере

1.4. Сроки реализации проекта

Для определения сроков реализации проекта используется диаграмма Ганта^[15]. Диаграмма Ганта представляет из себя отображение сроков выполнения задачи, проекта или бизнес-плана. В данном случае диаграмма отображает сроки разработки описываемого в данной работе проекта.

Данная разработка соответствует каскадной модели жизненного цикла, так как все фазы идут друг за другом. Однако в процессе работы над данным проектом возможны возвращения на предыдущие этапы с целью доработки.

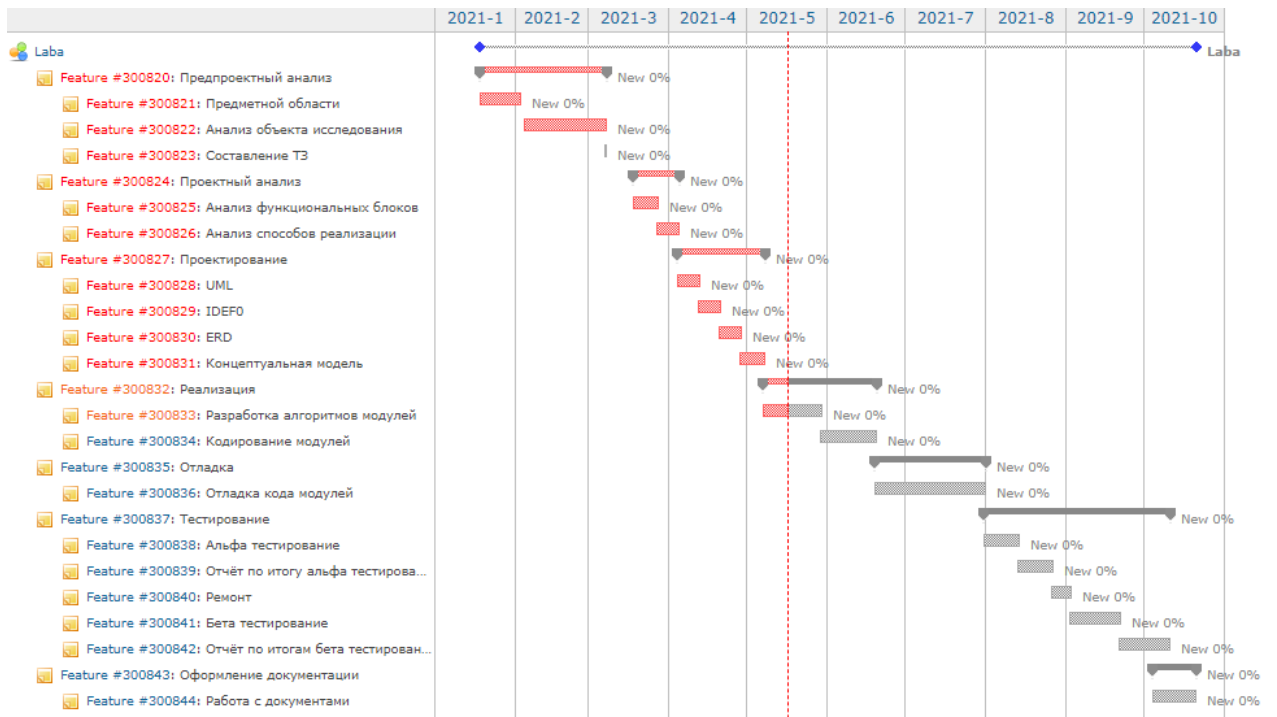


Рисунок 1.2. Диаграмма Ганта

2. ПРОЕКТИРОВАНИЕ ИНФОРМАЦИОННОЙ СИСТЕМЫ

2.1 UML моделирование

В качестве средств проектирования для данной работы были выбраны UML диаграммы. UML-диаграммы помогают при моделировании архитектуры проектов, в которой можно собрать крупные и более мелкие детали, что позволяет нарисовать схему основы приложения. По нему впоследствии будет строиться код и само приложение.

UML^[16] (англ. Unified Modeling Language — унифицированный язык моделирования) — язык графического описания для объектного моделирования в области разработки программного обеспечения, для моделирования бизнес-процессов, системного проектирования и отображения организационных структур.

Одна из задач UML — служить средством коммуникации внутри команды, а также при общении с заказчиками. В данном случае диаграммы будут служить для более простого и наглядного понимания работы приложения.

В данной работе представлены такие UML диаграммы как:

- диаграмма вариантов использования;
- диаграмма классов;
- диаграммы последовательности;
- диаграмма компонентов;
- диаграмма развёртывания.

Диаграмма вариантов использования^[17] (англ. use-case diagram) — диаграмма, описывающая, какой функционал разрабатываемой программной системы доступен каждой группе пользователей.

На этой диаграмме можно увидеть какой функционал предоставляется для каждой отдельной группы пользователей, которая может эксплуатировать разрабатываемое в данной ВКР приложение.

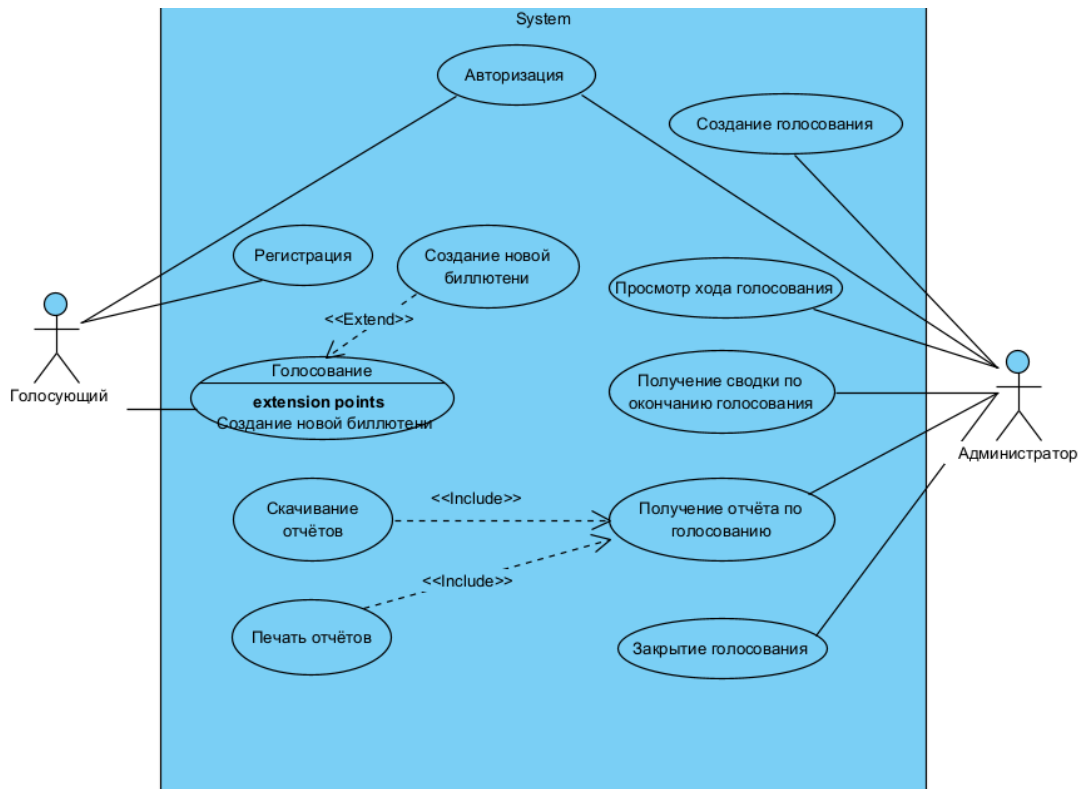


Рисунок 2.1. Use-case диаграмма

Диаграмма классов^[18] (англ. class diagram) — структурная диаграмма языка моделирования UML, демонстрирующая общую структуру иерархии классов системы, их коопераций, атрибутов (полей), методов, интерфейсов и взаимосвязей между ними.

Данная диаграмма, продемонстрированная в приложении 1, отображает классовую структуру информационной системы.

Диаграмма последовательности^[19] показывает жизненный цикл объектов, их взаимодействие в течении своего жизненного цикла в рамках прецедента.

Объекты расположены в горизонтальной последовательности и имеют свой жизненный цикл. Сами объекты на диаграмме последовательно обмениваются друг с другом, а также с пользователем(актёром) сообщениями.

Сообщения, которыми обмениваются объекты, бывают нескольких типов. Сообщения отображают как действия и процессы, так и свойства начального и целевого объектов.

Эта диаграмма позволяет более наглядно увидеть процессы, которые происходят при определённых взаимодействиях между пользователями и приложением.

На рисунке 2.2 показана диаграмма последовательности «Голосование». На ней показывается последовательность при выполнении голосования пользователем(голосующим). В ходе этого действия пользователь переходит на окно голосования, выбирает вариант ответа, нажимает на кнопку «Проголосовать» и таким образом голосует по рассматриваемой теме.

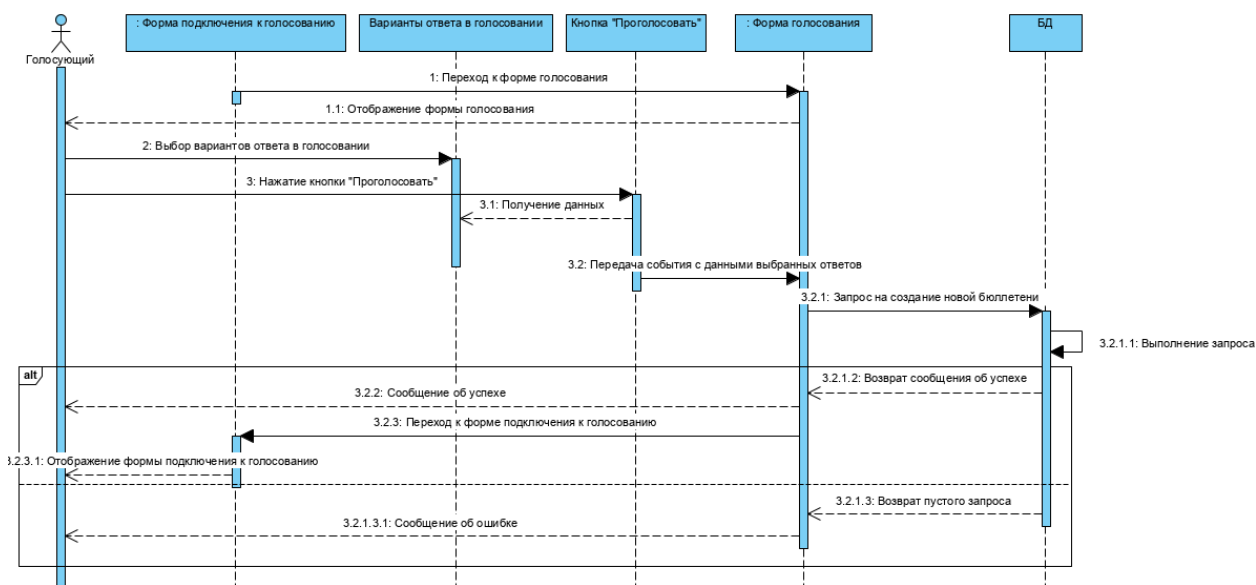


Рисунок 2.2. Диаграмма последовательности "Голосование"

На рисунке 2.3 показана диаграмма последовательности «Закрытие голосования». На ней показывается последовательность при выполнении закрытии голосования администратором. При закрытии голосования важно сначала получить отчёты в печатном виде, только после этого действия голосование можно будет закрыть.

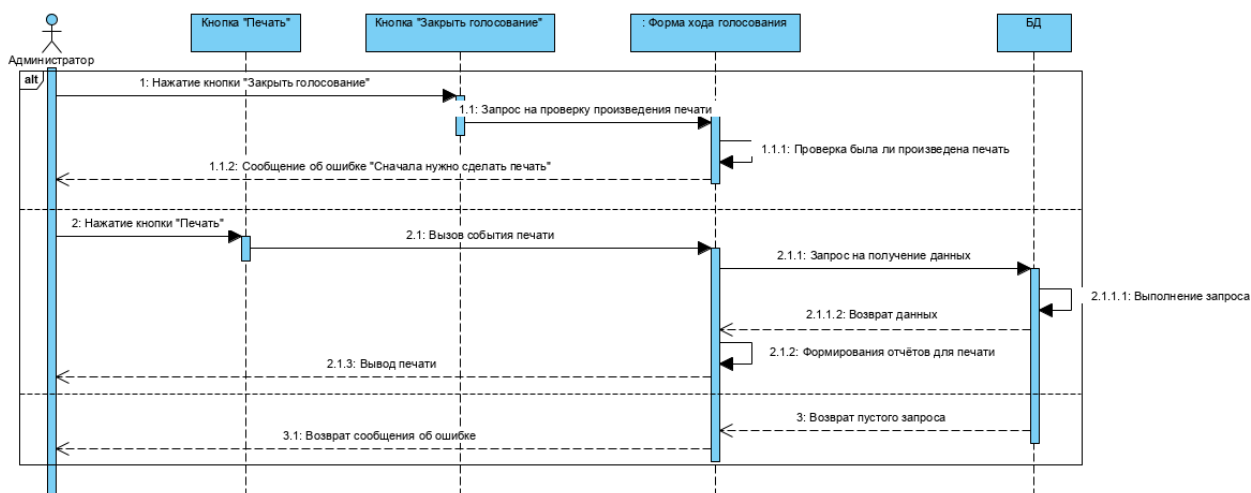


Рисунок 2.3. Диаграмма последовательности "Закрытие голосования"

Диаграмма компонентов^[20], в отличие от ранее рассмотренных диаграмм, описывает особенности физического представления системы. Диаграмма компонентов позволяет определить архитектуру разрабатываемой системы, установив зависимости между программными компонентами, в роли которых может выступать исходный, бинарный и исполняемый код.

На рисунке 2.4 можно увидеть диаграмму компонентов. Помимо компонентов, которые построены на основе упомянутых ранее интерфейсов, классов, также присутствует файл «SpringBootWeb.API», который является сторонней библиотекой, необходимой для функционирования сервера.

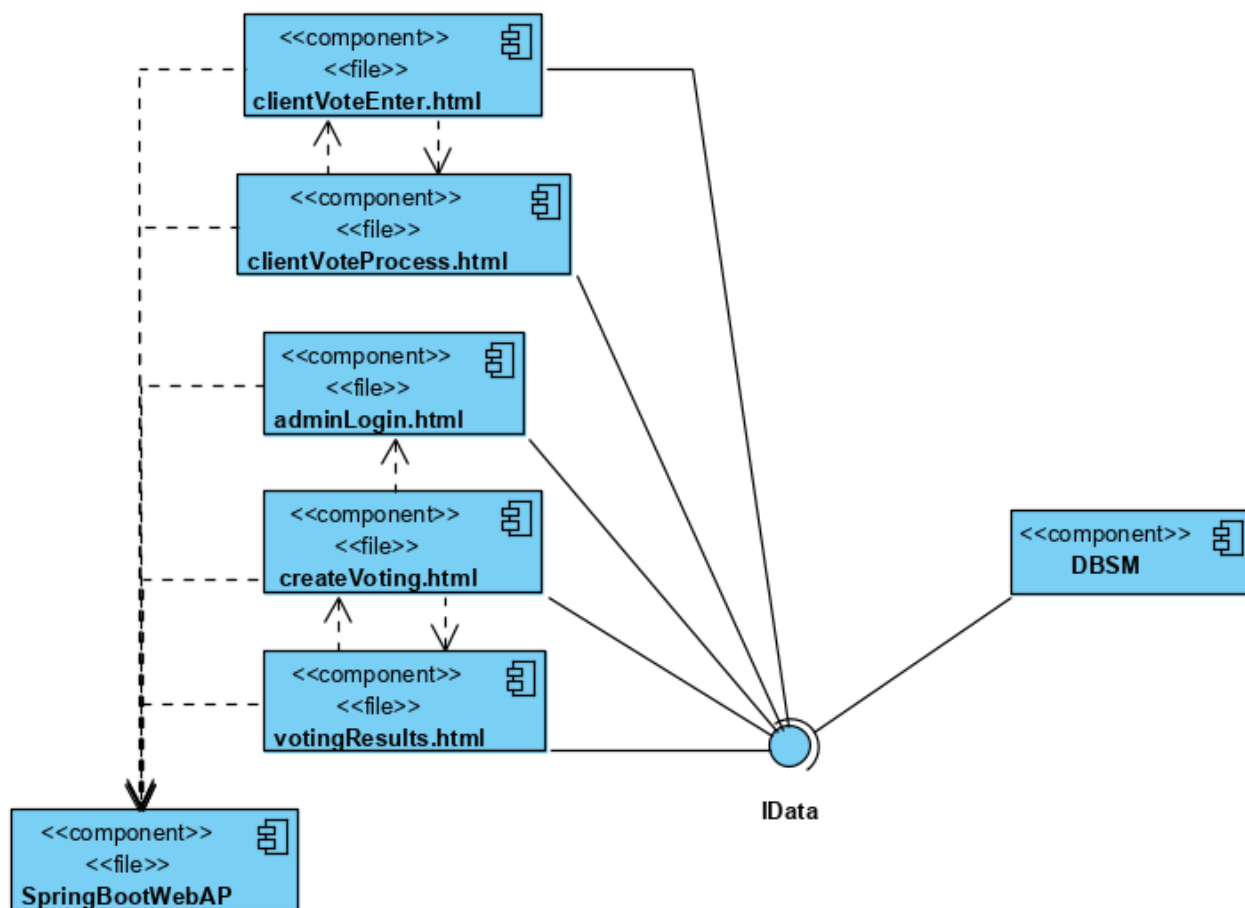


Рисунок 2.4. Диаграмма компонентов

Диаграммы развертывания^[21] используются для визуализации аппаратных процессоров/узлов/устройств системы, каналов связи между ними и размещения программных файлов на этом аппаратном обеспечении.

В разрабатываемой мной ИС, в соответствии с рисунком 2.5, на сервере находятся папки, которые содержат классы, интерфейсы обработчики и html-файлы соответственно названию папок, html-файл «index», который нужен для запуска сервера, а также файл со стилями «Style.css». Сервер посредством специализированного для баз данных TCP/IP обменивается данными с сервером базы данных, а также посредством локального TCP/IP взаимодействует с компьютером пользователя.

На сервере базы данных находится Database Management System, а пользовательском персональном компьютере находится браузер со встроенным в него Html5 Player'ом.

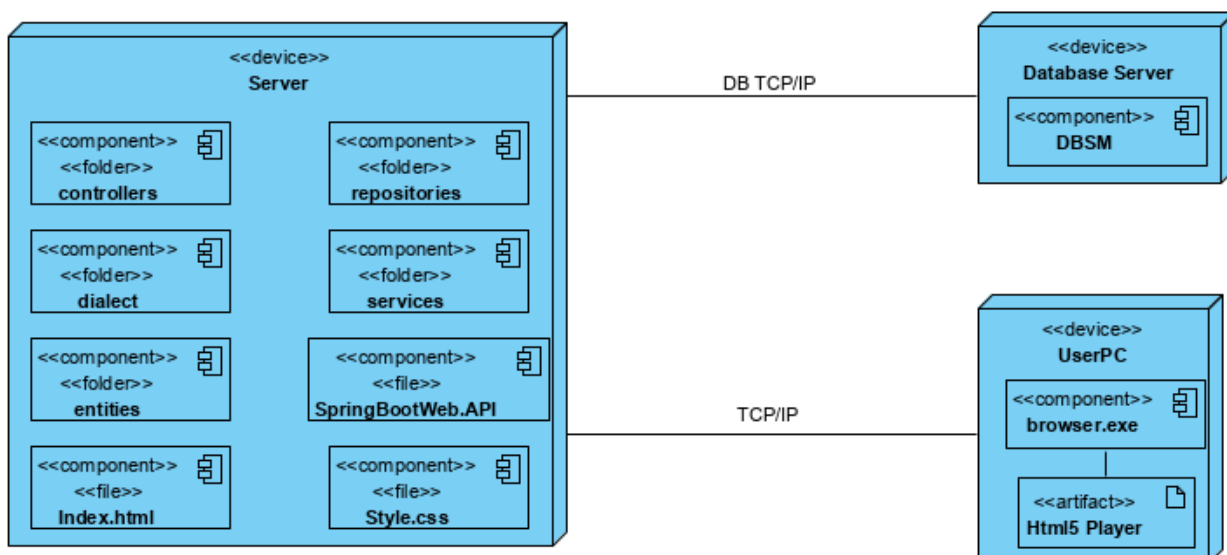


Рисунок 2.5. Диаграмма развёртывания

2.2 Моделирование бизнес-процессов в нотации BPMN

BPMN^[22] — система условных обозначений (нотация) и их описания в XML для моделирования бизнес-процессов^[23]. Данная диаграмма позволяет более наглядно увидеть процессы, которые происходят в информационной системе.

С её помощью можно более наглядно увидеть бизнес-процессы и понять алгоритм решения поставленных для каждого этапа работы системы задач.

На рисунке 2.6 представлена общая BPMN диаграмма. Согласно данной диаграмме, при запуске системы в браузере пользователем откроется окно авторизации. После успешной авторизации произойдёт переадресация на окно профиля пользователя. При открытии профиля пользователя нужные данные для заполнения нужных данных на странице используется обращение к базе данных. На окне профиля кнопка «Присоединиться к голосования», по нажатию которой произойдёт переадресация на страницу голосования. После голосования пользователя вернёт на окно профиля и если его ещё раз пригласили на голосование, то он по тому же алгоритму снова сможет проголосовать.

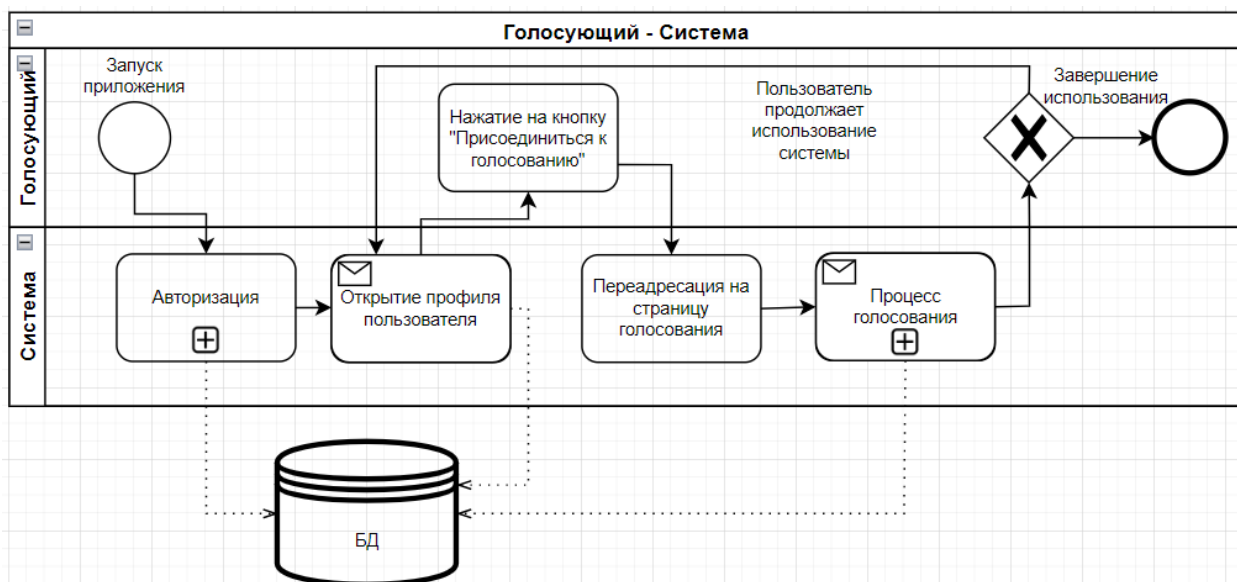


Рисунок 2.6. BPMN диаграмма

На рисунке 2.7 подробнее представлен процесс авторизации. Согласно алгоритму, который представлен на данной диаграмме, на форме авторизации голосующий вводит свои уникальные данные своей учётной записи, а именно логин и пароль, а после нажимает на кнопку «Войти», которая расположена на той же форме. После происходит проверка системой заполнения полей на корректность. К примеру, если одно поле будет пусто, то это считается некорректным вводом. В случае некорректного ввода пользователю нужно будет заново заполнить поля ввода логина и пароля. Далее происходит проверка данных пользователя с обращением к базе данных. В случае, если введённый логин не был найден в базе или пароль не соответствует учётной записи с введённым логином, то такой записи не существует и тогда пользователю придётся заново ввести в поля ввода логин и пароль. В случае, если данные найдены, то после этого вызывается переадресация на профиль пользователя.

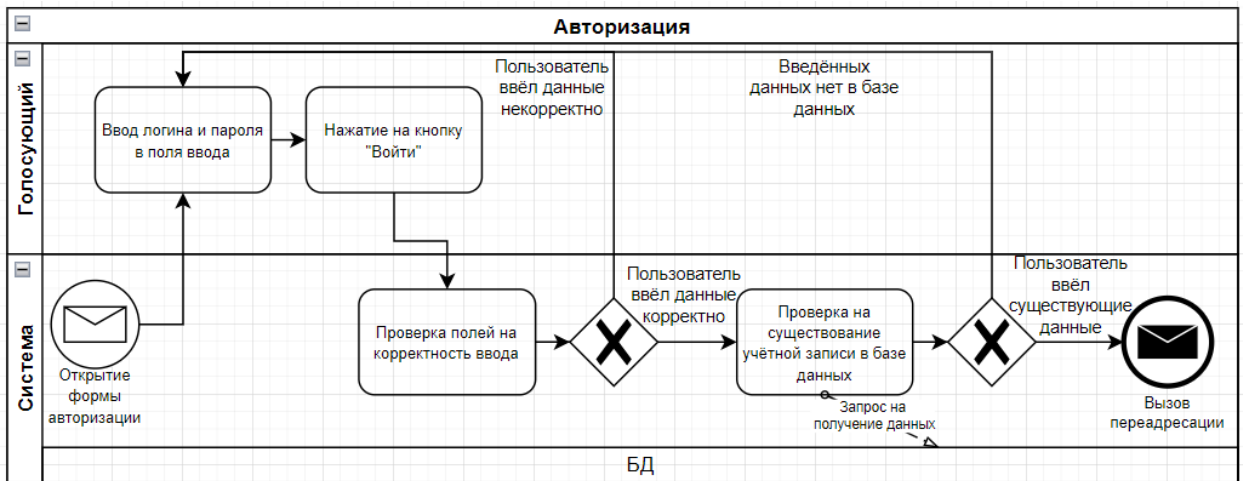


Рисунок 2.7. BPMN диаграмма «Авторизация»

На рисунке 2.8 подробнее представлен процесс голосования пользователем. Сначала открывается страница голосования. После для её полного отображения происходит запрос к базе данных на получение данных. Полученными данными заполняется тема голосования и страница полностью загружается и отображается. Далее пользователь выбирает варианты ответа по вопросу голосования, которые он считает верными для него, и нажимает на кнопку «Проголосовать». После нажатия на эту кнопку производится запрос к базе данных с изменением данных по голосованию. После этого происходит переадресация на страницу профиля пользователя.



Рисунок 2.8. BPMN диаграмма «Процесс голосования»

2.3 Схема базы данных

Схема базы данных^[24] включает в себя описания содержания, структуры и ограничений целостности, используемые для создания и поддержки базы данных.

В схеме на рисунке 2.9 можно увидеть базу данных. Она состоит из пяти таблиц:

- «voting», в которую записываются данные о суммарном количестве голосов по каждому варианту ответа и количество участников голосования;
- «vote», которая нужна для записи и создания бюллетеней с привязкой к нужному вопросу;
- «user», в которой записываются данные по учётным данным пользователей в системе;
- «recovery_link», в которой хранятся ссылки для восстановления пароля пользователем с привязкой к конкретной записи пользователя;
- «activate_link», в которой хранятся ссылки для активации учётной записи пользователей с привязкой к определённой записи пользователя.

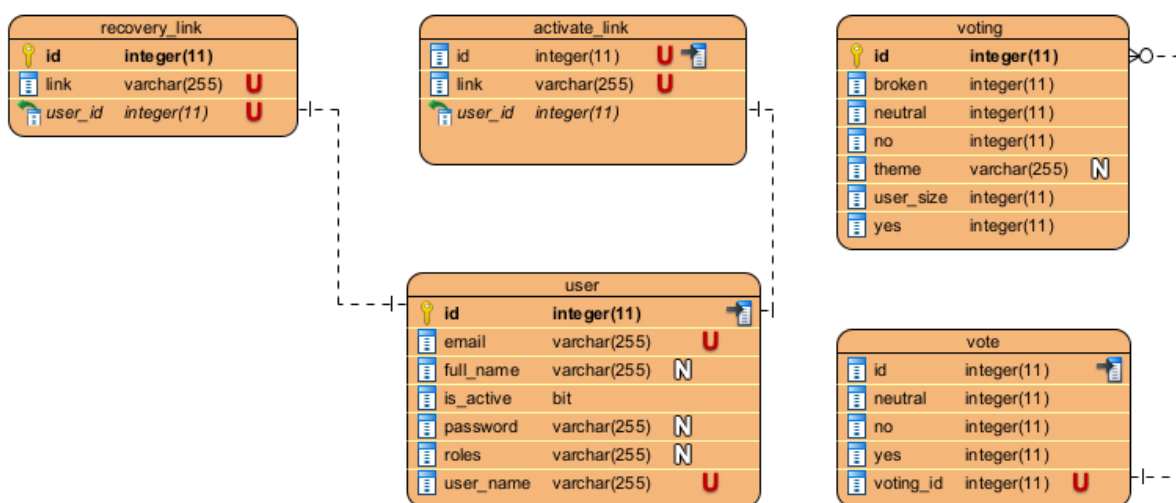


Рисунок 2.9. Схема базы данных

2.4. Схема восстановления и сохранения информации.

Восстановление информации^[25] – это процесс устранения разрушений данных с использованием сохраненной информации на некоторый момент времени (копии) и возможной корректуры с момента создания копии.

В свою же очередь восстановление базы данных — функция СУБД, которая в случае логических и физических сбоев приводит базу данных в актуальное и консистентное состояние.

Схема восстановления информации представлена на рисунке 2.10. Согласно этой схеме, в рабочей базе данных происходит бэкап данных. После, в случае сбоя, неисправности или других причин, которые служат для сбоя в базе данных, с помощью бэкапа происходит восстановление базы данных. В свою очередь, в процессе восстановления базы данных важную роль играют журнальные файлы.

Журнальные данные^[26] представляют собой данные о транзакциях, которые были произведены во время работы базы данных, и содержат в себе данные по изменениям, которые происходили во время работы.

Бэкап способствует восстановлению только файлов базы данных, а уже журнальные файлы, в свою очередь позволяют полноценно вернуть базу данных в состояние, в котором была база данных в определённый момент времени.

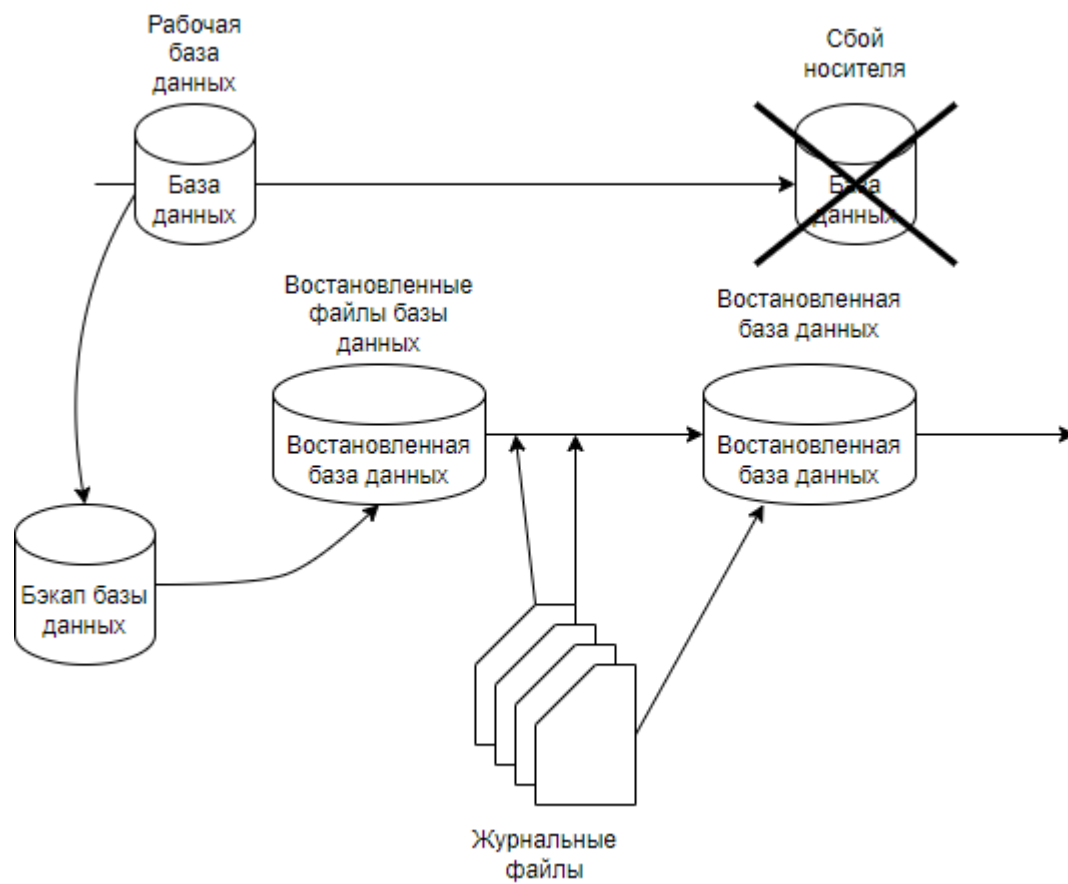


Рисунок 2.10. Схема восстановления информации

3. РАЗРАБОТКА ИНФОРМАЦИОННОЙ СИСТЕМЫ

3.1. Серверная сборка

В роли клиента в данной информационной системе выступает браузер, потому что все бизнес-процессы, вся логика и весь функционал, который необходим для функционирования системы, находится на сервере.

Для работоспособности функционала серверной части необходимо загрузить следующие библиотеки в соответствии с таблицей 3.1.

Таблица 3.1. Необходимые зависимости (библиотеки) для функционирования сервера.

Название зависимости (библиотеки)	Описание
org.springframework.boot:spring-boot-starter-web	Базовая библиотека для создания веб-приложений, включая RESTful приложения, на основе Spring MVC ^[27] . Использует встроенный контейнер сервлетов Tomcat ^[28] по умолчанию.
org.springframework.boot:spring-boot-starter-security	Фреймворк, который предоставляет функционал для обеспечения авторизации, регистрации, а также включает в себя базовые механизмы защиты информационной системы.
org.springframework.boot:spring-boot-starter-thymeleaf	Базовая библиотека для создания MVC веб-приложений с использованием движка шаблонов Java XML/XHTML/HTML5 Thymeleaf.

org.thymeleaf.extras:thymeleaf-extras-springsecurity5	Серверный движок шаблонов для автономных и веб-приложений.
org.springframework.boot:spring-boot-starter-data-jpa	Базовая библиотека для использования Spring Data JPA ^[29] вместе с Hibernate.
org.springframework.boot:spring-boot-starter-mail	Базовая библиотека для использования функционала по работе с электронной почтой.
org.mariadb.jdbc:mariadb-java-client	JDBC ^[30] драйвер для работы с MariaDB и MySQL.

Для загрузки зависимостей, необходимых для функционирования серверной части информационной системы, в проекте используется система автоматической сборки Gradle^[31].

Требуемые для функционирования зависимости, а также необходимые надстройки для сборки проекта, указываются в файле build.gradle в соответствии с рисунком 3.1.

В блоке plugins содержатся описания плагинов, которые в свою очередь предоставляют разработчику набор задач для выполнения системой сборки. В блоке repositories указываются репозитории с которых будут скачиваться указанные зависимости. В блоке dependencies указываются необходимые для загрузки библиотеки.

```

plugins {
    id 'org.springframework.boot' version '2.6.4'
    id 'io.spring.dependency-management' version '1.0.11.RELEASE'
    id 'java'
}

group = 'ru.vorobyov'
version = '0.0.1-SNAPSHOT'
sourceCompatibility = '1.8'

repositories {
    mavenCentral()
}

dependencies {
    implementation 'org.springframework.boot:spring-boot-starter-data-jpa'
    // https://mvnrepository.com/artifact/org.mariadb.jdbc/mariadb-java-client
    implementation group: 'org.mariadb.jdbc', name: 'mariadb-java-client', version: '3.0.4'
    implementation 'org.springframework.boot:spring-boot-starter-security'
    implementation 'org.springframework.boot:spring-boot-starter-thymeleaf'
    implementation 'org.springframework.boot:spring-boot-starter-web'
    implementation 'org.thymeleaf.extras:thymeleaf-extras-springsecurity5'
    // https://mvnrepository.com/artifact/org.springframework.boot/spring-boot-starter-mail
    implementation 'org.springframework.boot:spring-boot-starter-mail:2.6.6'
    developmentOnly 'org.springframework.boot:spring-boot-devtools'
    // https://mvnrepository.com/artifact/org.apache.poi/poi-ooxml
    implementation group: 'org.apache.poi', name: 'poi-ooxml', version: '5.2.2'
    testImplementation 'org.springframework.boot:spring-boot-starter-test'
    testImplementation 'org.springframework.security:spring-security-test'
}

```

Рисунок 3.1. build.gradle

Важно отметить, что после подключения зависимостей некоторые библиотеки требуют создания конфигурационных классов и методов для настройки их работы в проекте.

Таким образом, для настройки фреймворка Spring Security^[32] необходимо переопределить перегрузки метода `configure` путём расширения класса `WebSecurityConfigurerAdapter` в соответствии с рисунком 3.2. На представленном фрагменте кода указываются ограничения для пользователей по их роли в системе. Благодаря этому форму авторизации и регистрации могут пройти все пользователи, а к ресурсам администратора имеют доступ только пользователи с ролью «Администратор».

```

@Override
protected void configure(HttpSecurity http) throws Exception {
    http.authorizeRequests().antMatchers(...antPatterns: "/admin/**").hasRole(ADMIN) ExpressionUrlAutho
        .antMatchers(...antPatterns: "/").hasAnyRole(ADMIN, USER, VOTER)
        .antMatchers(...antPatterns: "/voter/**").hasRole(VOTER)
        .antMatchers(...antPatterns: "/all/**").permitAll()
        .and().formLogin() FormLoginConfigurer<HttpSecurity>
        .and().logout().logoutRequestMatcher(new AntPathRequestMatcher( pattern: "/logout"))
        .logoutSuccessUrl("/login?logout");

    http
        .formLogin() FormLoginConfigurer<HttpSecurity>
        .loginPage("/login")
        .failureUrl("/login?error")
        .permitAll()
        .and() HttpSecurity
        .logout() LogoutConfigurer<HttpSecurity>
        .permitAll();
}

```

Рисунок 3.2. Фрагмент класса WebSecurityConfigurerAdapter

Для настройки подключения к базе данных следует задать настройки для библиотеки Spring JPA. Для того, чтобы это сделать, нужно в конфигурационном файле application.properties указать необходимую информацию об используемой базе данных в соответствии с рисунком 3.3.

```

spring.datasource.url=jdbc:mariadb://141.8.195.31:3306/voting_db
spring.datasource.username=voting_admin
spring.datasource.password=dCh11AMtHzJzG*1F
spring.jpa.hibernate.ddl-auto=update
spring.datasource.driver-class-name=org.mariadb.jdbc.Driver

```

Рисунок 3.3. Настройки для JPA в application.properties

Для настройки использования функционала почтового сервиса следует прописать необходимую информацию в конфигурационном файле application.properties для работы Spring Mail^[33] в соответствии с рисунком 3.4.

```

spring.mail.host=smtp.gmail.com
spring.mail.port=587
spring.mail.username=anonymousvoting82@gmail.com
spring.mail.password=ydmfegopurhrtikk
spring.mail.properties.mail.smtp.auth=true
spring.mail.properties.mail.smtp.starttls.enable=true

```

Рисунок 3.4. Настройки для Spring Mail в application.properties

Используемые при разработке информационной системы библиотеки и фреймворки навязывают своим присутствием некоторые архитектурные особенности проекта.

Так, для работы с базой данных посредством Spring JPA требуется создать классы трёх типов: записей, репозитория и сервисов.

Класс типа «Запись», пример которого представлен на рисунке 3.5, представляет собой описание таблицы и всех её полей. Экземпляры данного класса служат для хранения информации с записей в соответствующих таблицах в базе данных.

```
@Entity
@Table(name = "vote")
public class Vote {
    @Id
    @Column(name = "ID")
    @GeneratedValue(strategy=GenerationType.SEQUENCE, generator = "id_Sequence")
    @SequenceGenerator(name = "id_Sequence", sequenceName = "ID_SEQ")
    private int id;

    private int yes;
    private int no;
    private int neutral;
    @ManyToOne(fetch = FetchType.LAZY, cascade = CascadeType.ALL)
    @JoinColumn( unique = true, nullable = false)
    private Voting voting;
```

Рисунок 3.5 – Пример класса типа «Запись»

Классы типа «Репозиторий» создаются фреймворком Spring. Обязанностью разработчика является создать интерфейс, пример которого представлен на рисунке 3.6, с указанием нужного репозитория для того, чтобы после можно было создать экземпляр объекта соответствующего класса и использовать его при помощи инъекции зависимости. Экземпляр данного класса представляет собой механизм, который нужен для того, чтобы получать информацию из базы данных и представлять её в форме объектов.

```

@Repository
public interface UserRepository extends JpaRepository<User, Integer> {

    Optional<User> findByUserName(String userName);

}

```

Рисунок 3.6. Пример интерфейса типа «Репозиторий»

Класс типа «Сервис», пример которого представлен на рисунке 3.7, содержит в себе функционал для работы с классами типа «Репозиторий» и отвечает за правильность сохранения информации из сервера в базу данных и корректность заполнения классов типа «Запись» информацией из базы данных.

```

@Service
public class UserDetailsServiceImpl implements UserDetailsService {

    @Autowired
    private UserRepository userRepository;

    @Override
    public UserDetails loadUserByUsername(String userName) throws UsernameNotFoundException {

        Optional<User> user = userRepository.findByUserName(userName);

        user.orElseThrow(() -> new UsernameNotFoundException(userName + " not found."));

        return user.map(UserDetailsImpl::new).get();

    }

    public User findUserById(int userId) {
        Optional<User> userFromDb = userRepository.findById(userId);
        return userFromDb.orElse(new User());
    }
}

```

Рисунок 3.7. Фрагмент класса типа «Сервис»

Основная серверная логика описана в классах-контроллерах. Классы данного типа служат для обработки запросов по одному или нескольким адресам. Фрагмент кода класса-контроллера^[34] представлен на рисунке 3.8.

```

@Controller
public class AdminControlPanelController {
    @Autowired
    private UserDetailsServiceImpl userService;
    @GetMapping("/admin/panel")
    public String getPage(Model model){
        AdditionVotersToVotingDto userForm = getAllUsersFromDb();
        if (!model.containsAttribute( attributeName: "userForm"))
            model.addAttribute( attributeName: "userForm", userForm);
        return "admincontrolpanel";
    }

    @GetMapping("/admin/panel/edit/{id}")
    public String getPageUpdateUser(@PathVariable("id") int id, Model model) {
        if (!userService.isUserExist(id))
            return "redirect:/admin/panel?notFound";
        model.addAttribute( attributeName: "userForm", userService.findUserById(id));
        return "adminpaneledituser";
    }
}

```

Рисунок 3.8. Фрагмент класса-контроллера

Основные классы-контроллеры и их описание представлены в таблице 3.2.

Таблица 3.2 Классы-контроллеры и их описание

Название класса-контроллера	Описание
AdminCabinetController	Обработка запросов, связанных с кабинетом администратора
AdminControlPanelController	Обработка запросов, связанных с панелью управления администратора
AdminVotingCreateController	Обработка запросов, связанных со страницей создания голосований
AdminVotingInviteController	Обработка запросов, связанных с заполнением таблицы пользователей для их приглашения на голосование
AdminVotingResultController	Обработка запросов, связанных с просмотром

	статистики по ходу голосования
AdminVotingResultGetReportController	Обработка запросов, связанных с печатью отчёта по голосованию и бюллетеней
UserProfileController	Обработка запросов, связанных с профилем пользователя
UserRedirectToVotingController	Обработка запросов, связанных с перенаправлением пользователя на другой адрес
VoterProcessController	Обработка запросов, связанных с голосованием пользователя
MainController	Обработка запросов, связанных с перенаправлением после авторизации
PasswordRecoveryController	Обработка запросов, связанных с восстановлением пароля
RegistrationController	Обработка запросов, связанных с регистрацией пользователей

Формирование страниц, отображающихся на клиенте (в браузере), происходит посредством движка шаблонов Thymeleaf. Он позволяет внедрять в html-страницы логику на специальном языке Thymeleaf для правильного отображения объектов, получаемых с сервера. Фрагмент использования Thymeleaf показан на рисунке 3.9.

```

<th:block th:each="user, iterStat : ${userList}">
  <tr>
    <td>
      <input type="checkbox" class="form-check-input my-2" th:field="${userList[__${iterStat.index}__].isVoter}">
    </td>
    <td hidden>
      <input type="text" class="form-check-input my-2" th:field="${userList[__${iterStat.index}__].id}">
    </td>
    <td hidden>
      <input type="text" class="form-check-input my-2" th:field="${userList[__${iterStat.index}__].email}">
    </td>
    <td th:text="${userList[__${iterStat.index}__].id}"></td>
    <td th:text="${userList[__${iterStat.index}__].userName}"></td>
    <td th:text="${userList[__${iterStat.index}__].email}"></td>
    <td th:text="${userList[__${iterStat.index}__].fullName}"></td>
  </tr>
</th:block>

```

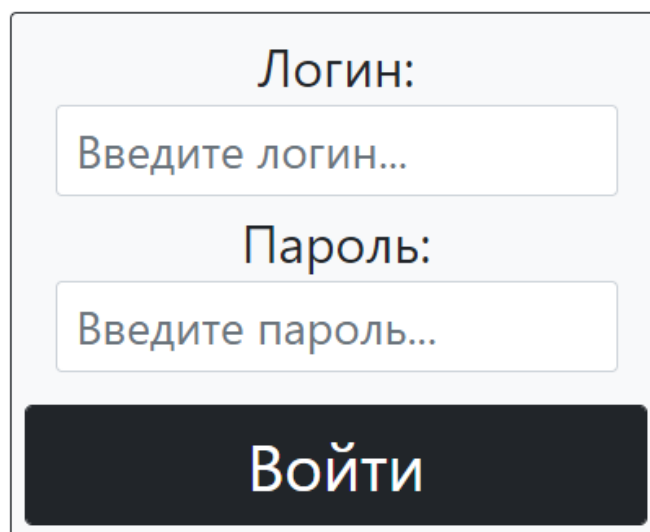
Рисунок 3.9. Фрагмент html-страницы с использованием Thymleaf

3.2. Взаимодействие с интерфейсом информационной системы

При открытии информационной системы в браузер всех пользователей встречается интерфейс приложения^[35]. Именно посредством него строится вся модель взаимодействия с системой.

Благодаря механизмам авторизации, которые были добавлены в проект посредством фреймворка Spring Security и настройкам конфигурационного класса, описанного в главе 3.1, первой встречается всех пользователей, которые ещё не вошли в систему под своей учётной записью, форма авторизации, которая представлена на рисунке 3.10.

Авторизация



Логин:

Пароль:

Войти

[Зарегистрироваться](#)

Забыли пароль? [Восстановить пароль](#)

Рисунок 3.10. Форма авторизации

В случае, если пользователь ещё не имеет своей учётной записи, то ему требуется пройти регистрацию. Для этого нужно нажать на ссылку «Зарегистрироваться» под основной формой авторизации.

После перехода по ссылке открывается форма регистрации пользователя, которая показана на рисунке 3.11, на которой можно увидеть поля для заполнения своими учётными данными. В случае, если пользователь уже имеет свою учётную запись, под основной формой регистрации присутствует форма «Войти», которая перенаправит пользователя обратно на форму авторизации.

Регистрация

Логин

ФИО

Email

Пароль

Зарегистрироваться

[Войти](#)

Рисунок 3.11. Форма регистрации

В случае, если при регистрации указаны некорректные данные или заполнены не все поля, то в форме регистрации появится сообщение с информацией о характере неверного ввода в соответствии с рисунком 3.12.

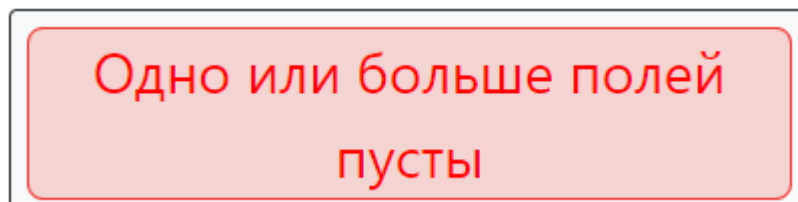
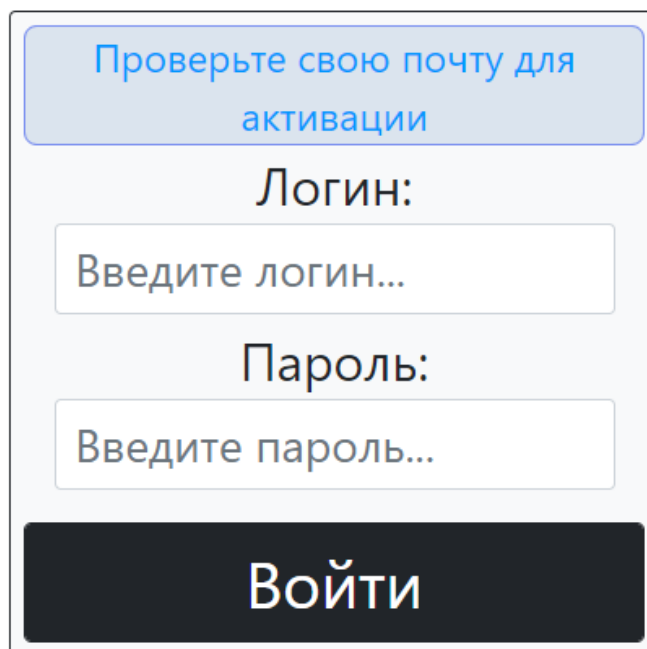


Рисунок 3.12. Сообщение об ошибке

После заполнения всех полей пользователь будет перенаправлен на форму авторизации, в которой будет выведено сообщение о том, что нужно

проверить свою почту для активации учётной записи, что можно увидеть на рисунке 3.13.

Авторизация



Проверьте свою почту для активации

Логин:

Пароль:

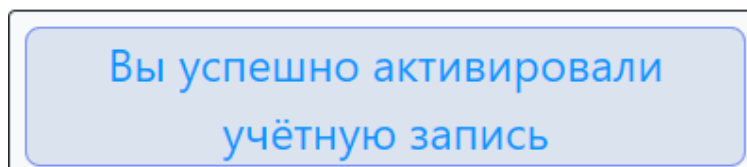
Войти

[Зарегистрироваться](#)

Забыли пароль? [Восстановить пароль](#)

Рисунок 3.13. Форма авторизации с сообщением о проверке почты для активации

Далее пользователю нужно зайти на свою почту, на которое придёт письмо с ссылкой для активации записи. После перехода по ссылке пользователя перенаправит на форму авторизации с сообщением об успешной активации, что можно увидеть на рисунке 3.14.



Вы успешно активировали учётную запись

Рисунок 3.14. Сообщение об успешной активации учётной записи

При авторизации с учётной записи администратора откроется кабинет администратора, на котором будет навигационная панель с кнопками перехода в профиль, панель управления, что можно увидеть на рисунке 3.15, и кнопкой

выхода, показанной на рисунке 3.16, стандартной для всех страниц при входе в систему с активированной учётной записи.

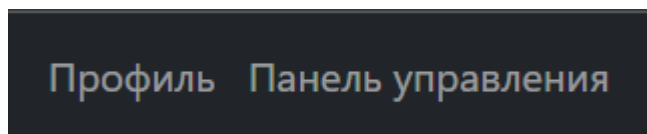


Рисунок 3.15. Кнопки профиля и панели управления на навигационной панели



Рисунок 3.16. Кнопка выхода из учётной записи на навигационной панели

В основном пространстве страницы располагается название страницы и кнопка создания анонимного голосования в соответствии с рисунком 3.17. В случае, если в данный момент проходит голосование, то вместо кнопки по созданию анонимного голосования будет находиться кнопка для перехода к просмотру статистики идущего голосования в соответствии с рисунком 3.18.

Кабинет администратора

Создать анонимное голосование

Рисунок 3.17. Кнопка создания анонимного голосования

Кабинет администратора

Вернуться к просмотру хода голосования

Рисунок 3.18. Кнопка возвращения к идущему голосованию

При переходе в панель управления по нажатию кнопки «Панель управления» в навигационной панели, администратора перенаправит на окно с панелью управления. На панели навигации находится кнопка «Кабинет», которая перенаправляет администратора обратно в кабинет администратора, и кнопка «Выйти» с функционалом, который описывался до этого. На основном

пространстве страницы располагается таблица со всеми пользователями, в соответствии с рисунком 3.19, с пагинацией, с возможностью поиска записей по словам, и кнопкой для редактирования выбранной записи.

Show entries Search:

Id	Логин	Email	ФИО	Роли	Активирован	Изменить
5	admin	123	123	ROLE_ADMIN	true	
6	user1	user1@user.ru	user1 user1 user1	ROLE_ADMIN	true	
7	user2	user2@user.ru	user2 user2 user2	ROLE_USER	true	
8	user3	user3@user.ru	user3 user3 user3	ROLE_USER	true	
9	user4	user4@user.ru	user4 user4 user4	ROLE_USER	true	
10	user5	user5@user.ru	user5 user5 user5	ROLE_USER	true	
13	DenisStudent	denis.vorobyov.00@gmail.com	Воробьев Денис Игоревич	ROLE_USER	true	

Previous Next

Рисунок 3.19. Панель управления пользователями

При нажатии на кнопку изменения откроется форма изменения данных выбранной учётной записи в соответствии с рисунком 3.20.

Логин

ФИО

Email

Роль

Активирован

ИЗМЕНИТЬ

Рисунок 3.20. Форма изменения данных учётной записи

При нажатии кнопки «Создать анонимное голосование» в кабинете администратора, показанной на рисунке 3.17, откроется форма создания голосования в соответствии с рисунком 3.21. В поле на этой форме вводится тема, которая будет выведена на голосовании. При нажатии на кнопку «Добавить вопрос» откроется дополнительное поле для создания ещё одного вопроса на голосовании.

Создание голосования

Рисунок 3.21. Форма изменения данных учётной записи

При нажатии на кнопку «Создать» администратора перенаправит на окно приглашения пользователей к голосованию с таблицей пользователей в соответствии с рисунком 3.22. Для приглашения пользователя к голосованию нужно в таблице напротив строки учётной записи пользователя поставить галочку и нажать кнопку «Пригласить».

Пригласить	Id	Логин	Email	ФИО
<input type="checkbox"/>	7	user2	user2@user.ru	user2 user2 user2
<input type="checkbox"/>	8	user3	user3@user.ru	user3 user3 user3
<input type="checkbox"/>	9	user4	user4@user.ru	user4 user4 user4
<input type="checkbox"/>	10	user5	user5@user.ru	user5 user5 user5
<input type="checkbox"/>	13	DenisStudent	denis.vorobyov.00@gmail.com	Воробьев Денис Игоревич

Пригласить

Рисунок 3.21. Форма приглашения пользователей к голосованию

После нажатия кнопки «Пригласить» выбранным пользователям на почту придёт оповещение о приглашении на голосование и администратора

перенаправит на окно просмотра статистики хода голосования. В основном пространстве страницы содержится статистика по каждому вопросу, обсуждаемому на голосовании, в соответствии с рисунком 3.22. Под статистикой по ходу голосования находятся кнопки «Обновить», «Получить отчёт», «Закрыть голосование». При нажатии кнопки «Обновить» обновляется статистика по вопросам голосования. При нажатии кнопки «Получить отчёт» формируется отчёт по голосованию и формируются бюллетени. По нажатию кнопки «Закрыть голосование» голосование закрывается и администратор перенаправляется в кабинет администратора.

Тестовый вопрос 2

За

Против

Воздержались

Испорченные бюллетени

Не проголосовали

2

Количество участников:2

Рисунок 3.22. Статистика по вопросу в голосовании

При авторизации через учётную запись пользователя открывается профиль пользователя. В навигационной панели, как у администратора, находятся кнопки «Профиль» и «Выйти». В основном пространстве страницы если пользователь не приглашён на голосование, то он увидит надпись, что он ещё не был приглашён в соответствии с рисунком 3.23. Если пользователь был приглашён к участию на голосование, то вместо надписи, представленной на рисунке 3.23, он увидит надпись, что был приглашён, а также кнопку перехода на страницу с голосование в соответствии с рисунком 3.24.

Профиль

Вас ещё не пригласили на голосование

Рисунок 3.23 – Надпись в профиле пользователя о том, что он ещё не приглашён на голосование

Профиль

Вас пригласили на голосование

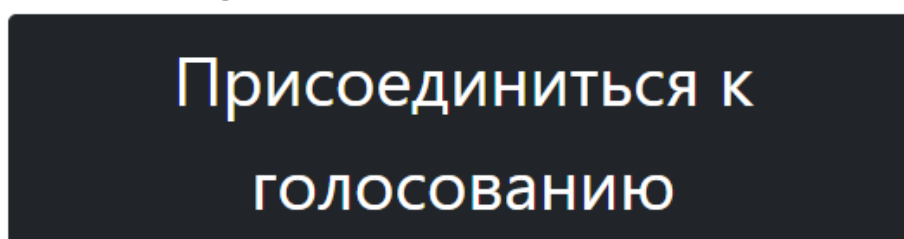


Рисунок 3.24. Надпись в профиле пользователя о том, что он приглашён на голосование и кнопка перехода

После нажатия кнопки «Присоединиться к голосованию» пользователь будет перенаправлен на страницу голосования. На этой странице будут вопросы, которые подняты на обсуждение в голосовании, как представлено на рисунке 3.25. После выбора ответов пользователь должен проголосовать, нажав кнопку «Проголосовать», которая находится под блоком вопросов. После данного действия пользователя направит обратно в профиль пользователя, а надпись приглашения с кнопкой заменится надписью, представленной на рисунке 3.23.

Тестовый вопрос 2

- За
- Против
- Воздержусь

Рисунок 3.25. Вопрос поднятый на обсуждение в голосовании

3.3. Надёжность системы поддержки дистанционного голосования

Надёжность программного обеспечения^[36] – вероятность работы программного обеспечения без критических ошибок и отказов в течении определённого времени.

Основные виды ошибок в программном обеспечении:

- Логические ошибки – ошибки данного характера происходят в результате неверного выполнения алгоритма. Такие ошибки возникают в связи с некорректной записью уравнений, некорректном постановлении условий, указания неверных диапазонов и т.д.
- Ошибки вычислений – ошибки данного характера возникают в результате составления неверных математических выражений или некорректных результатов.
- Ошибки ввода-вывода – подобные ошибки возникают в связи с неверным управлением ввода и вывода, выходных данных и т.д.
- Ошибки манипулирования данными – ошибки данного характера возникают в связи с неверным присвоением значений различным

переменным, неправильного их наименования, неверной задачи длины операнды и т.д.

Сначала для проверки надёжности используем метод, основанный на модели Нельсона-Коркорэна^[37].

Для проверки надёжности по этой модели запустим сервер 200 раз и посмотрим сколько запусков будет с отказом.

После запуска программы на одном устройстве количество отказов равно 11. Надёжность сервера считается согласно формуле 3.1. В числителе находится количество запусков с отказом, а в знаменателе общее количество запусков.

$$R(n) = 1 - \frac{n^-}{n}. \quad (3.1)$$

Вычислим надёжность по формуле 3.2.

$$R(200) = 1 - \frac{9}{200} = 0,955. \quad (3.2)$$

Из данного вычисления следует, что надёжность сервера 0,96.

С аналогичным количеством запусков проверим надёжность сервера базы данных.

Сервер базы данных запустился с отказом 2 раза. Рассчитаем надёжность сервера базы данных по формуле 3.3.

$$R(200) = 1 - \frac{2}{200} = 0,99. \quad (3.3)$$

Таким образом, надёжность серверной части информационной системы по модели Нельсона-Коркорэна составляет 0,955, а надёжность сервера базы данных 0,99.

Далее посчитаем надёжность по модели Шумана^[38]. По данной модели нужно провести проверки в несколько этапов, каждый из которых представляет из себя запуск программы по набору тестовых данных. Выявление в течении одной проверки ошибки записываются и регистрируются, но не исправляются. По завершении одного этапа зарегистрированные ошибки исправляются, происходит корректировка тестовых наборов и запускается следующий этап.

Результаты этапов тестирования серверной части отображены в таблице 3.3.

Таблица 3.3. Этапы тестирования серверной части

Этапы	t, ч	m
1	15	4
2	20	2
3	20	1

Определим количество ошибок N из зависимости по формуле 3.4.

$$\sum_{i=1}^k m_i * \frac{\sum_{i=1}^k t_i}{\sum_{i=1}^k \frac{m_i}{N - n_{i-1}}} = \sum_{i=1}^k (N - n_{i-1}). \quad (3.4)$$

По расчёту количества ошибок из данной зависимости определили, что N равно 8.

Надёжность рассчитывается по формуле 3.5.

$$R(t) = e^{-\lambda * t}. \quad (3.5)$$

Интенсивность отказов^[39] λ равна вычисляется по формуле 3.6.

$$\lambda = (N - n) * \frac{\sum_{i=1}^k \frac{m_i}{N - n_{i-1}}}{\sum_{i=1}^k t_i}. \quad (3.6)$$

Подставив параметры получаем, что интенсивность отказов равна 0,026404669.

Далее вычисляем надёжность по формуле 3.5 и получаем, что она равна 0,973940886.

После этого проводим этапы тестирования для сервера базы данных соответственно таблице 3.4.

Таблица 3.4. Этапы тестирования сервера базы данных

Этапы	t, ч	m
1	15	2
2	20	1
3	20	1

По расчёту количества ошибок с помощью формулы 3.4 определили, что N равно 5. Интенсивность отказов, рассчитанная по формуле 3.6, равна 0,028484848. Надёжность, рассчитанная по формуле 3.5, равна 0,97191702.

Таким образом, надёжность серверной части информационной системы по модели Шумана составляет 0,973940886, а надёжность сервера базы данных 0,97191702.

После этих подсчётов рассчитаем надёжность системы исходя из схемы взаимодействия частей информационной системы^[40], представленной на рисунке 3.26. Поскольку клиентом в данной системе выступает браузер, то его в расчёт надёжности брать не будем.



Рисунок 3.26. Схема взаимодействия частей системы Сервера подключены через параллельное соединение.

Рассчитаем надёжность системы по модели Нельсона-Коркорэна с помощью формулы 3.7.

$$P = \frac{(0,955 + 0,99)}{2} = 0,9725. \quad (3.7)$$

Далее рассчитаем надёжность системы по модели Шумана с помощью формулы 3.8.

$$P = \frac{(0,973940886 + 0,97191702)}{2} = 0,972928953. \quad (3.8)$$

После расчётов надёжности по различным моделям мы можем получить среднюю вероятность безотказной работы системы. Для этого проведём расчёт согласно формуле 3.9.

$$P = \frac{(0,9725 + 0,972928953)}{2} = 0,9727144765. \quad (3.9)$$

Из произведённых вычислений следует, что вся система поддержки режима дистанционного голосования имеет надёжность равную 0,9727144765. Данной вероятности безотказной работы информационной системы достаточно для длительного безошибочного пользования системой.

ЗАКЛЮЧЕНИЕ

В настоящее время всё больше различных видов деятельности приобретают дистанционный формат.

В связи с последними событиями, а именно по причине вспышки пандемии, многие были вынуждены перейти на самоизоляцию, в следствии чего многие люди оценили всё удобство удалённых информационных систем.

В настоящее время такие приложения, в числе которых находится описываемый в данной работе проект, пользуются в настоящее время большой актуальностью, а в дальнейшем будущем востребованность похожих ИС будет только расти.

Полученные в ходе выполнения данного проекта опыт и знания неоценимы, так как создание и работа над востребованными системами позволяет лучше понимать как сам процесс разработки, так и актуальные течения на рынке.

СПИСОК ЛИТЕРАТУРЫ

1. Медач. Портал для молодых врачей и студентов медицинских университетов. [Электронный ресурс]. — Режим доступа: <https://medach.pro/post/2726>
2. ТАСС. События в России и мире. [Электронный ресурс]. — Режим доступа: https://tass.ru/ekonomika/8312275?utm_source=yandex.ru&utm_medium=organic&utm_campaign=yandex.ru&utm_referrer=yandex.ru
3. Studwork. Фриланс-биржа помощи студентам. [Электронный ресурс]. — Режим доступа: <https://studwork.org/spravochnik/oformlenie/diplomnye-raboty/obekt-i-predmet-issledovaniya>
4. Studwork. Фриланс-биржа помощи студентам. [Электронный ресурс]. — Режим доступа: <https://studwork.org/spravochnik/oformlenie/diplomnye-raboty/obekt-i-predmet-issledovaniya>
5. Java. Официальный сайт. [Электронный ресурс]. — Режим доступа: https://www.java.com/ru/download/help/whatis_java.html
6. Spring IO. Единая платформа корпоративного уровня для создания современных приложений. [Электронный ресурс]. — Режим доступа: <https://spring.io/projects/spring-boot>
7. Skillbox. Образовательная платформа с онлайн-курсами. [Электронный ресурс]. — Режим доступа: https://skillbox.ru/media/code/chto_takoe_html/
8. Skillbox. Образовательная платформа с онлайн-курсами. [Электронный ресурс]. — Режим доступа: https://skillbox.ru/media/code/chto_takoe_css/
9. Thymeleaf. Официальный сайт. [Электронный ресурс]. — Режим доступа: <https://www.thymeleaf.org/>

10. Прохорова И.А. Теория систем и системный анализ: учеб. пособие / И.А. Прохорова. – Челябинск: Издательский центр ЮУрГУ, 2013. – 8 с.
11. StudFiles. Файловый архив для студентов. [Электронный ресурс]. — Режим доступа: <https://studfile.net/preview/4452586/page:3/>
12. StudFiles. Файловый архив для студентов. [Электронный ресурс]. — Режим доступа: <https://studfile.net/preview/4452586/page:3/>
13. StudFiles. Файловый архив для студентов. [Электронный ресурс]. — Режим доступа: <https://studfile.net/preview/4452586/page:3/>
14. Голованов А. А. проектирование информационных систем: учеб. пособие ч. 2 / Голованов А. А. – Киров: ФГБОУ ВПО «ВятГУ», 2014. – 27 с.
15. Голованов А. А. проектирование информационных систем: учеб. пособие ч. 2 / Голованов А. А. – Киров: ФГБОУ ВПО «ВятГУ», 2014. – 78 с.
16. Моделирование на UML. Интернет-книга. [Электронный ресурс]. — Режим доступа: http://book.uml3.ru/sec_1_1
17. Моделирование на UML. Интернет-книга. [Электронный ресурс]. — Режим доступа: http://book.uml3.ru/sec_2_2
18. Википедия. Бесплатная электронная энциклопедия. [Электронный ресурс]. — Режим доступа: https://ru.wikipedia.org/wiki/%D0%94%D0%B8%D0%B0%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B0_%D0%BA%D0%BB%D0%B0%D1%81%D1%81%D0%BE%D0%B2
19. IT-GOST.RU. Электронная библиотека стандартов оформления проектной документации. [Электронный ресурс]. — Режим доступа: http://www.it-gost.ru/articles/view_articles/94

20. Википедия. Бесплатная электронная энциклопедия. [Электронный ресурс]. — Режим доступа:
https://ru.wikipedia.org/wiki/%D0%94%D0%B8%D0%B0%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B0_%D0%BA%D0%BE%D0%BC%D0%BF%D0%BE%D0%BD%D0%B5%D0%BD%D1%82%D0%BE%D0%B2
21. Creately. Visual workspace software to collaborate on brainstorming, planning, executing, and analyzing projects. [Электронный ресурс]. — Режим доступа: <https://creately.com/blog/ru/uncategorized-ru/%D1%83%D1%87%D0%B5%D0%B1%D0%BD%D0%BE%D0%B5-%D0%BF%D0%BE%D1%81%D0%BE%D0%B1%D0%B8%D0%B5-%D0%BF%D0%BE-%D0%B4%D0%B8%D0%B0%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B5-%D1%80%D0%B0%D0%B7%D0%B2%D1%91%D1%80%D1%82/>
22. BPMN Specification. Providing the most accurate and up to date information about BPMN 2.0. [Электронный ресурс]. — Режим доступа: <https://www.bpmn.org/>
23. Википедия. Бесплатная электронная энциклопедия. [Электронный ресурс]. — Режим доступа:
<https://ru.wikipedia.org/wiki/%D0%91%D0%B8%D0%B7%D0%BD%D0%B5%D1%81-%D0%BF%D1%80%D0%BE%D1%86%D0%B5%D1%81%D1%81>
24. Википедия. Бесплатная электронная энциклопедия. [Электронный ресурс]. — Режим доступа:
https://ru.wikipedia.org/wiki/%D0%A1%D1%85%D0%B5%D0%BC%D0%B0_%D0%B1%D0%B0%D0%B7%D1%8B_%D0%B4%D0%B0%D0%BD%D0%BD%D1%8B%D1%85

25. Studref. Студенческие реферативные статьи и материалы. [Электронный ресурс]. — Режим доступа: https://studref.com/690818/informatika/vosstanovlenie_informatsii_baza_h_dannyh
26. Life-Prog. Блог о языках программирования. [Электронный ресурс]. — Режим доступа: https://life-prog.ru/2_47007_zhurnalnie-fayli-v-subd-HyTech.html
27. Baeldung. In-depth, to-the-point tutorials on Java, Spring, Spring Boot, Security, and REST. [Электронный ресурс]. — Режим доступа: <https://www.baeldung.com/spring-mvc-tutorial>
28. Apache Tomcat. Официальный сайт. [Электронный ресурс]. — Режим доступа: <https://tomcat.apache.org/>
29. JavaRush. Интерактивный онлайн-курс по изучению Java-программирования с нуля. [Электронный ресурс]. — Режим доступа: <https://javarush.ru/groups/posts/2259-jpa--znakomstvo-s-tekhnologiey>
30. Oracle Help Center. Getting started guides, documentation, tutorials, architectures, and more content for Oracle products and services. [Электронный ресурс]. — Режим доступа: <https://docs.oracle.com/javase/tutorial/jdbc/overview/index.html>
31. Gradle. Официальный сайт. [Электронный ресурс]. — Режим доступа: <https://gradle.org/>
32. Spring IO. Единая платформа корпоративного уровня для создания современных приложений. [Электронный ресурс]. — Режим доступа: <https://spring.io/projects/spring-security>
33. Baeldung. In-depth, to-the-point tutorials on Java, Spring, Spring Boot, Security, and REST. [Электронный ресурс]. — Режим доступа: <https://www.baeldung.com/spring-email>

34. Baeldung. In-depth, to-the-point tutorials on Java, Spring, Spring Boot, Security, and REST. [Электронный ресурс]. — Режим доступа: <https://www.baeldung.com/spring-controllers>
35. Bigenc. Большая российская энциклопедия - электронная версия. [Электронный ресурс]. — Режим доступа: https://bigenc.ru/technology_and_technique/text/4426494
36. OSP. Издательство «Открытые системы». [Электронный ресурс]. — Режим доступа: <https://www.osp.ru/os/1994/04/178540>
37. Монахов Ю. М. Функциональная устойчивость информационных систем. В 3 ч. Ч. 1. Надежность программного обеспечения: учеб. пособие / Монахов Ю. М.; Владим. гос. ун-т. – Владимир: Издво Владим. гос. ун-та, 2011. – 30 с.
38. Монахов Ю. М. Функциональная устойчивость информационных систем. В 3 ч. Ч. 1. Надежность программного обеспечения: учеб. пособие / Монахов Ю. М.; Владим. гос. ун-т. – Владимир: Издво Владим. гос. ун-та, 2011. – 28 с.
39. Википедия. Бесплатная электронная энциклопедия. [Электронный ресурс]. — Режим доступа: https://ru.wikipedia.org/wiki/%D0%98%D0%BD%D1%82%D0%B5%D0%BD%D1%81%D0%B8%D0%B2%D0%BD%D0%BE%D1%81%D1%82%D1%8C_%D0%BE%D1%82%D0%BA%D0%B0%D0%B7%D0%BE%D0%B2
40. ОБЖ.ру. Информационно-образовательный веб-сайт, посвященный обучению и воспитанию основам безопасности. [Электронный ресурс]. — Режим доступа: <http://www.obzh.ru/nad/4-5.html>

ПРИЛОЖЕНИЕ 1

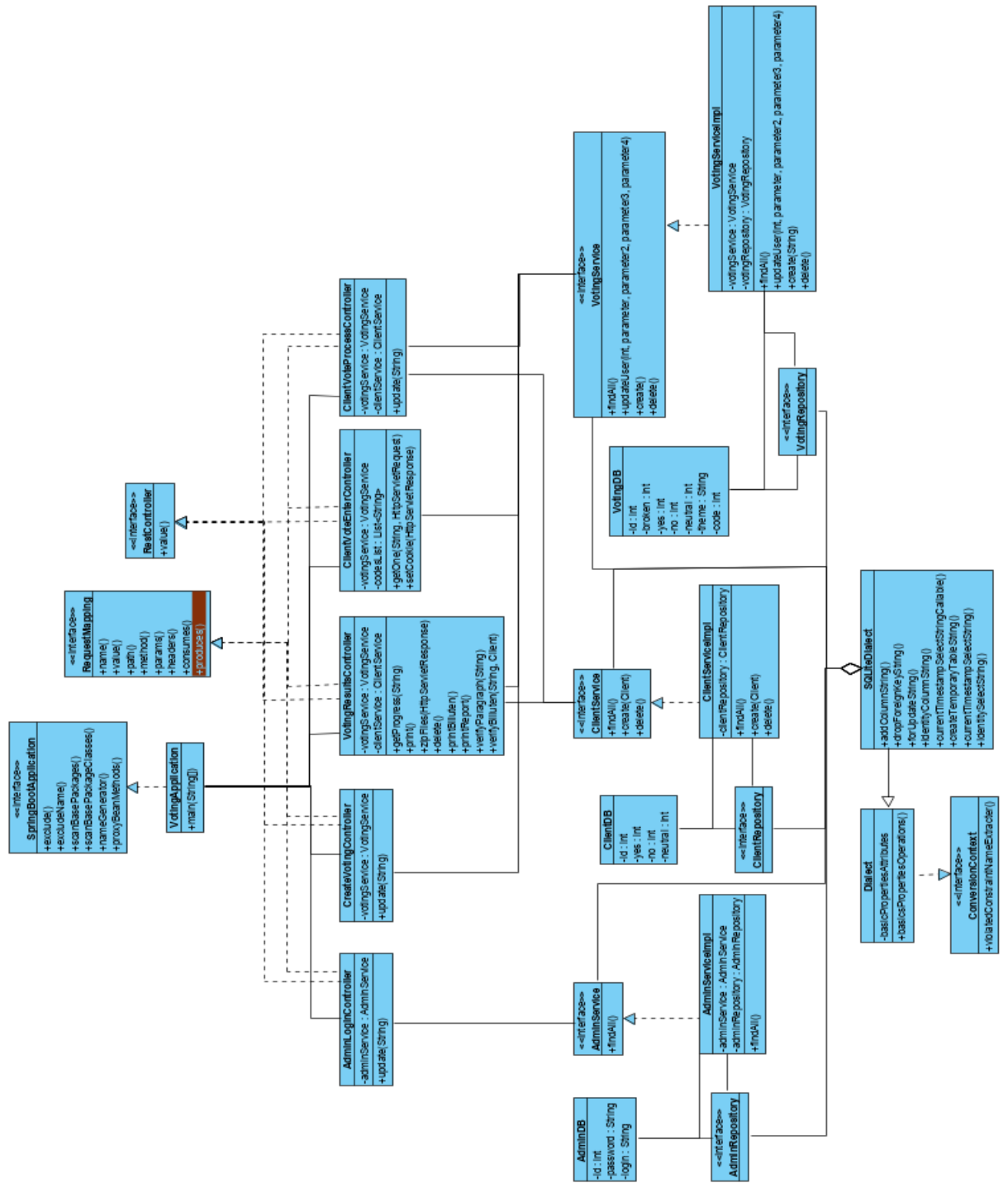


Рисунок 1. Классовая диаграмма