

МИНОБРНАУКИ РОССИИ
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ
ВЫСШЕГО ОБРАЗОВАНИЯ
“ВОРОНЕЖСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ”

Факультет компьютерных наук

Кафедра программирования и информационных технологий

Контроль деятельности сотрудников в условиях удаленного формата работы.

*Курсовая работа
по дисциплине
Технологии программирования*

09.03.02. *Информационные системы и технологии*

Программирование и информационные технологии

Преподаватель _____ *В.С. Тарасов, ст. преп* _____.2021

Обучающийся _____ *А.А. Ударцев, 3 курс, д/о*

Обучающийся _____ *Д.Э. Лисаев, 3 курс, д/о*

Руководитель _____ *К.В. Сиволапов, ст. преп*

Воронеж 2019

Содержание

Введение	3
1. Постановка задачи.....	4
2. Анализ предметной области	5
2.1. Анализ существующих решений	5
2.2. Продуктовые воронки	6
2.3. Анализ задачи.....	7
2.3.1. Варианты использования приложения	7
2.3.2. Взаимодействие компонентов системы.....	8
2.3.3. Варианты состояния системы	14
2.3.4. Варианты действия в системе	16
2.3.5. Развертывание приложения.....	19
3. Реализация	20
3.1. Анализ средств реализации	20
3.2. Архитектура приложения.....	21
3.3. Реализация серверной части приложения	25
3.4. Реализация настольного приложения	26
3.5. Реализация WEB-приложения	27
3.6. Сущности.....	27
4. Описание пользовательского интерфейса	28
4.1. Интерфейс WEB-приложения	28
4.1.1 Главная страница	28
4.1.2 Страница профиля	29
4.1.3 Страница команд	29
4.1.4 Страница команды	29
4.1.5 Страница задач	30
4.1.6 Страница задачи	30
4.1.7 Страница статистики	31
4.2. Интерфейс настольного приложения	31
4.2.1 Экран входа	32
4.2.2 Главный экран	33
4.2.3 Экран выбора задачи	34
4.2.4 Экран выбора команды	34
4.2.5 Экран настроек	35
Заключение	37
Список использованных источников	38

Введение

За последние годы люди убедились в преимуществах удаленной работы, а также осознали ее незаменимость при чрезвычайных ситуациях. Такой вид деятельности имеет ряд плюсов, это: и отсутствие необходимости добираться до рабочего места, и возможность сотрудников более рационального распоряжаться собственным временем, а также многое другое. Но есть и обратная сторона.

Вирус Covid-19, разбушевавшийся в начале 2020 года по всему миру, заставил не просто перевести многие компании своих сотрудников на удаленную работу, но и показал нам множество недоработок, изъянов, которые необходимо решить для возможности вести комфортную, а главное продуктивную деятельность дистанционно.

Одной из важнейших проблем, вставших перед владельцами компаний, являлась проблема возможности полноценного и достоверного расчета эффективности того или иного сотрудника. Необходимо было правильно оценить объем проделанной работы и потраченное сотрудником время. Возлагать ответственность за это на самих сотрудников - не самая лучшая идея, ведь рядовой работник вполне может предоставить недостоверные данные для собственной выгоды, из-за чего возникла потребность в программном обеспечении, которое могло бы вести сбор и анализ данных независимо от сотрудника.

Данная курсовая работа показывает процесс создания приложения, которое решает данный вопрос. Основной задачей приложения является предоставление возможности отслеживать деятельность своих сотрудников в рабочее время. Здесь и далее будут рассмотрены цели, задачи, специфика и результаты работы по созданию приложения.

1. Постановка задачи

Цель нашего приложения создать оптимальную среду для организации управления удаленной работников сотрудников компаний. А конкретно это возможность создания задач, их распределение между участниками и затем, впоследствии, просмотр статистики, включающая в себя информацию о ходе работы участника над выданной ему задачей. Важно сделать интерфейс для работы над организацией и приложение, которое будет автоматически отправлять данные о ходе работы сотрудника над задачей на сервер.

Основная задача данного проекта - создать приложение удаленного контроля работников, полностью соответствующее подготовленному техническому заданию.

Задачи, которые необходимо решить в ходе развития проекта:

1. Собрать необходимую информацию об исследуемой предметной области
2. Провести анализ полученных данных
3. Разработать концепции работы приложения.
4. Выбрать наилучший вариант и создать модель программы
5. Написать код приложения, отладить его, откорректировать до стабильной работоспособности
6. Тестирование и доработка приложения

2. Анализ предметной области

2.1. Анализ существующих решений

Разрабатываемое приложение не является единственным в своем роде, и у него есть аналоги, например Worksmart, Hubstaff, Actitime.

Hubstaff - мощный инструмент контроля удаленной работы как для крупной компании, так и для маленьких фирм.

Сильные стороны приложения:

1. Настройка требуемого функционала
2. Подсчет затрат на разработку проекта
3. Множество интеграций с сервисами разработки как GitHub, Trello...

Слабые стороны:

1. Ограниченность функционала в бесплатной версии

Worksmart - инструмент разработанный Crossover который помогает удаленным сотрудникам более эффективно управлять своим временем и получать удобную рабочую среду.

Сильные стороны инструмента:

1. Возможность ведения дискуссий
2. Совместная работа
3. Сравнение статистики по времени

Слабые стороны:

1. Высокий порог вхождения
2. Слабая база интеграций

Actitime - программное обеспечение для учета рабочего времени и развития бизнеса с помощью интеллектуальных данных.

Сильные стороны:

1. Установка времени работы сотрудников и задач
2. Внедрен своя доска задач
3. Подсчет расходов и сопоставлением с бюджетом

Слабые стороны:

1. Нет интеграции с внешними инструментами разработки

2.2. Продуктовые воронки

Рассмотрим количество шагов, которое необходимо сделать для достижение конечной цели администратора команды - создания рабочего пространства:

1. Пользователь регистрируется в системе.
2. Пользователь создаёт команду.
3. Пользователь добавляет сотрудников в команду.
4. Пользователь создает список задач.
5. Пользователь распределяет задачи между сотрудниками в команде.

Также, рассмотрим сколько шагов потребуется участнику команды для старта выполнения задачи:

1. Пользователь регистрируется в системе.
2. Пользователь присоединяется к команде, становясь участником команды.
3. Пользователь авторизируется через настольное приложения.
4. Пользователь начинает выполнять выбранную задачу.

Таким образом для начало работы суммарно потребуется 9 шагов с двух сторон: администратора команды и её участника.

2.3. Анализ задачи

2.3.1. Варианты использования приложения

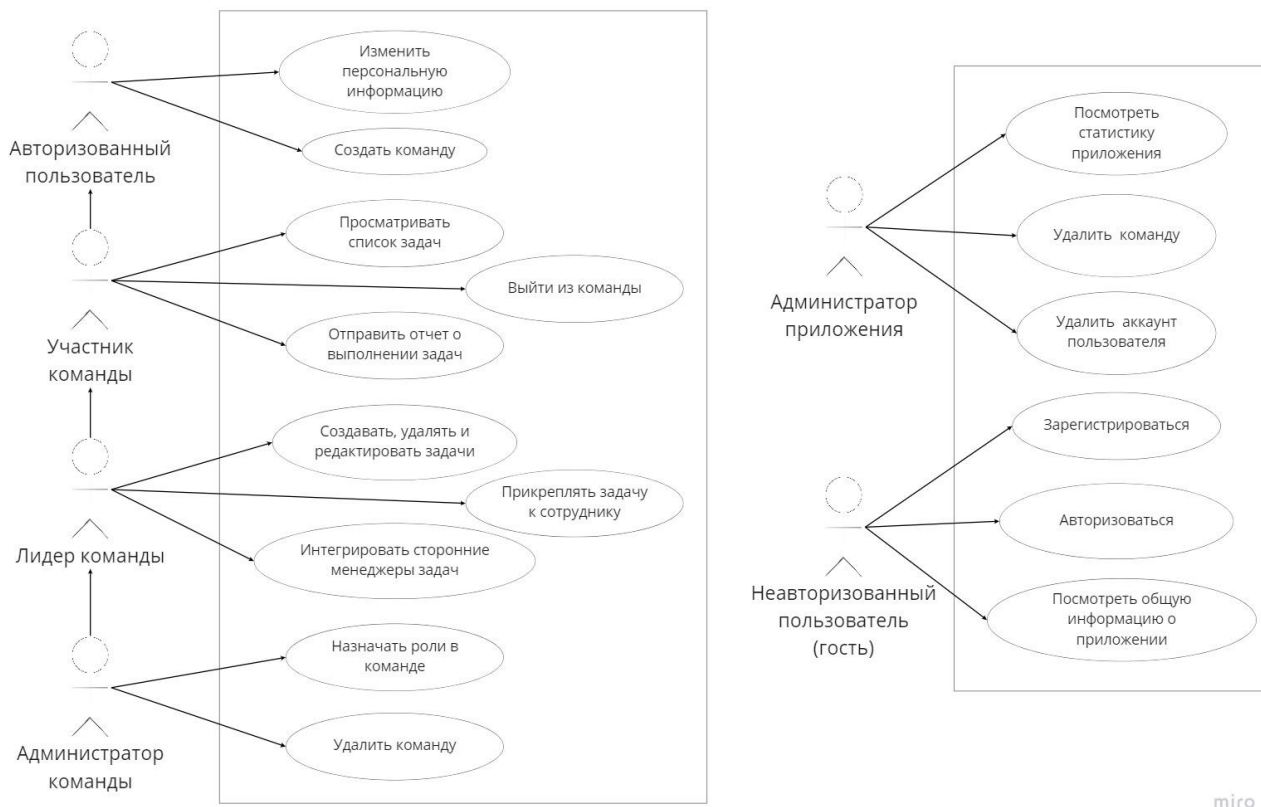


Рисунок 1 - Диаграмма прецедентов

На данной диаграмме, изображенной на Рисунке 1, представлены функции актеров (пользователей) в системе.

Неавторизованный пользователь: регистрация, авторизация и просмотр общей информации о приложении.

Авторизованный пользователь, как участник команды: отправлять отчетность о выполнении назначенных ему задач, просмотр списка назначенных ему задач, выйти из команды.

Авторизованный пользователь, как лидер команды (дополнительно к функциям участника команды): создание, удаление, редактирование задач, назначение задач участникам команды, внешняя интеграция задач со сторонними ресурсами, поддерживаемыми приложением.

Авторизованный пользователь, как администратор команды (дополнительно к функциям лидера команды): назначение ролей в команде, удаление команды.

Администратор приложения: удаление команд и пользователей, просмотр статистики приложения.

2.3.2. Взаимодействие компонентов системы

Диаграммы последовательности нужна для отображения взаимодействий объектов и субъектов в динамике, отображая временные особенности передачи и приема сообщений между объектами и субъектами.

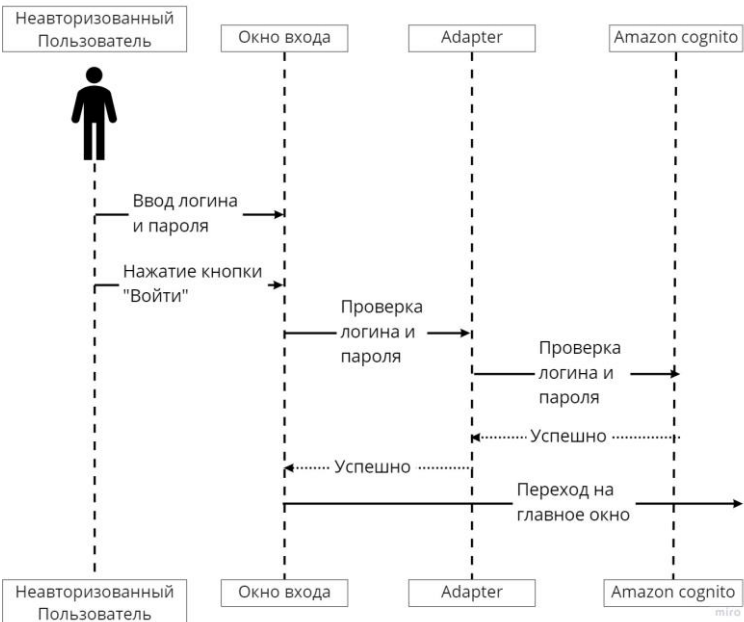


Рисунок 2 - Диаграмма последовательностей для неавторизованного пользователя в настольном приложении

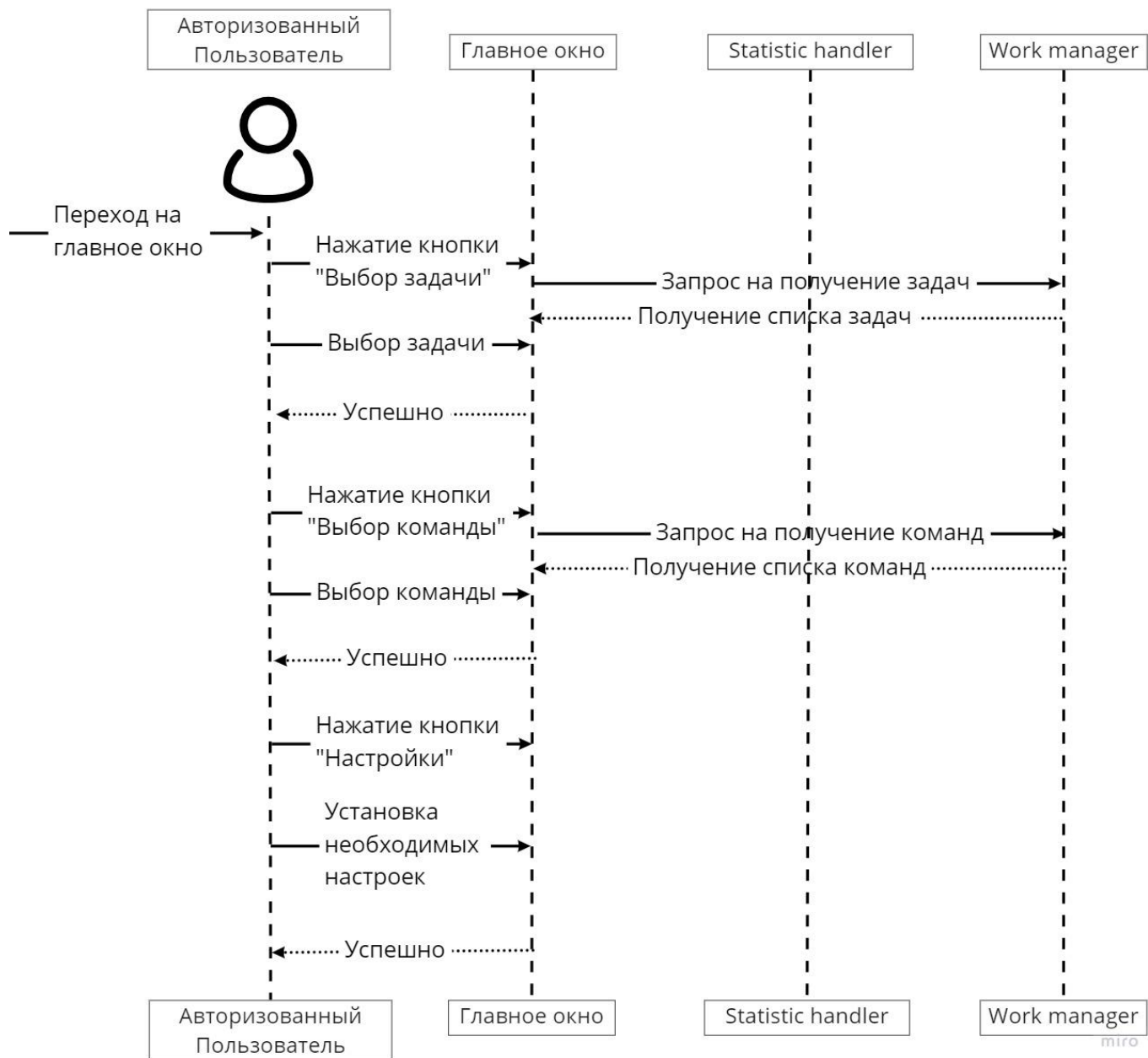


Рисунок 3 - Диаграмма последовательностей для авторизованного пользователя в настольном приложении (часть 1)

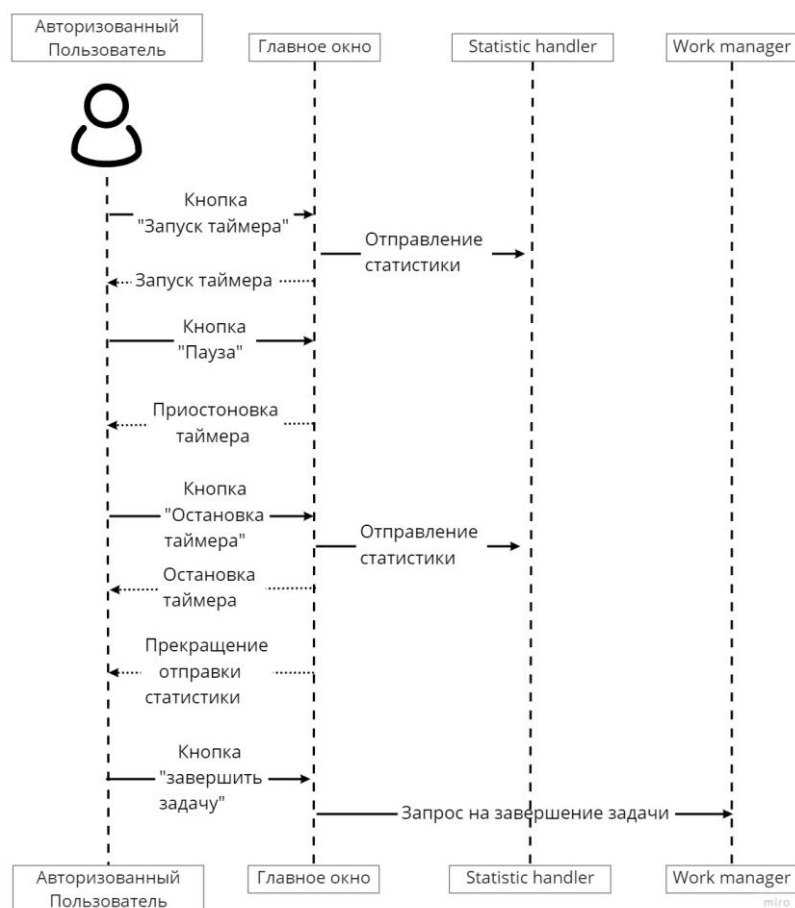


Рисунок 4 - Диаграмма последовательностей для неавторизованного пользователя в настольном приложении (часть 2)

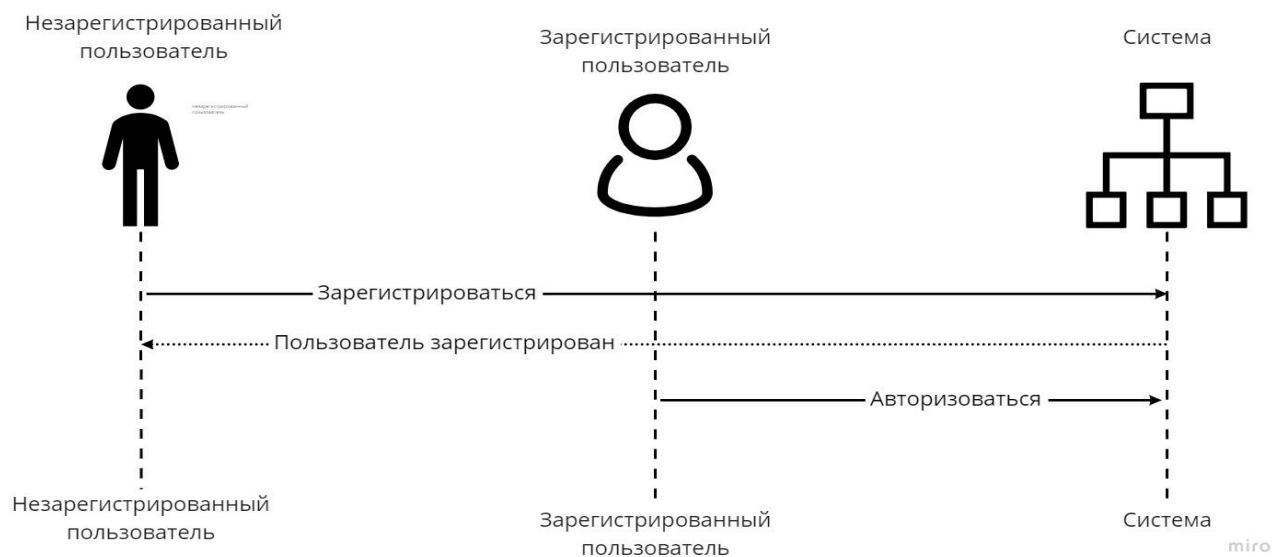


Рисунок 5 - Диаграмма последовательностей для неавторизованного пользователя в WEB приложении

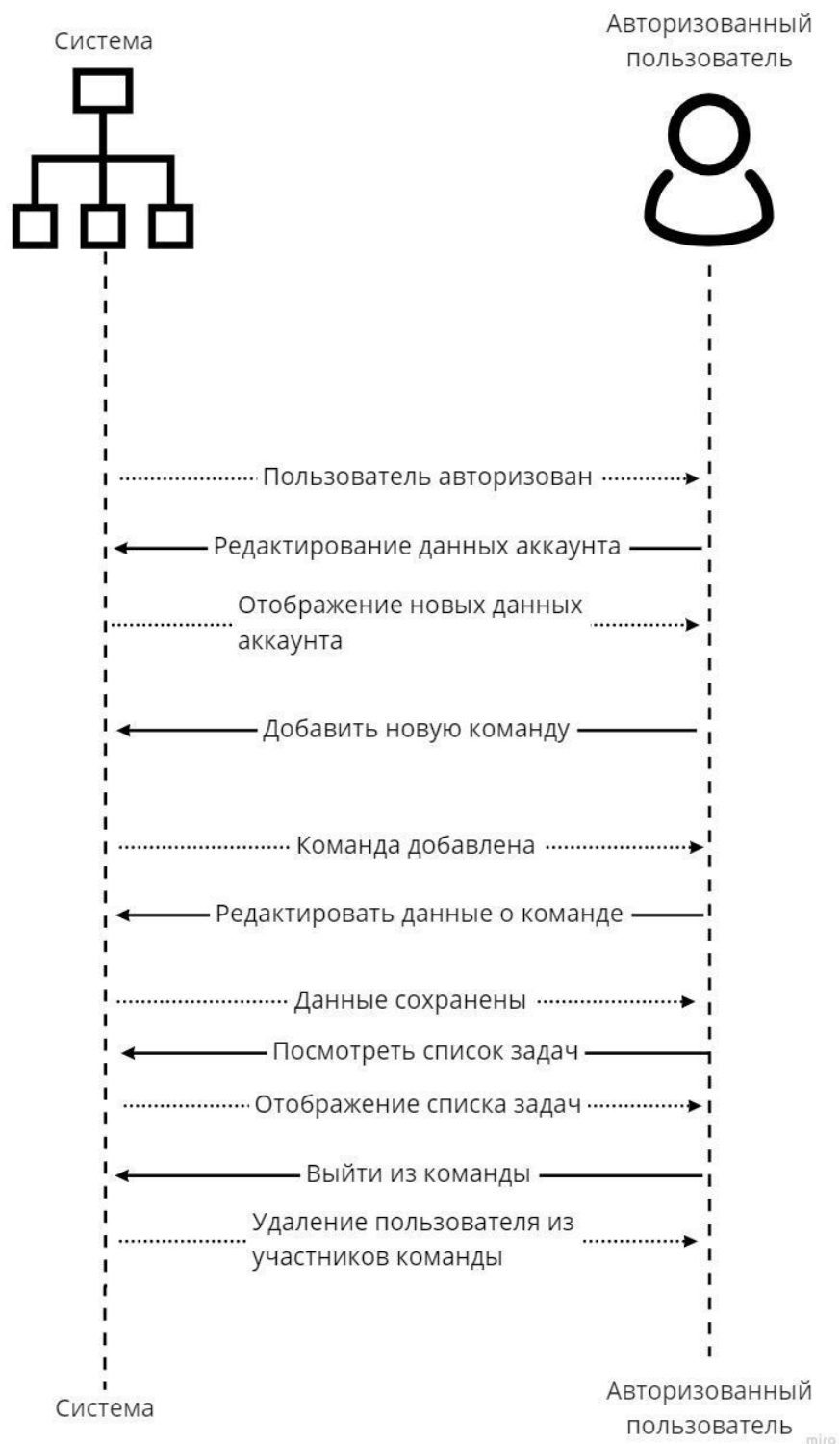


Рисунок 6 - Диаграмма последовательностей для авторизованного пользователя в WEB приложении



Рисунок 7 - Диаграмма последовательностей для лидера команды в WEB приложении

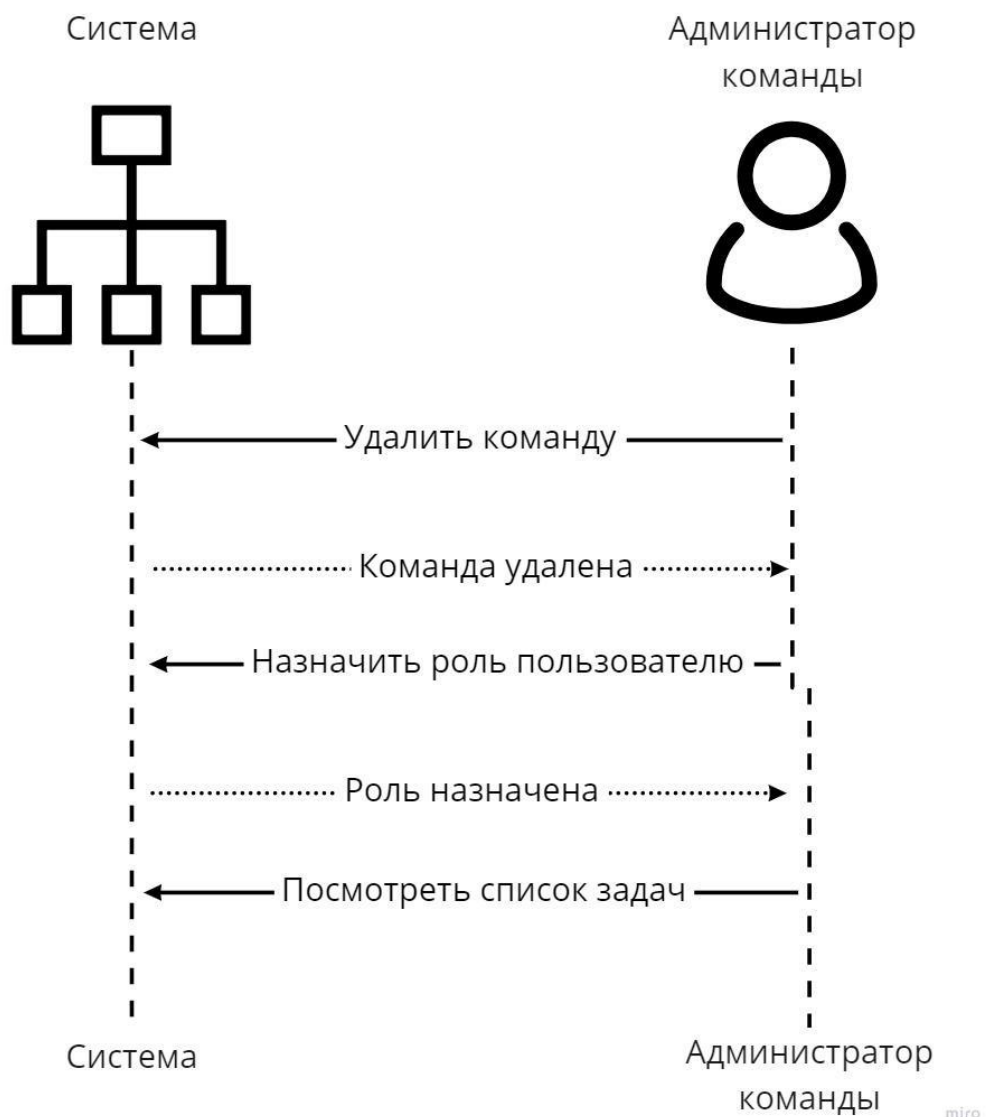


Рисунок 8 - Диаграмма последовательностей для администратора команды в WEB приложении

2.3.3. Варианты состояния системы

Диаграммы состояний определяет, как объект переходит из одного состояния в другое.

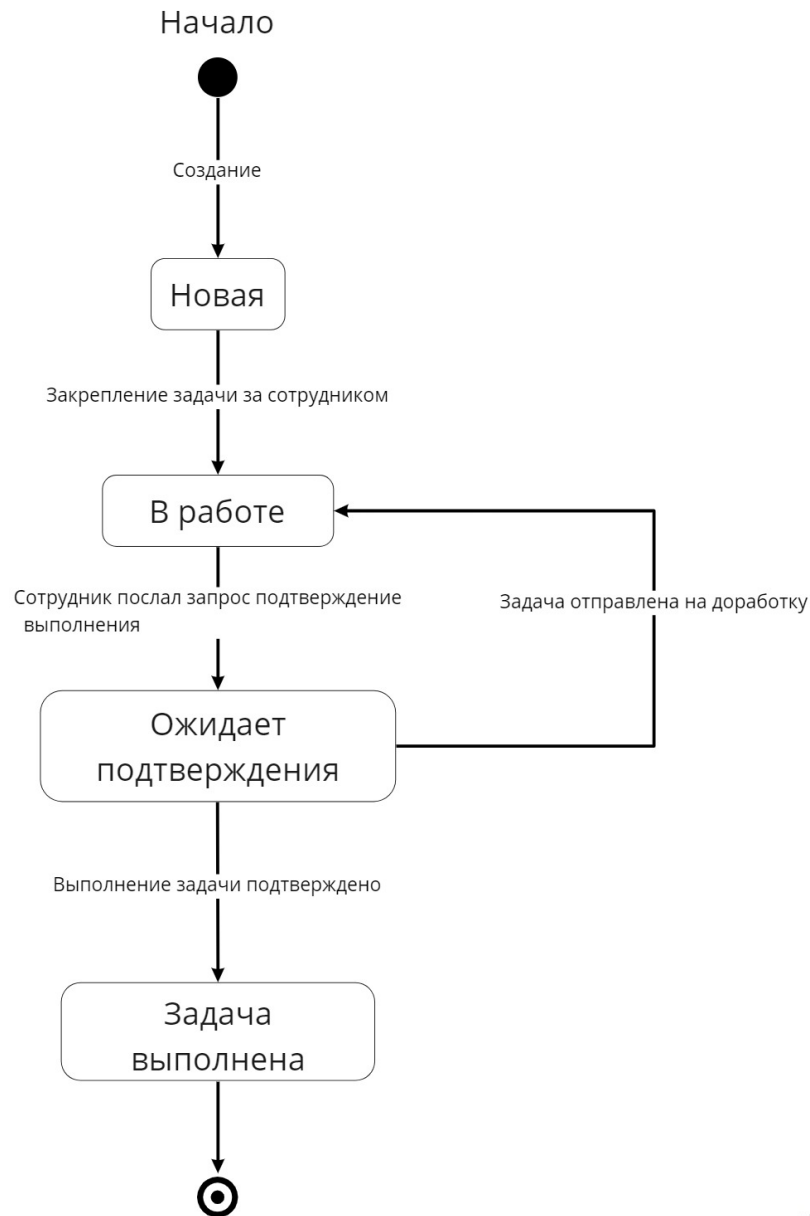


Рисунок 9 - Диаграмма состояний задачи

Диаграмма состояний, изображенная на Рисунке 9, отражает возможные состояния задачи. После создания задаче присваивается статус “новая”, после чего, данная задача закрепляется за сотрудником. После закрепления задачи за сотрудником, статус задачи меняется на “в работе”. После того, как работник завершил выполнение задачи, он делает запрос на подтверждение завершения задачи, в это время статус задачи меняется на “ожидает подтверждения”. В

случае, если задача требует доработки, ее статус меняется на “в работе”, и она снова отправляется на доработку сотруднику. В случае, если задача не требует доработки и может считаться выполненной, ей присваивается статус “выполнено”.

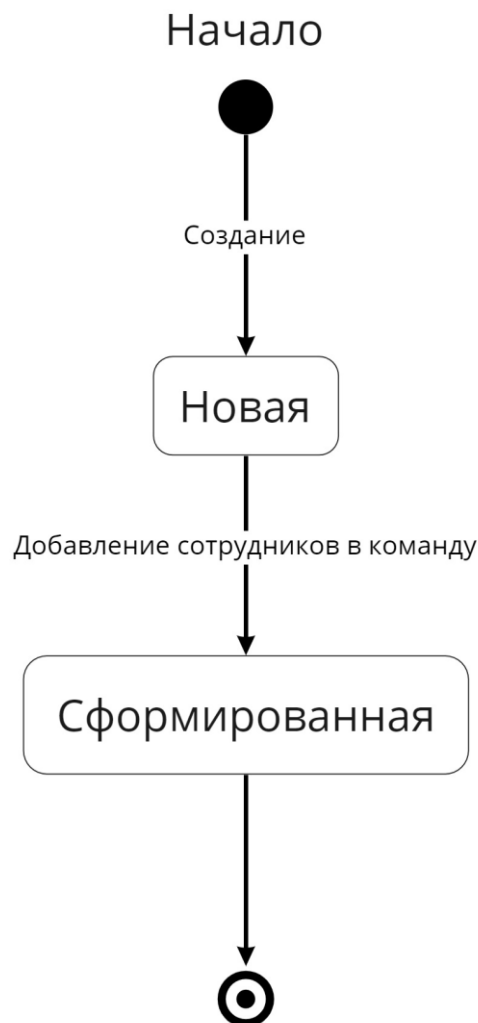


Рисунок 10 - Диаграмма состояний команды

Диаграмма состояний, изображенная на Рисунке 10, отражает возможные состояния команд. Изначально создается пустая команда, в которой присутствует только администратор. Затем, в команду добавляются сотрудники, назначается “лидер команды”.

2.3.4. Варианты действия в системе

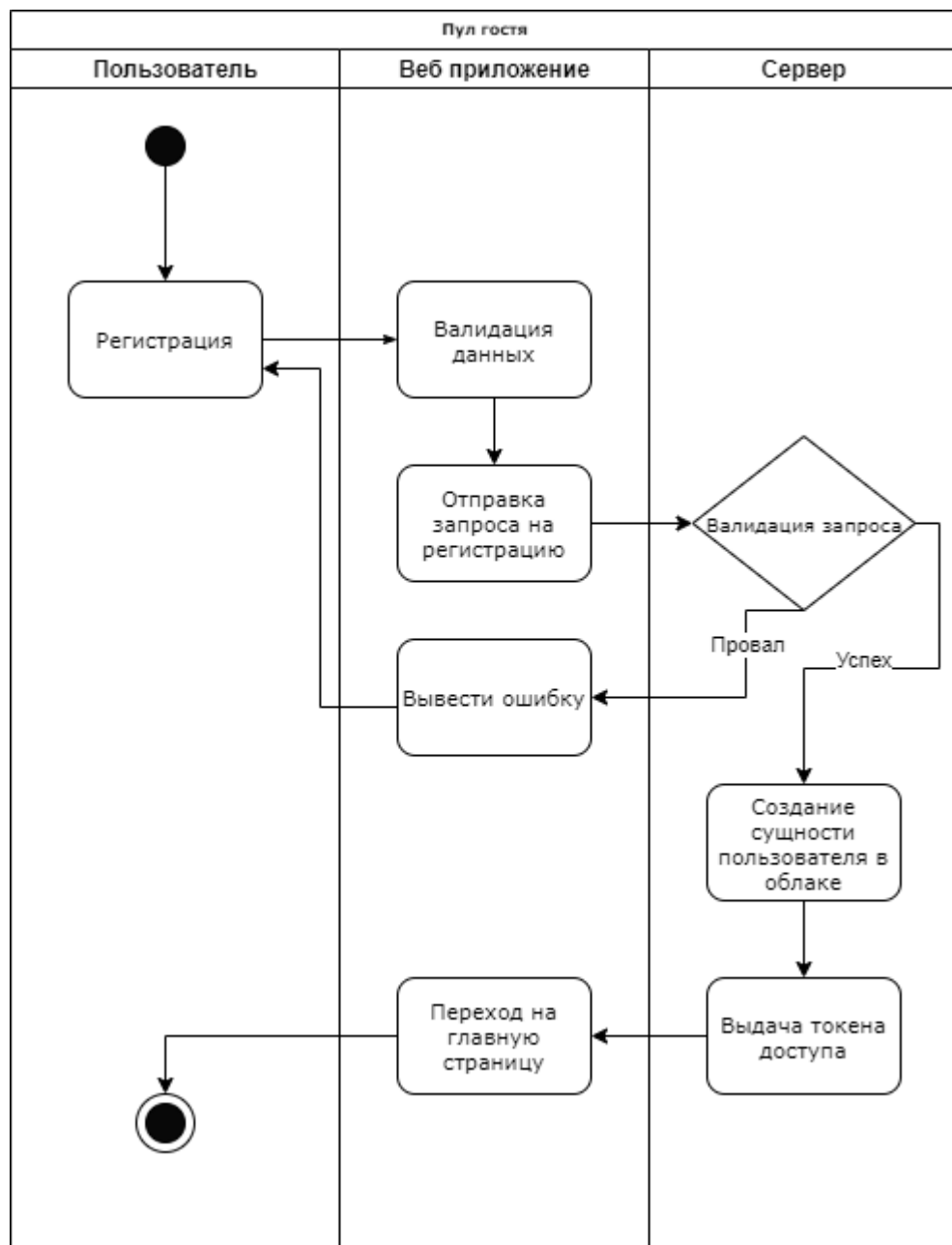


Рисунок 11 - Диаграмма активности гостя

Диаграмма активности, изображенная на Рисунке 11, отражает действия гостя при регистрации.

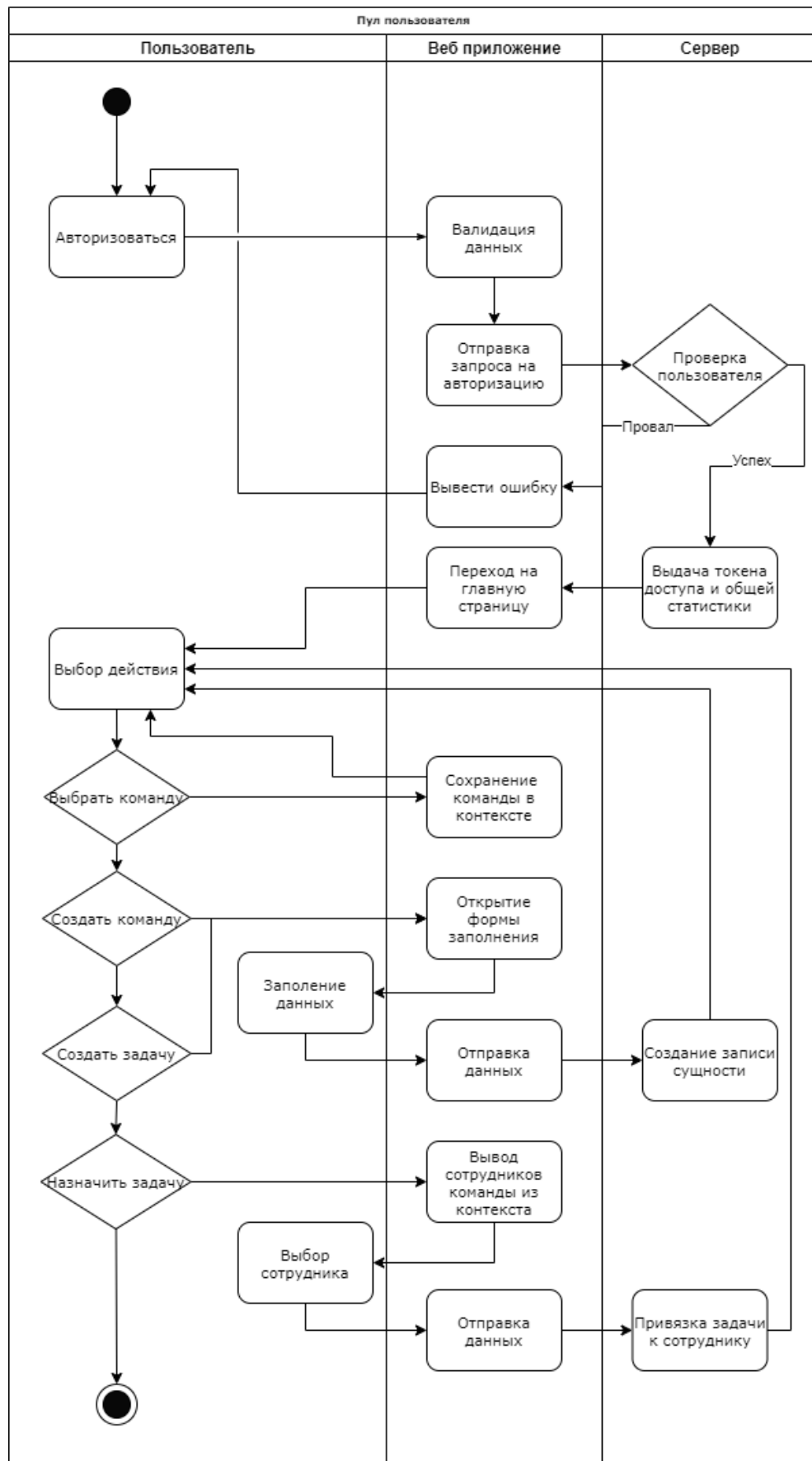


Рисунок 12 - Диаграмма активности пользователя

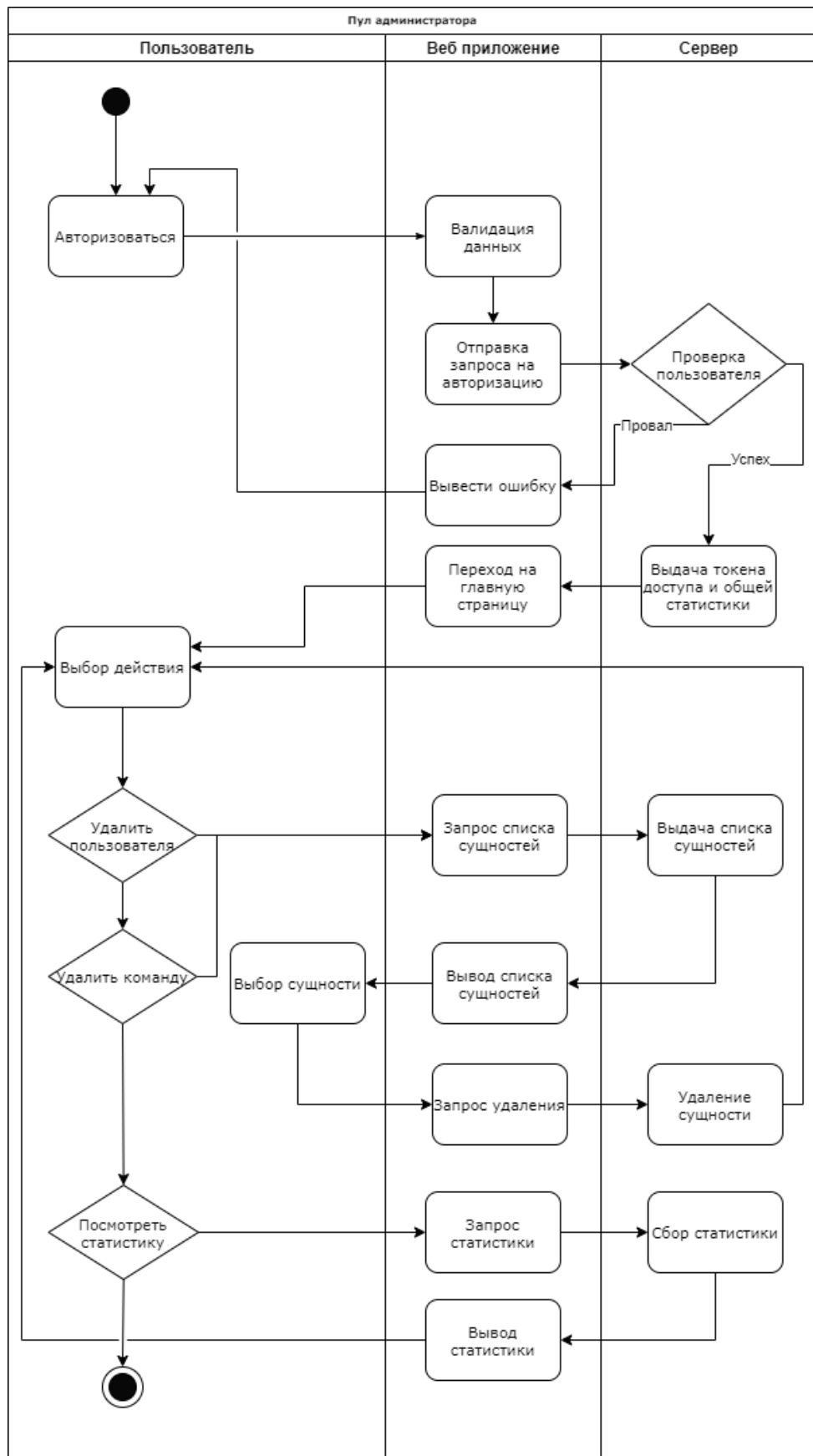


Рисунок 13 - Диаграмма активности администратора

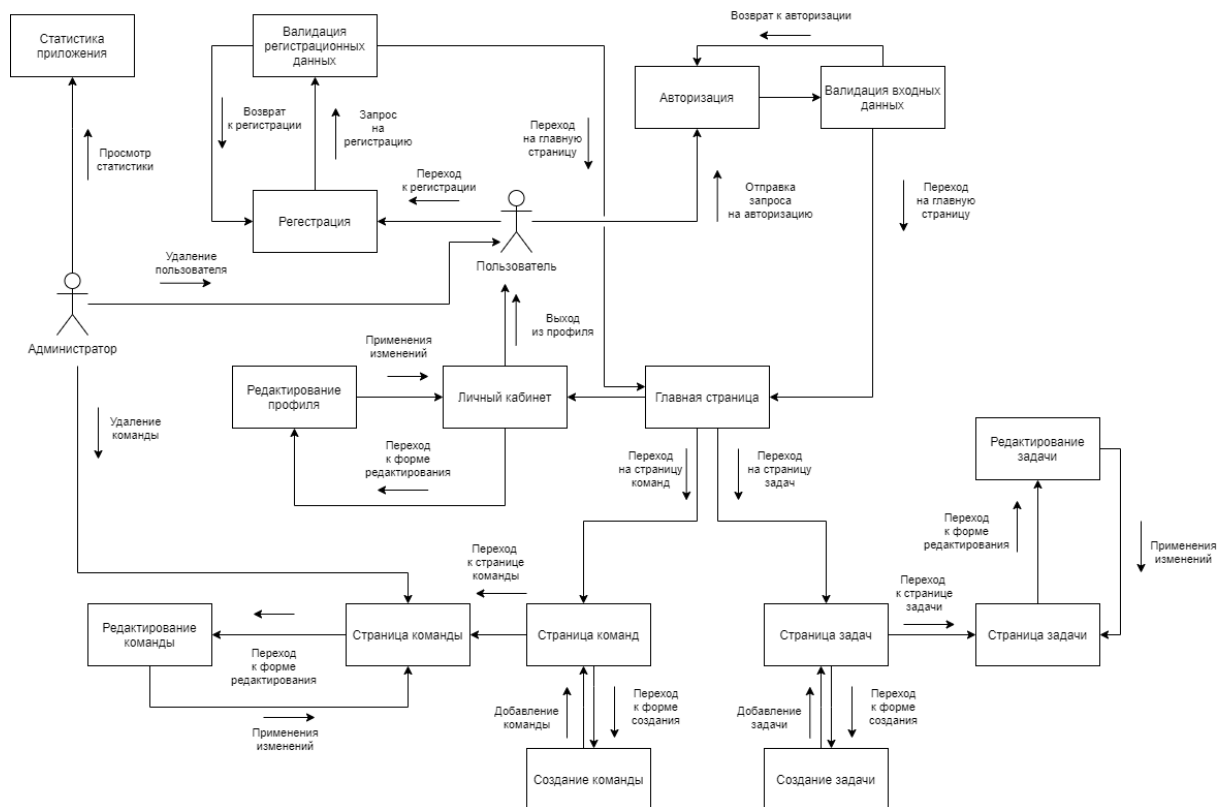


Рисунок 14 - Диаграмма взаимодействий

Диаграмма взаимодействия, изображенная на рисунке 14, определяет какие способы взаимодействия с приложением есть у пользователя.

2.3.5. Развертывание приложения

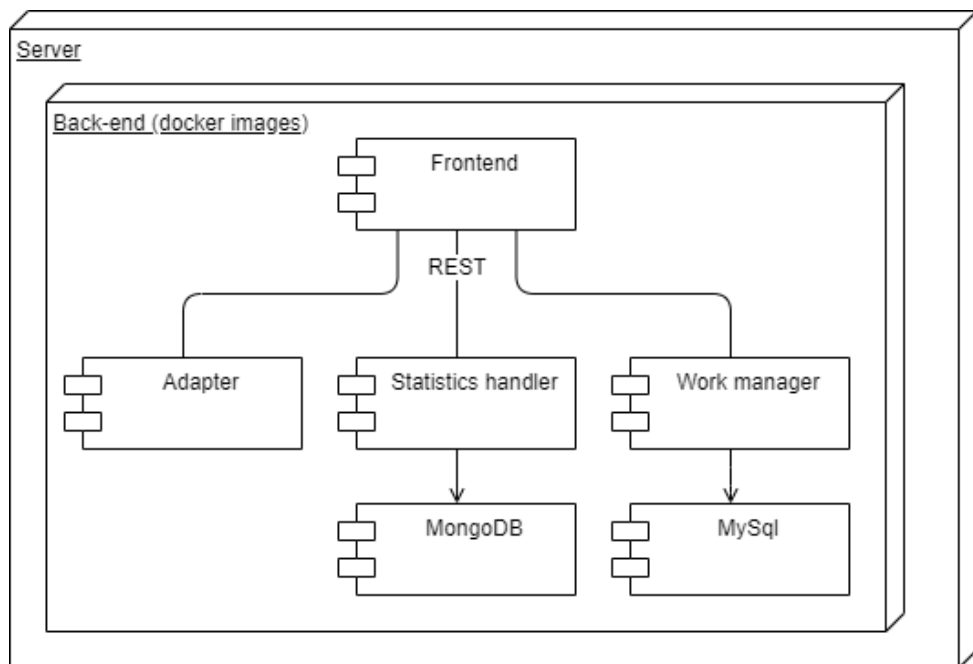


Рисунок 15 - Диаграмма развертывания

Данная диаграмма, изображенная на рисунке 15, определяет аппаратные компоненты, их взаимодействие друг с другом и функционирование.

3. Реализация

3.1. Анализ средств реализации

Веб-приложение функционирует в стандартной операционной серверной среде. Произведена интеграция с Amazon Cognito для регистрации и авторизации, аутентификации пользователей.

Для технической реализации серверной части приложения используются следующие средства:

1. СУБД (MySQL, MongoDB)
2. Spring Boot 2
3. Amazon Cognito

Для технической реализации настольного приложения используются следующие средства:

- JavaFX

Для реализации дизайна веб приложения используются следующие средства:

- HTML;
- CSS;
- Grommet;
- React;

Spring Boot - универсальный фреймворк с открытым исходным кодом для Java-платформы. Обладает большим функционалом, но его наиболее значимыми особенностями являются: управление зависимостями, автоматическая конфигурация и встроенные контейнеры сервлетов.

MongoDB - представитесь NoSQL баз данных, имеющий отличную документацию, который отлично подходит для хранения статистики выполнения задач работниками.

React - JavaScript-библиотека с открытым исходным кодом для разработки пользовательских интерфейсов, которая отлично подходит для разработки web-приложения.

MySQL - свободная реляционная система управления базами данных, которая подходит для начальной разработки приложения, предполагающей удаленный сервер.

3.2. Архитектура приложения

Пользователь взаимодействует с серверной составляющей приложения посредством клиентского приложения (web-браузер) и настольного приложения, установленного на его персональном компьютере.

Сервер, посредством REST API, возвращает пользователю доступную ему запрашиваемую информацию. Серверная составляющая приложения основывается на нескольких сервисах, которые с помощью технологии m2m могут общаться между собой, тем самым образуя микросервисную архитектуру, в которую входят:

- сервис адаптер для взаимодействия с Amazon Cognito (Adapter)
- сервис управления пользователями, командами и задачами (Work manager)
- сервис хранения прогресса задач (Statistics handler)

На рисунке 16 изображена диаграмма классов микросервиса Work manager.

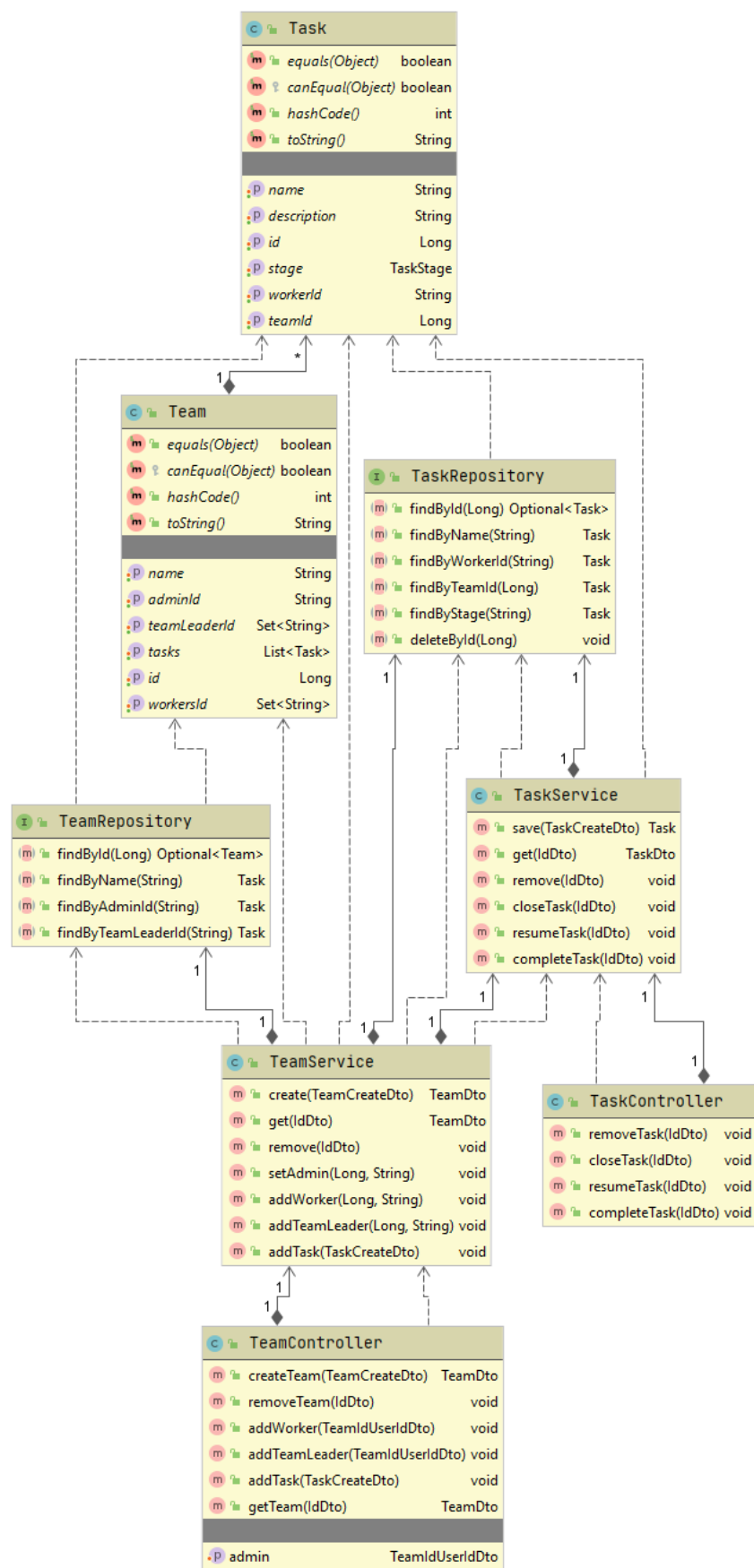


Рисунок 16 - Диаграмма классов Work Manager

На рисунке 17 изображена диаграмма классов микросервиса Adapter.

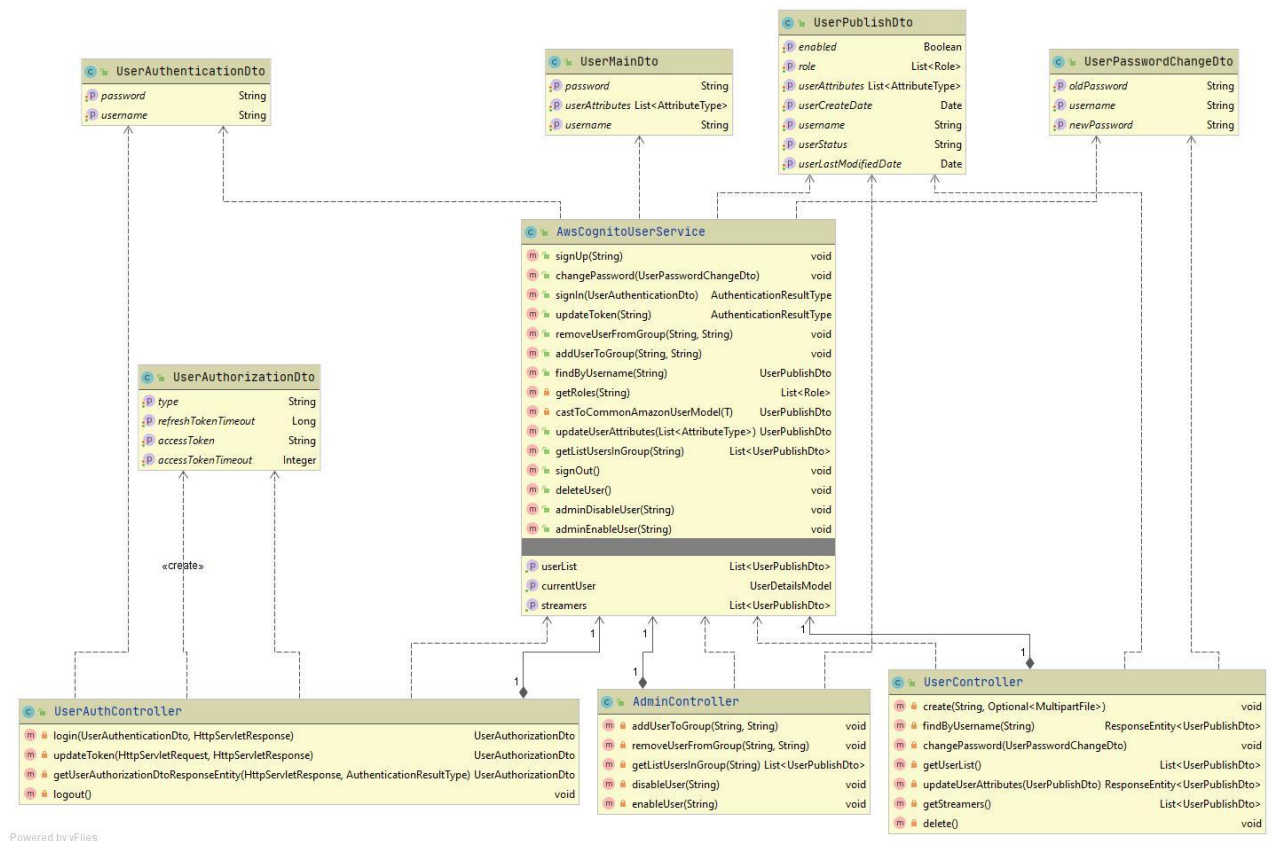


Рисунок 17 - Диаграмма классов Adapter

На рисунке 18 изображена диаграмма классов микросервиса Statistic Handler.

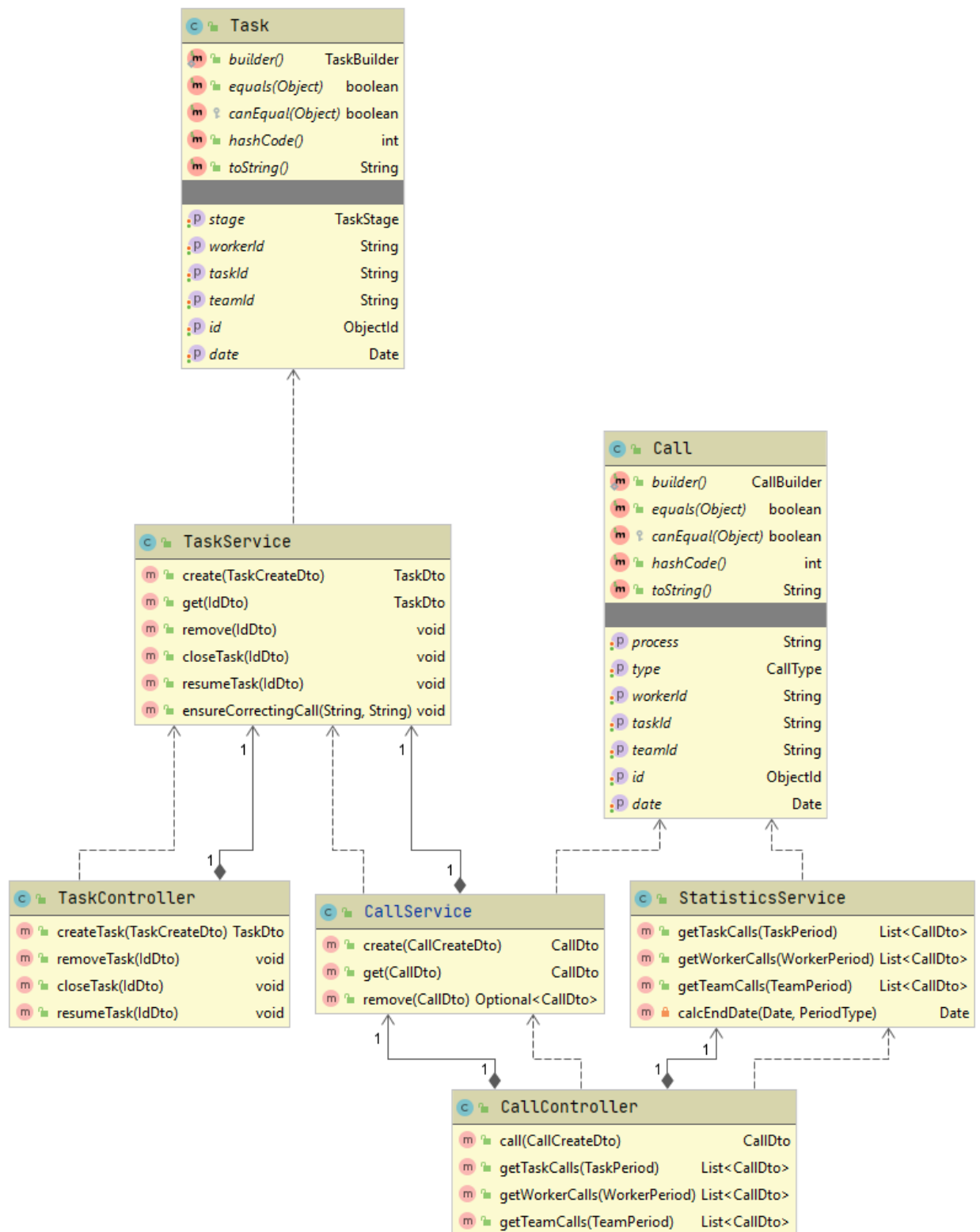


Рисунок 18 - Диаграмма классов Statistic Handler

В приложении используются две базы данных. NoSQL для хранения статистики прогресса задач, как самой нагруженной части приложения и SQL для хранения сущностей и построения связей между работниками, командами и задачами. Выбраны следующие представители этих баз данных:

- MySQL - хранение пользователей, команд и задач
- MongoDB - хранение прогресса задач

Общее представление архитектуры представлено на рисунке 19.

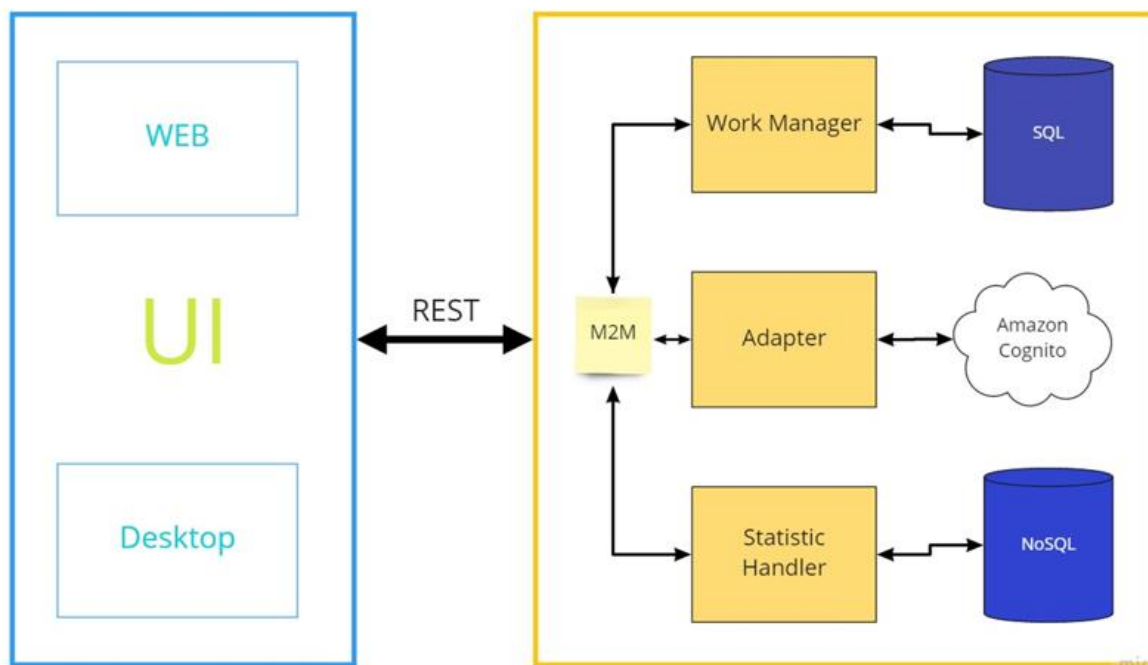


Рисунок 19 – Общее представление архитектуры

3.3. Реализация серверной части приложения

Серверная часть приложения реализована с помощью языка программирования Java, а также с использованием фреймворка Spring. Архитектура серверной части включает следующие компоненты:

- Контроллер;
- Сервис;
- Репозитории;
- Модели.

Для приёма запросов, а также их обработки используется компонент контроллер. Он также необходим для создания и передачи моделей. К каждому контроллеру прикреплен уникальный URL-адрес. Часть контроллеров, которые не предусматривают публичный доступ, защищены.

Для реализации бизнес-логики приложения используется компонент сервис.

Для описания сущностей и связей между сущностями используется компонент модель.

3.4. Реализация настольного приложения

Для создания графического интерфейса настольного приложения используется платформа JavaFX. Каждый экран описан в отдельном файле с помощью языка разметки FXML.

Для каждого экрана создан отдельный контроллер.

Передача запросов на сервер, а также их получение реализовано с помощью библиотеки OkHttp. Для безопасности, в заголовке каждого запроса, отправляемого на сервер, кроме запроса на авторизацию, посылает токен авторизации.

На экране авторизации вводятся необходимые данные (имя пользователя и пароль), после чего, отправляется запрос на микросервис Adapter. Если авторизация произошла успешно, в ответе сервера содержится токен авторизации, который сохраняется в приложении на время действия токена. В отдельном потоке происходит проверка действительности токена, и в случае если это необходимо, при помощи запроса на сервер, токен обновляется. После успешной авторизации происходит переход на главный экран.

В контроллере главного экрана реализован таймер в отдельном потоке, управление которым производится с помощью кнопок на графическом интерфейсе. Периодически, через определенный интервал времени посылается запрос на микросервис Statistic Handler, который содержит в себе информацию о задаче, выполняемой пользователем в данный момент.

С помощью сохраненного при авторизации токена, а также имени пользователя запрашиваются данные о командах и задачах для конкретного пользователя. Для получения данных о командах и задачах посылаются запросы на микросервис Work manager.

3.5. Реализация WEB-приложения

3.6. Сущности

Ниже, на рисунке 20, изображена диаграмму объектов, отображающая используемые в проекте сущности. А именно:

- User – пользователь приложения
- Team – команда, в которую объединяются пользователи для выполнения задач.
- Task – задача, которую должна выполнить команда, а также пользователь в частности.
- Call – запрос, периодически посылаемый настольным приложением на сервер, в процессе выполнения задачи пользователем.

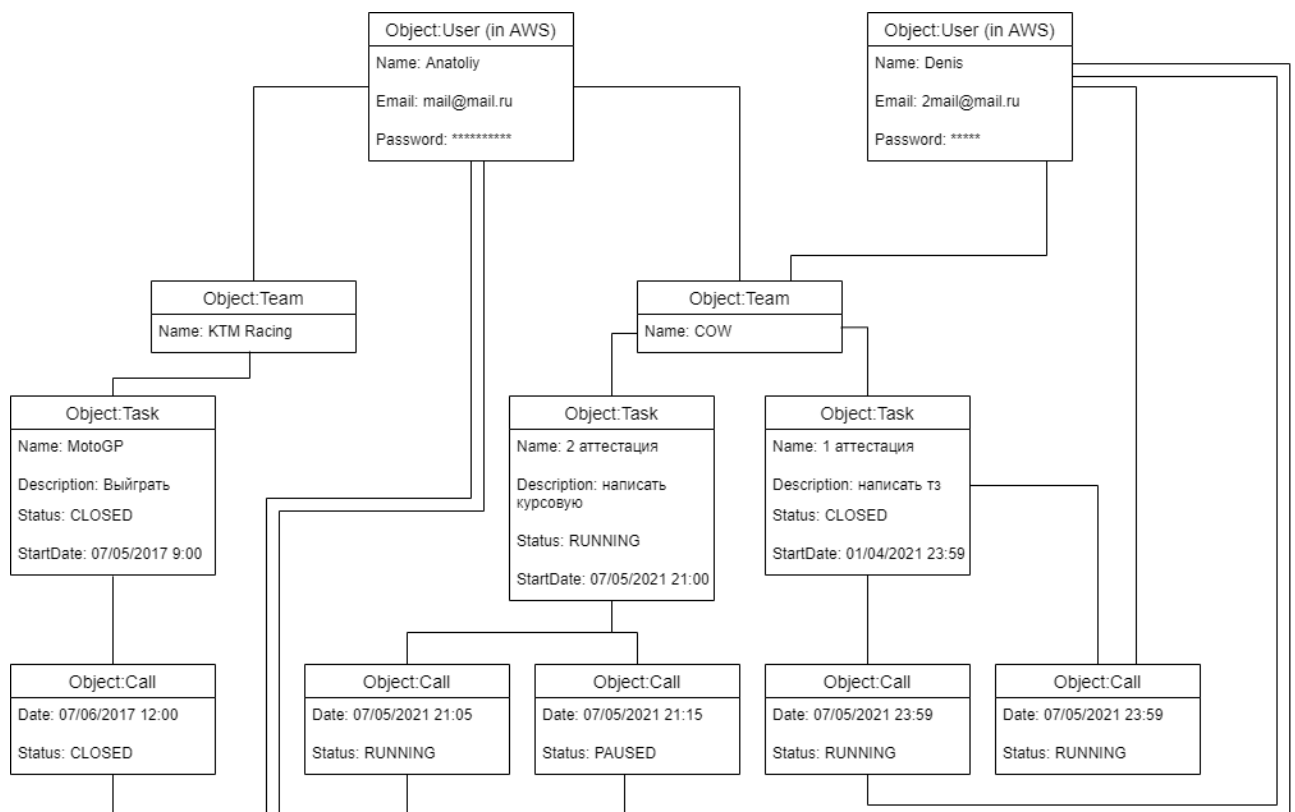


Рисунок 20 - Диаграмма объектов

4. Описание пользовательского интерфейса

4.1. Интерфейс WEB-приложения

Оформление веб-приложения выполнено преимущественно в светлых тонах, с использованием следующей цветовой палитры:

- темно бирюзовый цвет #4FAEB4 - используется для фона верхнего навигационное меню;
- черный цвет #3E3E3E - используется для фона дополнительного навигационного меню;
- серый цвет #E5E5E5 - используется для фона тела сайта и для выделения действий отмены;
- бирюзовый цвет #09CF83 - используется для выделения действий;
- зеленая кнопка #11D600 - используется для выделения действий характера “создания”, “принятия”;
- красная кнопка #F30000 - используется для выделения опасных действий, таких как “удаление”, “выгнать”;
- желтая кнопка #FFDA18 - используется для выделения действий редактирование, обновления;
- темно бирюзовый цвет #4FAEB4 - используется для выделения действий выбора;
- темно фиолетовый цвет #6F4BFF - используется для выделения действий характера “посмотреть”, “подробнее”.

4.1.1 Главная страница

Главная страница приложения открывается по умолчанию если пользователь авторизован при открытии сайта, если не выбрана какая-либо другая. Страница содержит общее навигационное меню, а также в теле страницы:

- Список команд с общей информацией, в которых пользователь включен;
- Список задач с общей информацией, которые присвоены пользователю.

4.1.2 Страница профиля

Страница профиля пользователя для редактирования информации о себе. Страница содержит общее навигационное меню, а также в теле страницы:

- кнопка выход - выход из приложения;
- аватар - при нажатии можно сменить аватар;
- имя - публичное имя пользователя, при нажатии можно изменить;
- статус - текущий статус пользователя.

4.1.3 Страница команд

Страница списка всех команд, к которым принадлежит авторизованный пользователь, доступная через соответствующую кнопку в навигационном меню. Страница содержит общее навигационное меню, а также в теле страницы:

- список команд:
 - общая информация о команде;
 - действия:
 - участник команды:
 - выбрать - выбрать команду в качестве контекста для работы приложения;
 - выйти - выйти из команды
 - лидер команды:
 - пригласить - добавить пользователя к команде;
 - редактировать - редактировать команду.
 - администратор команды:
 - удалить - удалить команду.
- кнопка создание команды - переход к форме созданию команды.

4.1.4 Страница команды

Страница команды содержит основную информацию о команде и доступна через действия из страницы команд для авторизованного пользователя который принадлежит данной команде. Страница содержит общее навигационное меню, а также в теле страницы:

- список участников:
 - общая информация об участнике
 - действия, если администратор команды:

- сделать администратором команды;
- сделать лидером команды;
- сделать участником команды;
- выгнать.
- удалить - удалить команду;
- пригласить - пригласить пользователя в команду;
- задачи - переход к списку задач команды;
- интеграции - выбрать интеграцию из предоставленных.

4.1.5 Страница задач

Страница задач команды доступна через соответствующую кнопку в навигационном меню авторизованному пользователю, который является лидером данной команды или выше. Страница содержит общее навигационное меню, а также в теле страницы:

- список задач:
 - общая информация;
 - действия для каждой задачи:
 - закончить - закончить задачу
 - выбрать - выбрать участника, который будет выполнять данную задачу;
 - посмотреть - перейти на страницу задачи;
 - редактировать - редактировать задачу;
 - удалить - удалить задачу.
- создание задач - переход к форме создания задачи.

4.1.6 Страница задачи

Страница задачи отображает историю ее выполнения доступная со страницы задач для авторизованного пользователя который является лидером команды и выше. Страница содержит общее навигационное меню, а также в теле страницы:

- выбрать - выбрать участника, который будет выполнять данную задачу;
- изменить - отредактировать задачу;
- удалить - удалить задачу;
- закончить - закончить задачу;
- описание;

- информация выполнения;
- список участников которые выполняли эту задачу:
 - аватар участника;
 - имя участника;
 - основная информация;
 - детали - переход на страницу детали работы участника команды над задачей.

4.1.7 Страница статистики

Страница статистики отображает информацию об выполнении задачи одним участником доступная из страницы задачи для авторизованного пользователя который является лидером команды и выше. Страница содержит общее навигационное меню, а также в теле страницы:

- информация об участнике;
- детали выполнения:
 - время выполнения;
 - активные приложения.

4.2. Интерфейс настольного приложения

Дизайн настольного приложения выполняется в темных тонах, с редким использованием ярких цветов для выделения и подчеркивания требующих этого элементов.

Цветовая палитра

- черный цвет - #000000 - используется как основной фоновый цвет и как цвет текста на светлом фоне;
- белый цвет - #FFFFFF - используется как фоновый цвет и как цвет выделения некоторых элементов;
- серый оттенок - #D6D6D6 - используется для фона вторичных элементов;
- зеленый цвет - #33FF00 - используется для кнопок, начинающих/одобряющих какие-либо действия;
- желтый цвет - #DBFF00- используется для кнопок, приостановки каких-либо действий;
- красный цвет - #FF0505 - используется для кнопок, завершающих/отклоняющих какие-либо действия;

- темный серый цвет - #2F2C2C - используется для фона вторичных элементов;
- синий цвет - #0500FF - используется для выделения некоторых компонентов при нажатии.

4.2.1 Экран входа

Экран входа, на который попадает пользователь, если он не авторизован. Данный экран необходим для авторизации пользователя. В теле содержатся:

- форма авторизации;
- кнопка входа.

Данный экран изображен на рисунке 21.

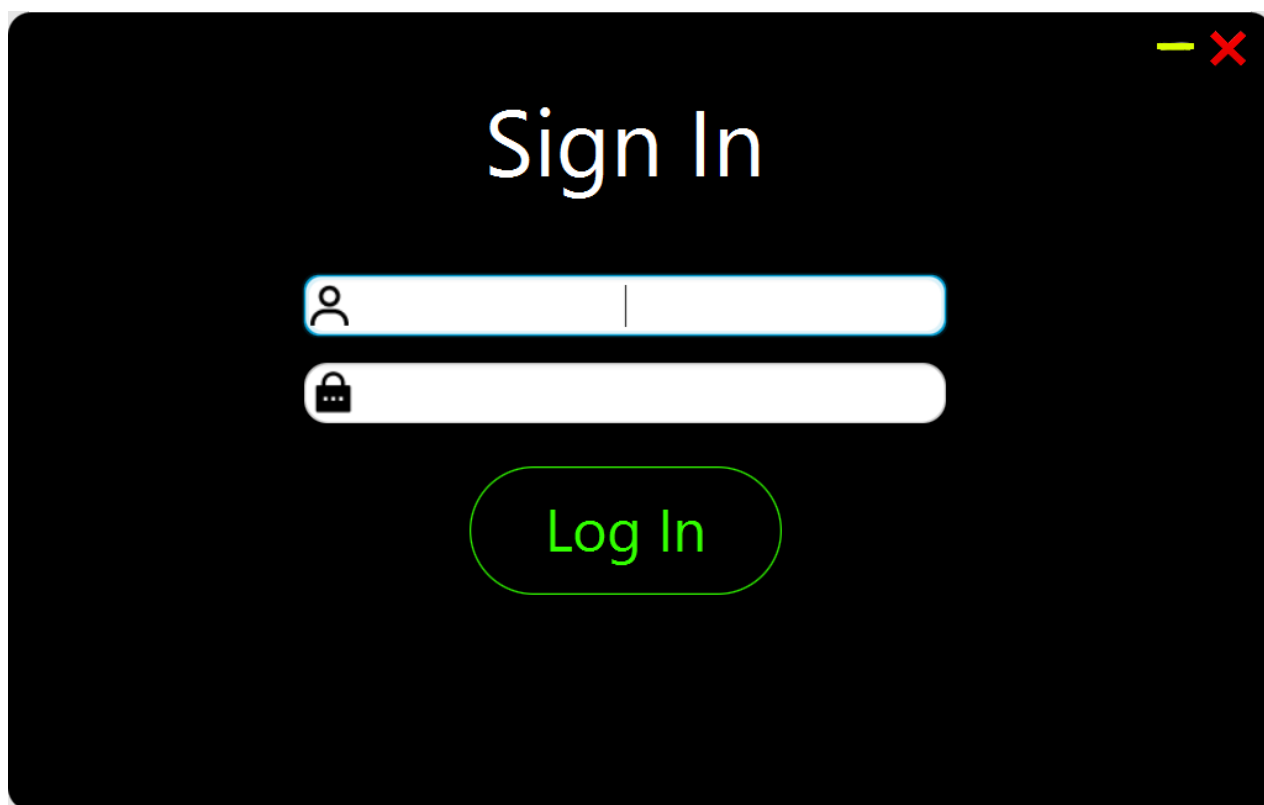


Рисунок 21 – Экран входа настольного приложения

4.2.2 Главный экран

Пользователь попадает на главный экран после авторизации, изображенный на рисунке 22. На главном экране располагаются основные элементы управления, такие как:

- кнопка выбора задачи;
- кнопка выбора команды;
- кнопка отправки готовности задачи;
- кнопка старта таймера;
- кнопка завершения задачи;
- кнопка паузы таймера;
- кнопка остановки таймера;
- кнопка перехода к настройкам приложения;
- кнопка выхода из аккаунта;
- таймер;
- строка, показывающая выбранную задачу;
- блок с информацией о пользователе (аватар, имя, фамилия).

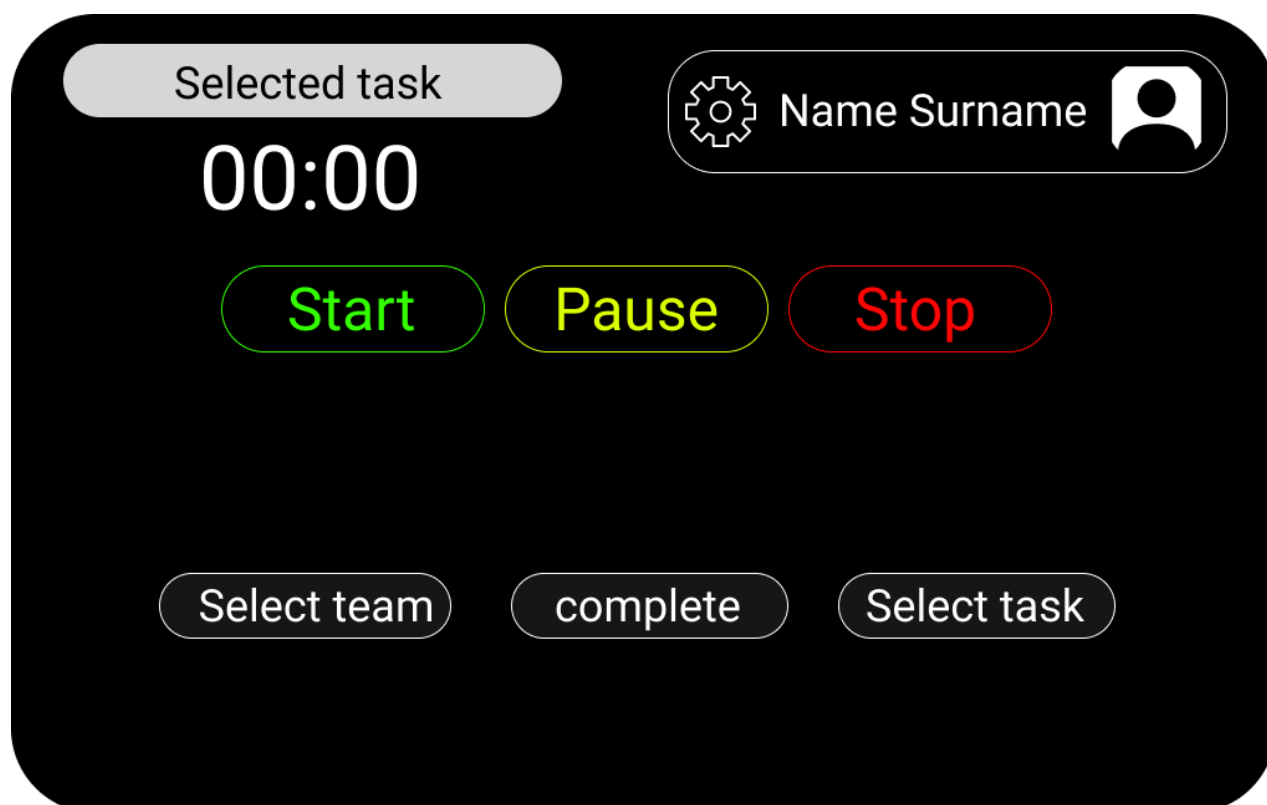


Рисунок 22 – Главный экран настольного приложения

4.2.3 Экран выбора задачи

Пользователь попадает на экран выбора задачи, изображенный на рисунке 23 после нажатия на кнопку выбора задачи. На экране выбора задачи располагаются такие элементы управления, как:

- список задачи, с возможностью выбора (выделения) конкретной задачи;
- кнопка “ok” для сохранения выбранной задачи и перехода на главный экран;
- кнопка перехода к настройкам приложения;
- кнопка выхода из аккаунта;
- таймер;
- строка, показывающая выбранную задачу;
- блок с информацией о пользователе (аватар, имя, фамилия).

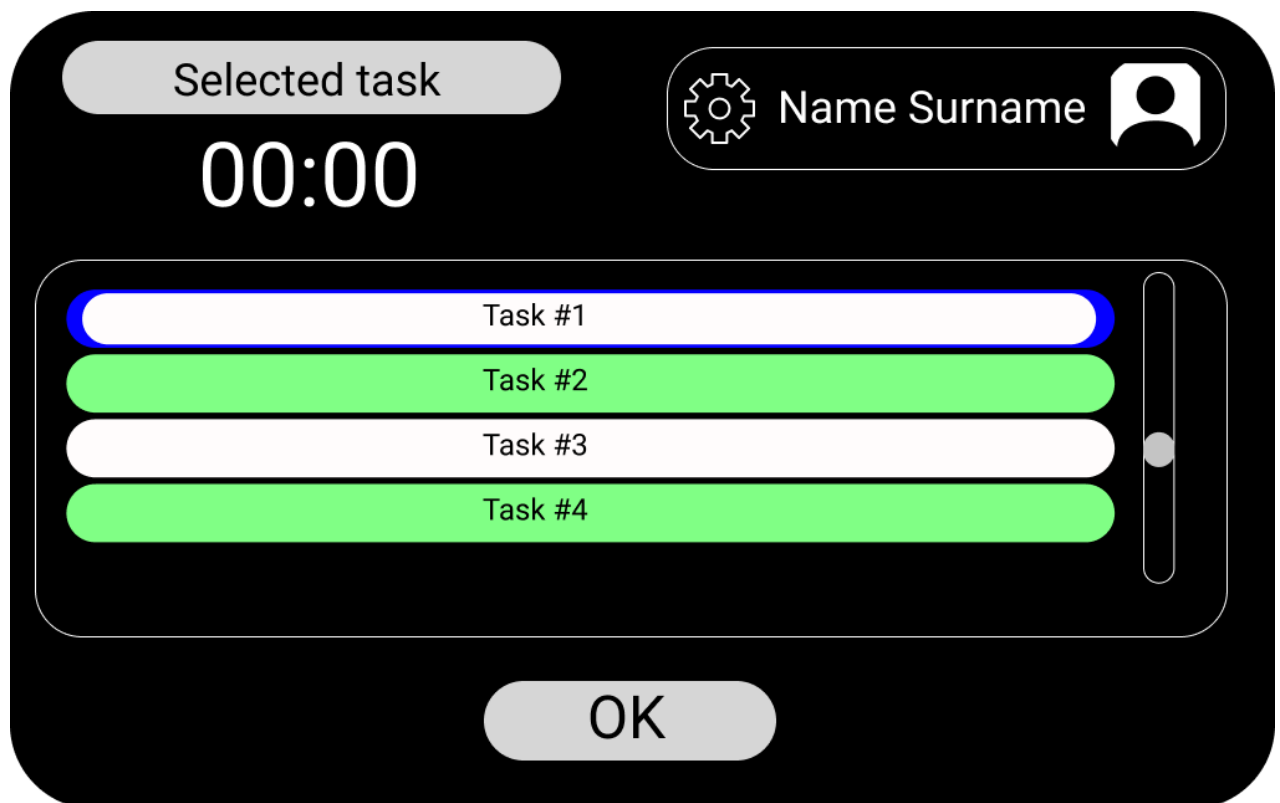


Рисунок 23 – экран выбора задачи настольного приложения

4.2.4 Экран выбора команды

Пользователь попадает на экран выбора команды, изображенный на рисунке 24 после нажатия на кнопку выбора команды. На экране выбора команды располагаются такие элементы управления, как:

- список команд, с возможностью выбора (выделения) конкретной команды;

- кнопка “ok” для сохранения принадлежности пользователя к выбранной команде и перехода на главный экран;
- кнопка перехода к настройкам приложения;
- кнопка выхода из аккаунта;
- таймер;
- строка, показывающая выбранную задачу;
- блок с информацией о пользователе (аватар, имя, фамилия).

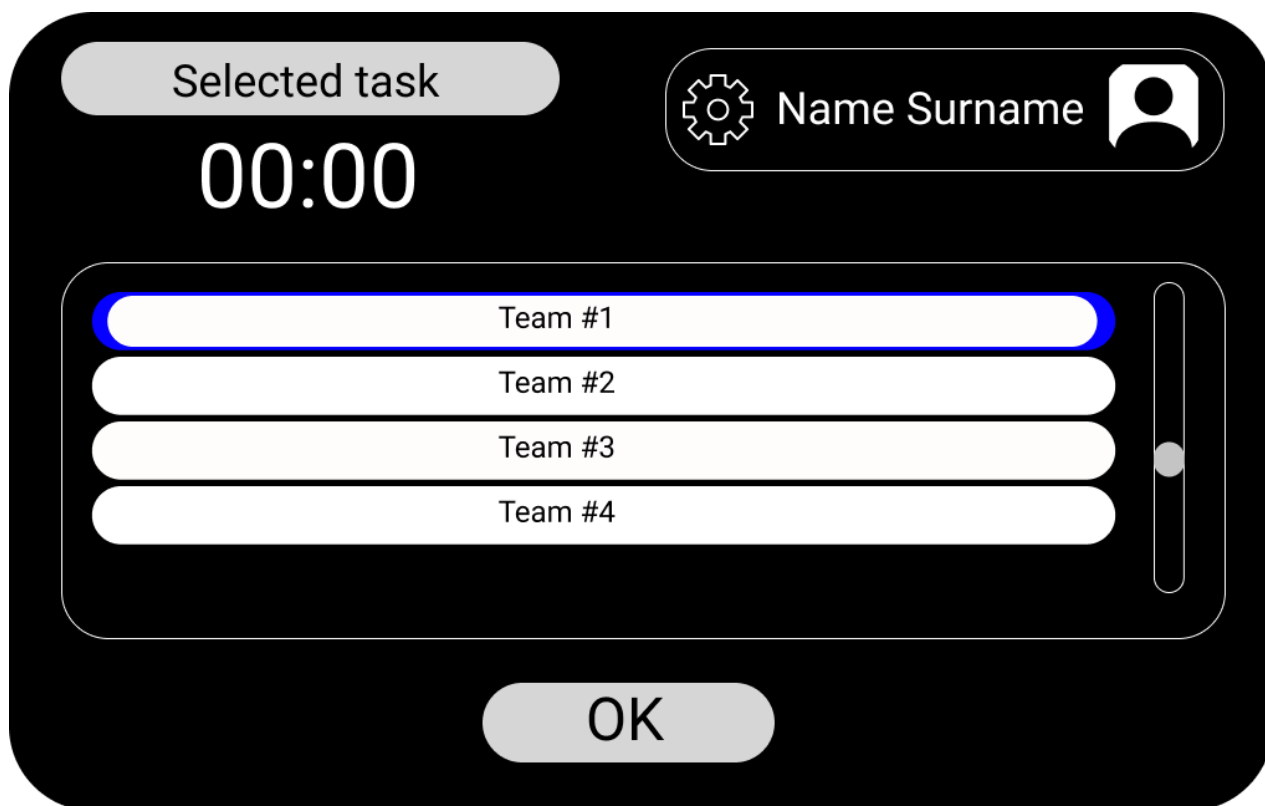


Рисунок 24 – экран выбора команды настольного приложения

4.2.5 Экран настроек

Пользователь попадает на экран настроек команды, изображенный на рисунке 25, после нажатия на кнопку выбора команды. На экране выбора команды располагаются такие элементы управления, как:

- список настроек приложения;
 - кнопка перехода к настройкам приложения;
 - кнопка выхода из аккаунта;
 - таймер;
 - строка, показывающая выбранную задачу;
- блок с информацией о пользователе (аватар, имя, фамилия).

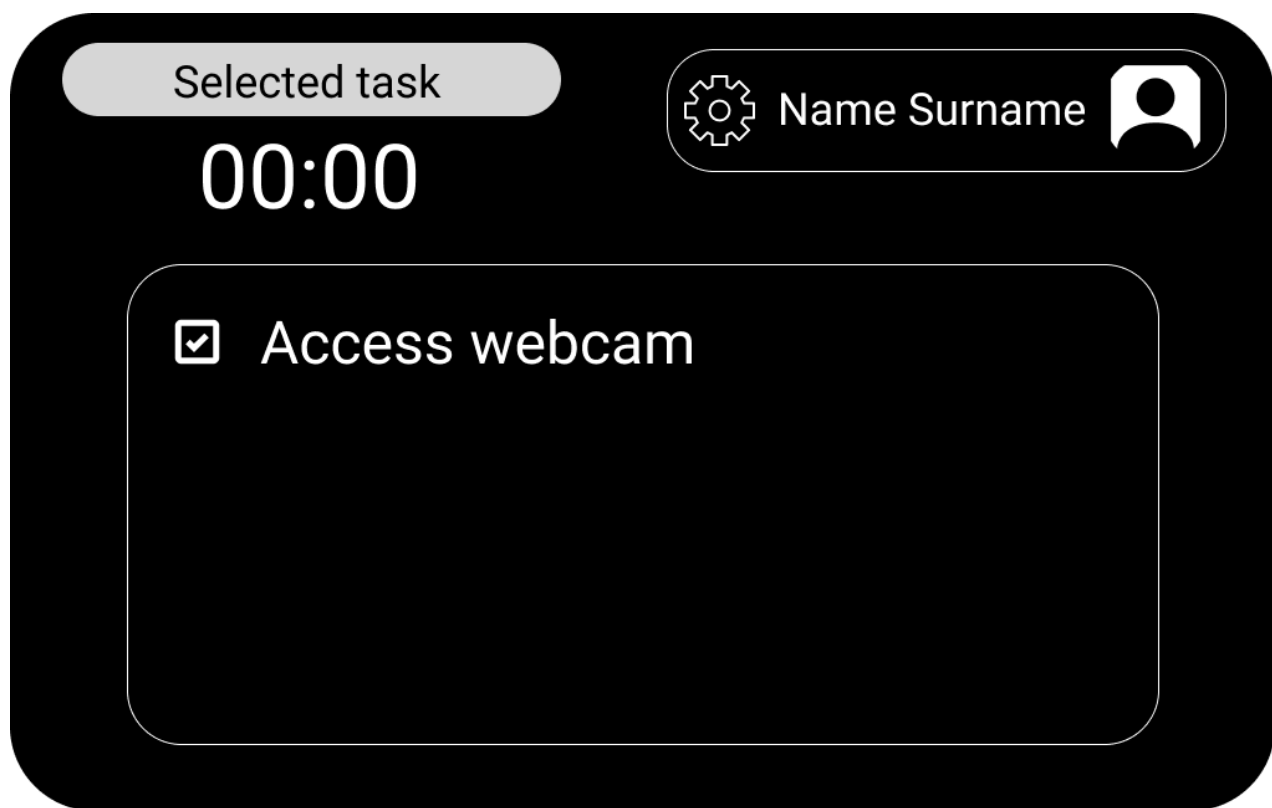


Рисунок 25 - экран настроек настольного приложения

Заключение

В результате работы было разработано клиент-серверное приложение, позволяющее вести учет времени, потраченного сотрудниками на выполнение задач, а также, позволяющее управлять задачами:

- 1) Разработана Front-end часть сайта, находящаяся на компьютере/телефоне пользователя.
- 2) Разработана Back-end часть сайта, развернутая на удаленном сервере.
- 3) Была создана связь между Front-end и Back-end частями приложения.
- 4) Разработана база данных, для хранения статистики, расположенная на удаленном сервере.
- 5) Разработана база данных для хранения сущностей задачи/команды.
- 6) Разработано настольное приложение.

Приложение отвечает всем заявленным требованиям.

Список использованных источников

1. Spring Framework Documentation - URL: <https://docs.spring.io/spring-framework/docs/current/reference/html/> (дата обращения: 30.03.2021 – 18.06.2021)
2. MySql Documentation - URL: <https://dev.mysql.com/doc/> (дата обращения: 30.03.2021 – 18.06.2021)
3. MongoDB Documentation - URL: <https://docs.mongodb.com/> (дата обращения: 30.03.2021 – 18.06.2021)
4. JavaFX Documentation - URL: <https://openjfx.io/javadoc/16/>
<https://openjfx.io/openjfx-docs/> (дата обращения: 30.03.2021 – 18.06.2021)
5. Trello API Documentation - URL: <https://developer.atlassian.com/cloud/trello/rest/> (дата обращения: 30.03.2021 – 18.06.2021)
6. Amazon Cognito Documentation - URL: <https://docs.aws.amazon.com/cognito/> (дата обращения: 30.03.2021 – 18.06.2021)