

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

Факультет компьютерных наук

Кафедра программирования и информационных систем

Контроль деятельности сотрудников в условиях удаленного формата работы.

Курсовая работа

09.03.02 Информационные системы и технологии

Обучающийся _____ А.А. Ударцев, Д.Э. Лисаев, 3 курс

Руководитель _____ В.С.Тарасов, преподаватель

Воронеж 2021

Содержание

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 непросто 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 нацелено помочь компаниям в поднятом ранее вопросе. Основной задачей данного приложения будет являться возможность предоставить владельцам компаний возможность 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 работой 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

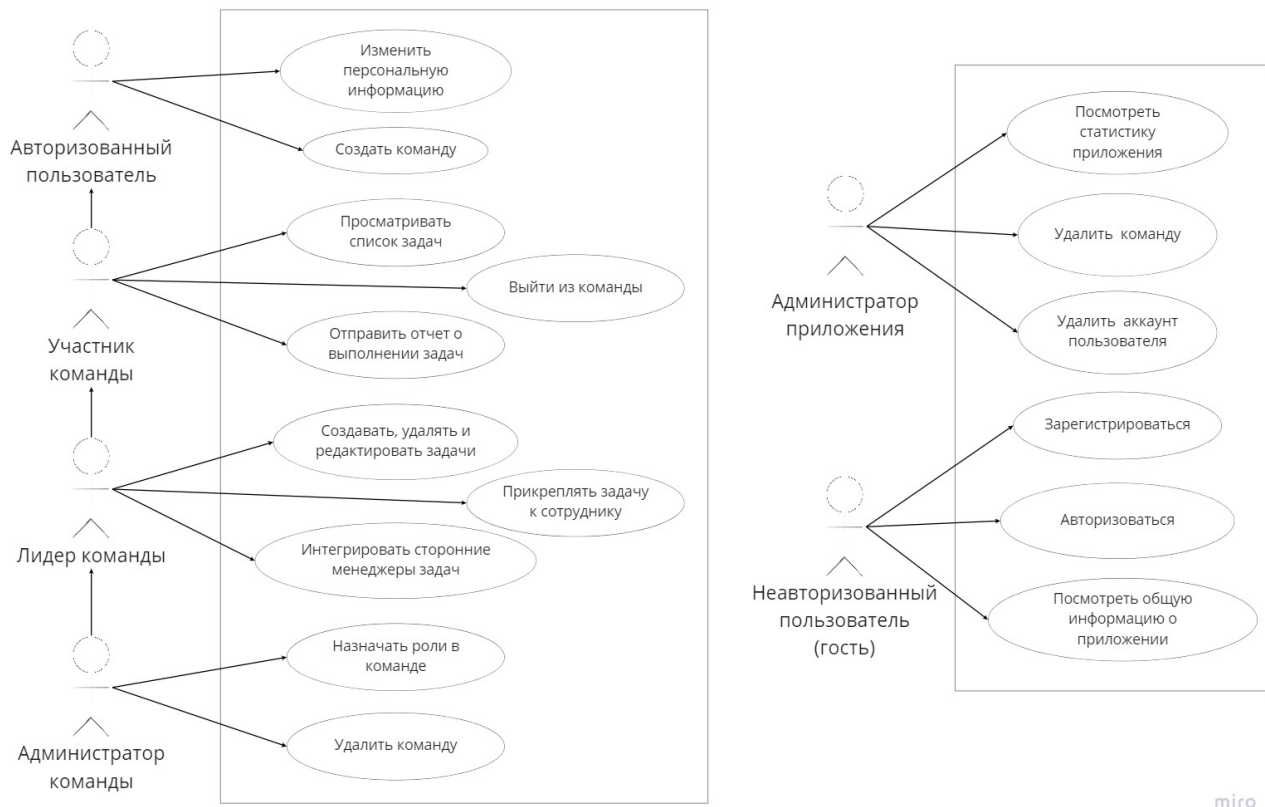


Рисунок 1 - Диаграмма прецедентов

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.

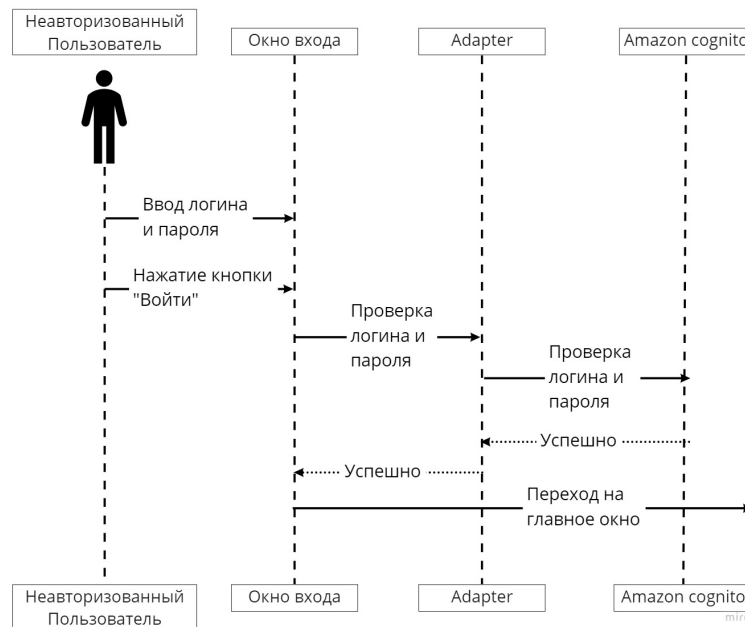


Рисунок 2 - Диаграмма последовательностей для неавторизованного пользователя в настольном приложении

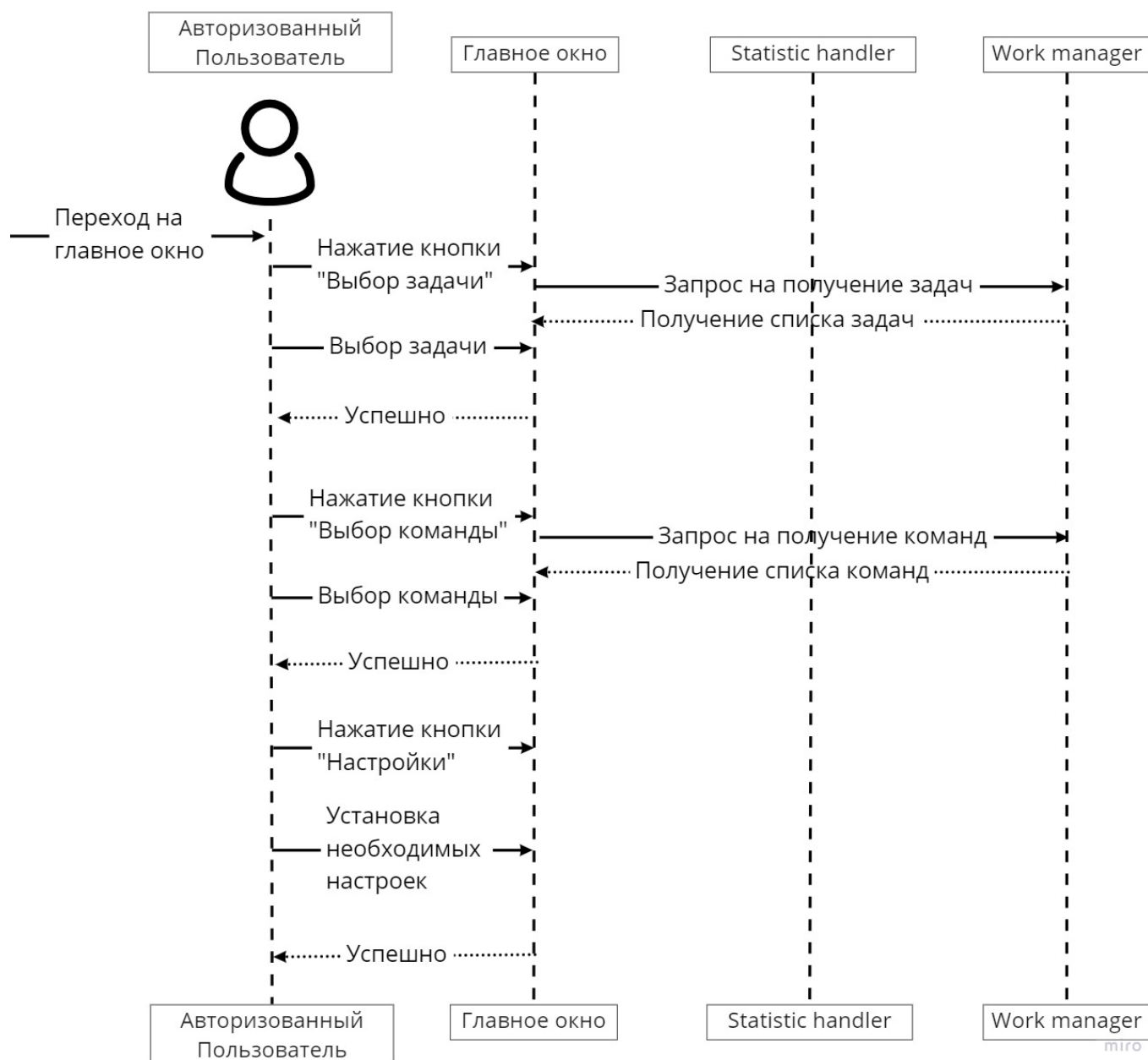


Рисунок 3 - Диаграмма последовательностей для авторизованного пользователя в настольном приложении (часть 1)

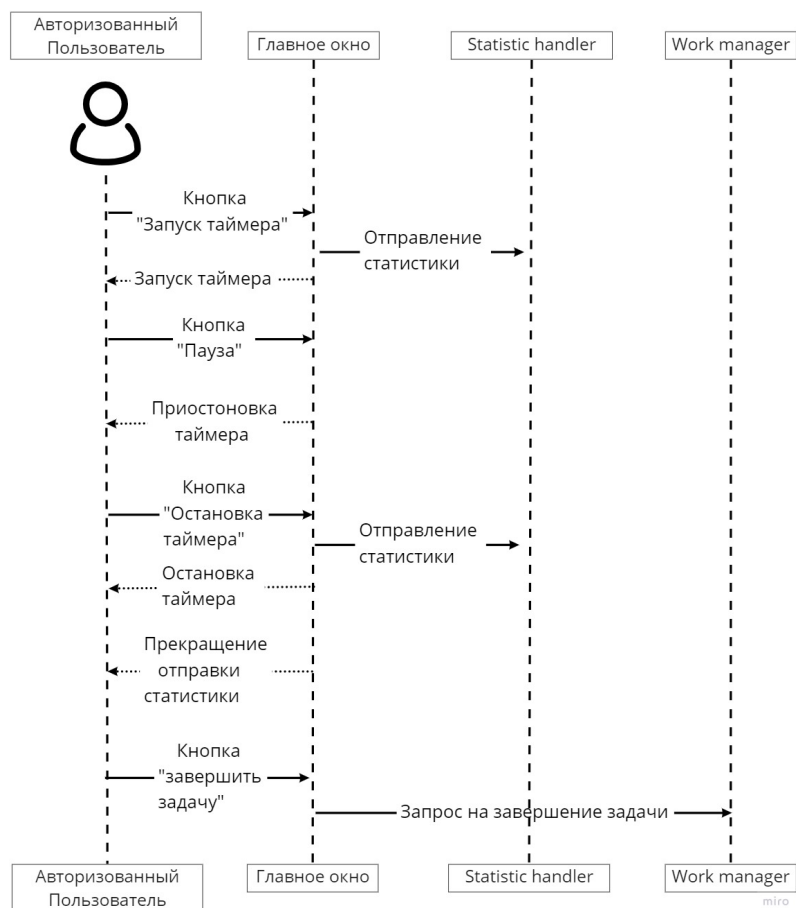


Рисунок 4 - Диаграмма последовательностей для неавторизованного пользователя в настольном приложении (часть 2)

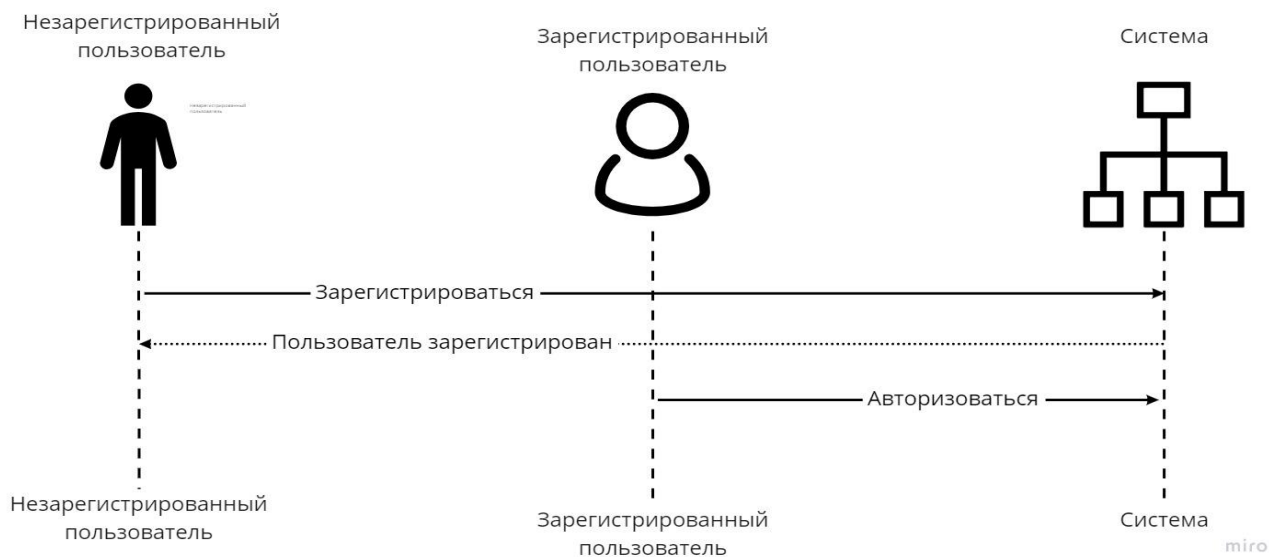


Рисунок 5 - Диаграмма последовательностей для неавторизованного пользователя в WEB приложении

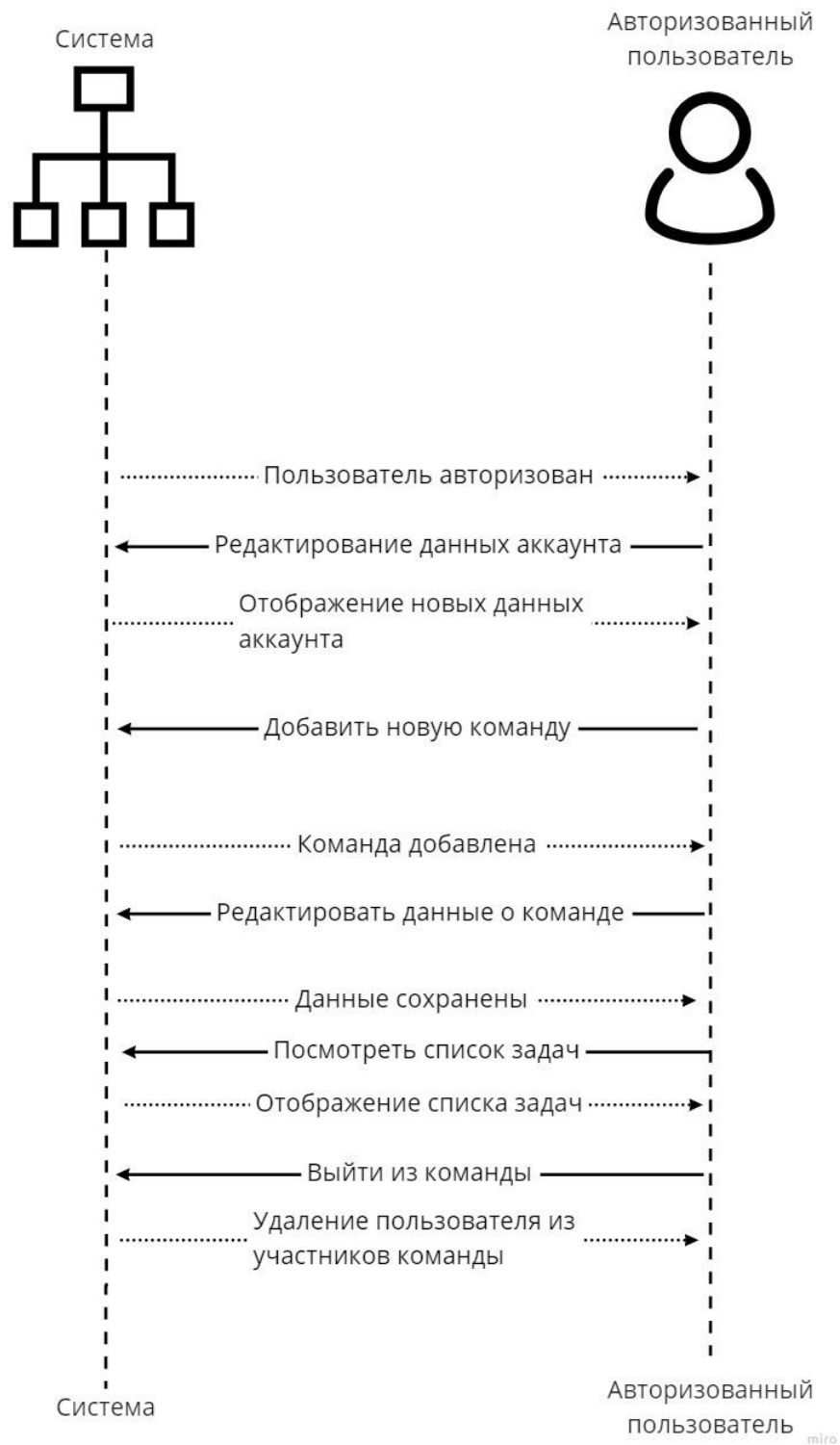


Рисунок 6 - Диаграмма последовательностей для авторизованного пользователя в WEB приложении



Рисунок 7 - Диаграмма последовательностей для лидера команды в WEB приложении

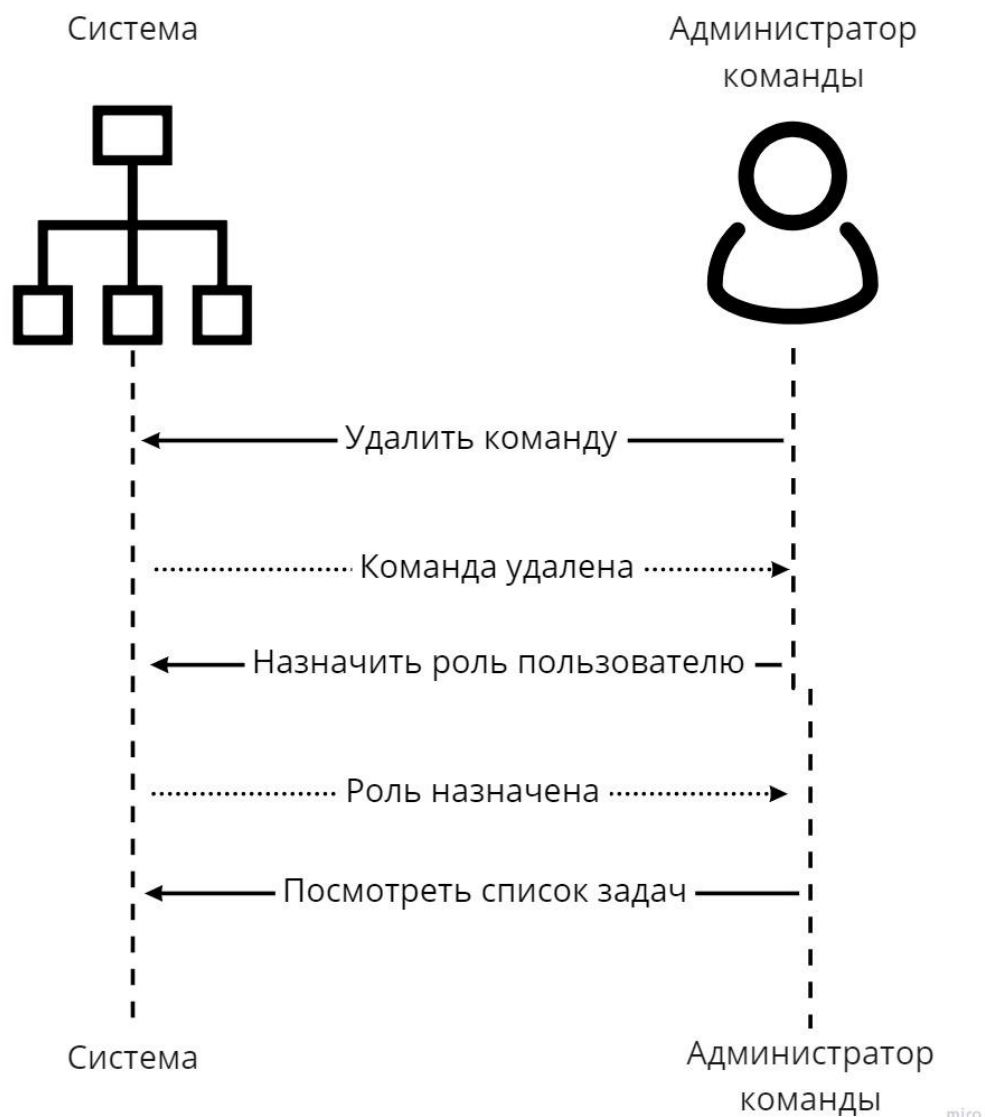


Рисунок 8 - Диаграмма последовательностей для администратора команды в WEB приложении

2.3.3. Варианты состояния системы

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

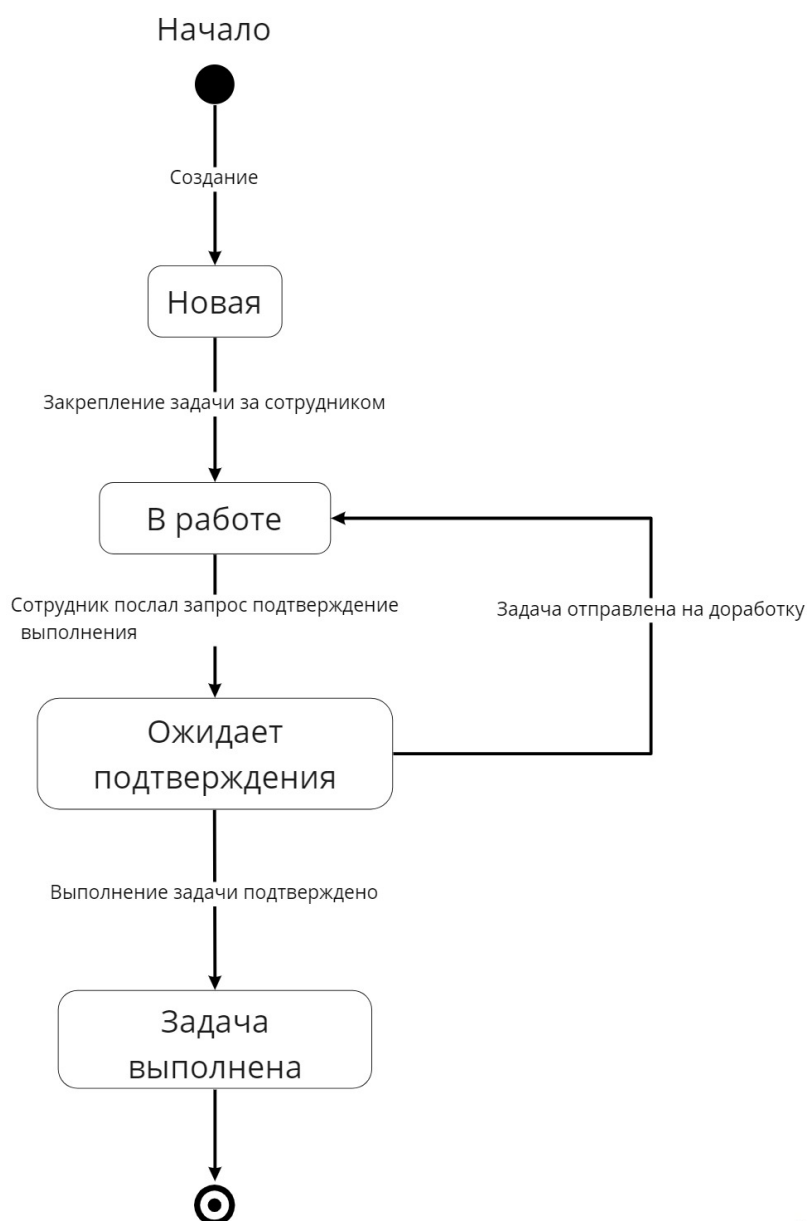


Рисунок 9 - Диаграмма состояний задачи

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 считать выполненной, ей присваивается статус “выполнено”.

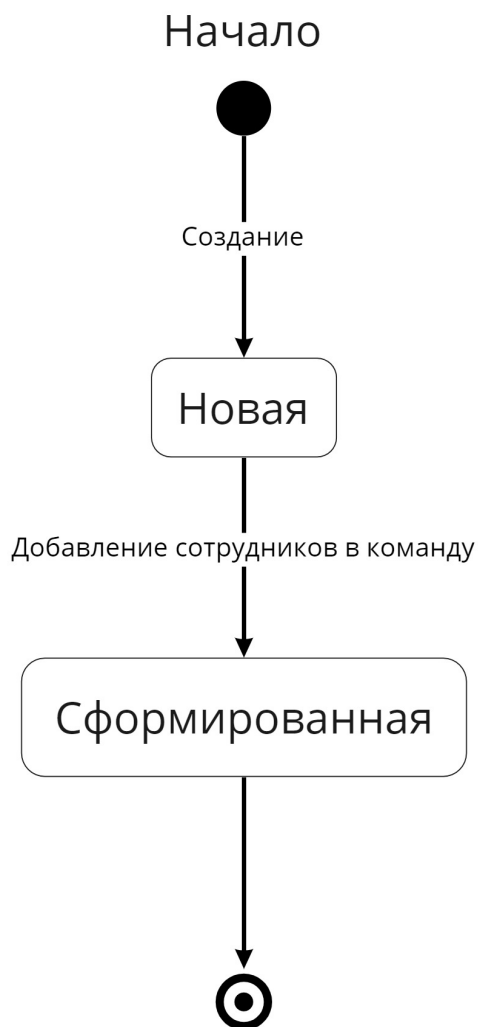


Рисунок 10 - Диаграмма состояний команды

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

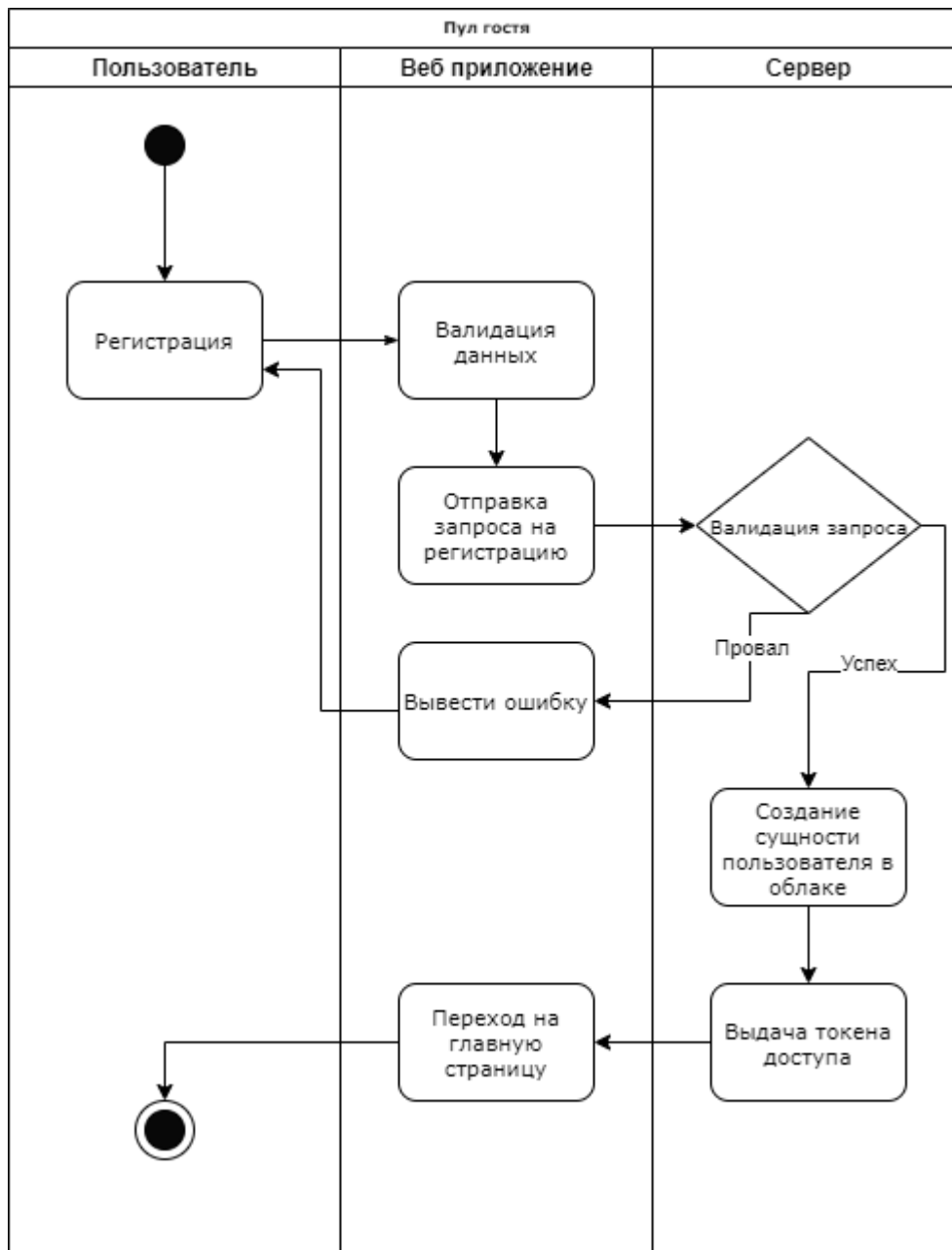


Рисунок 11 - Диаграмма активности гостя

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

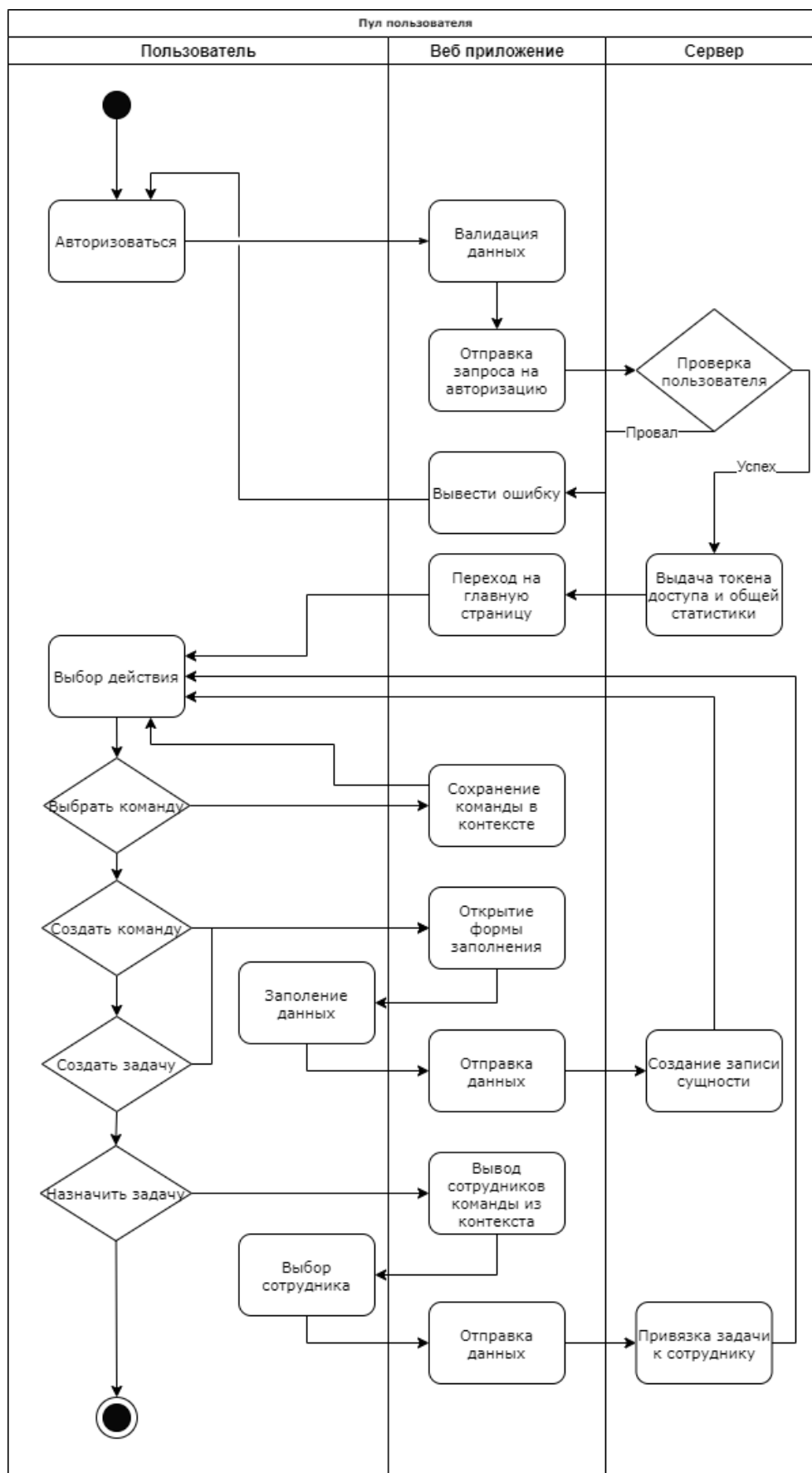


Рисунок 12 - Диаграмма активности пользователя

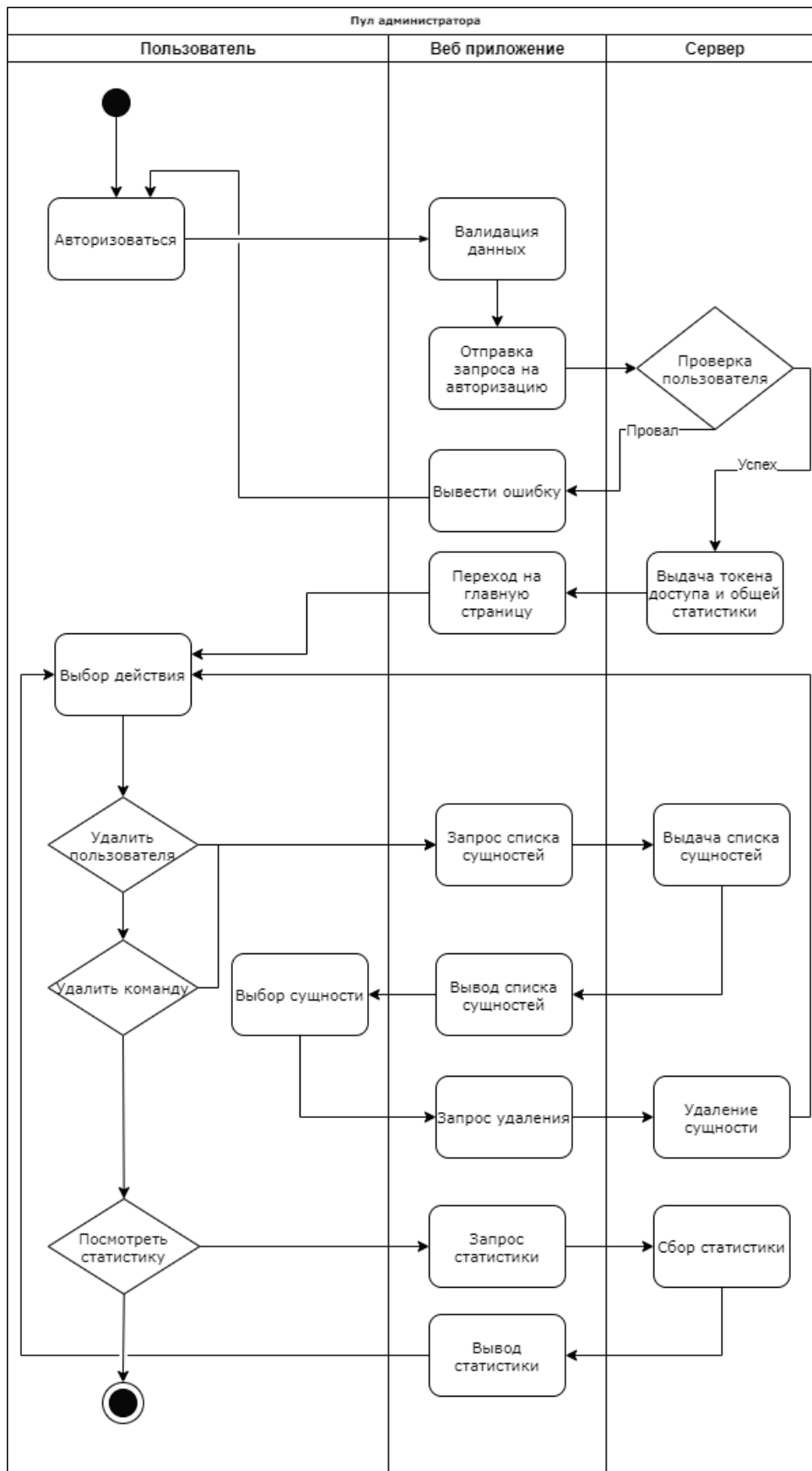


Рисунок 13 - Диаграмма активности администратора

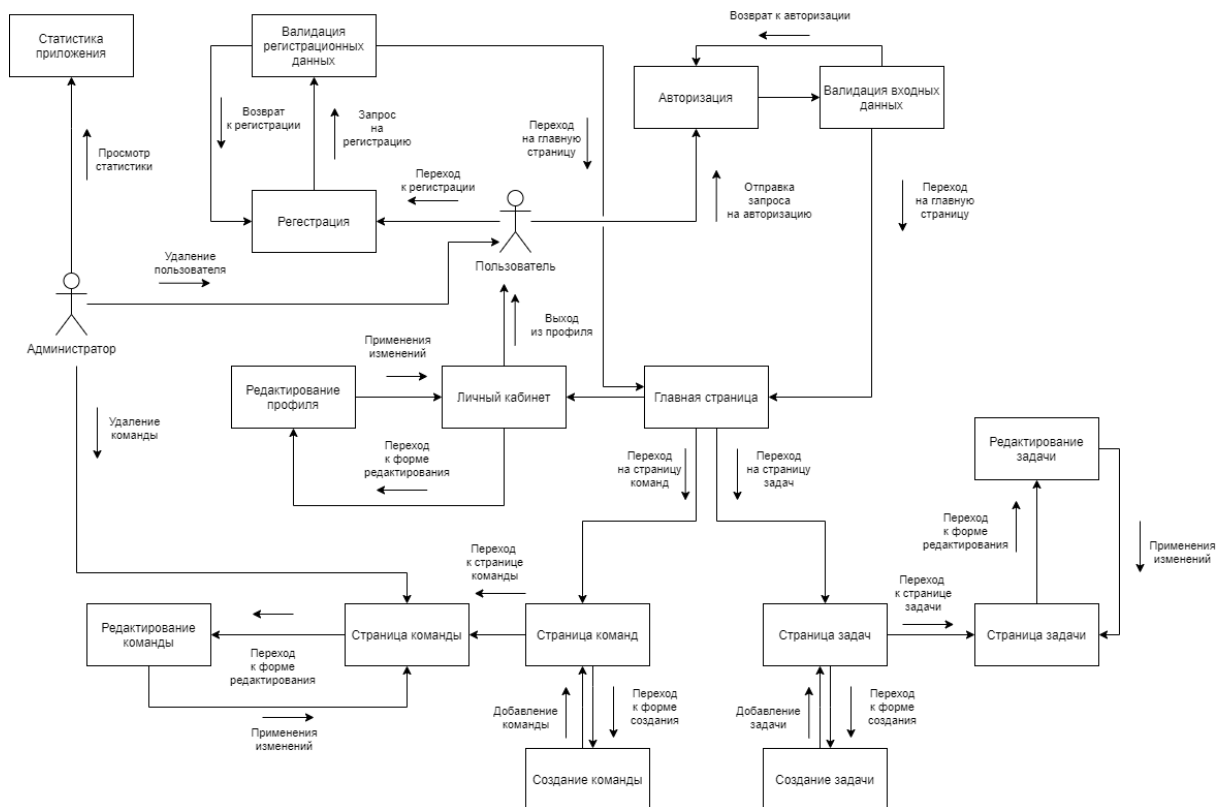


Рисунок 14 - Диаграмма взаимодействий

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

2.3.5. Развертывание приложения

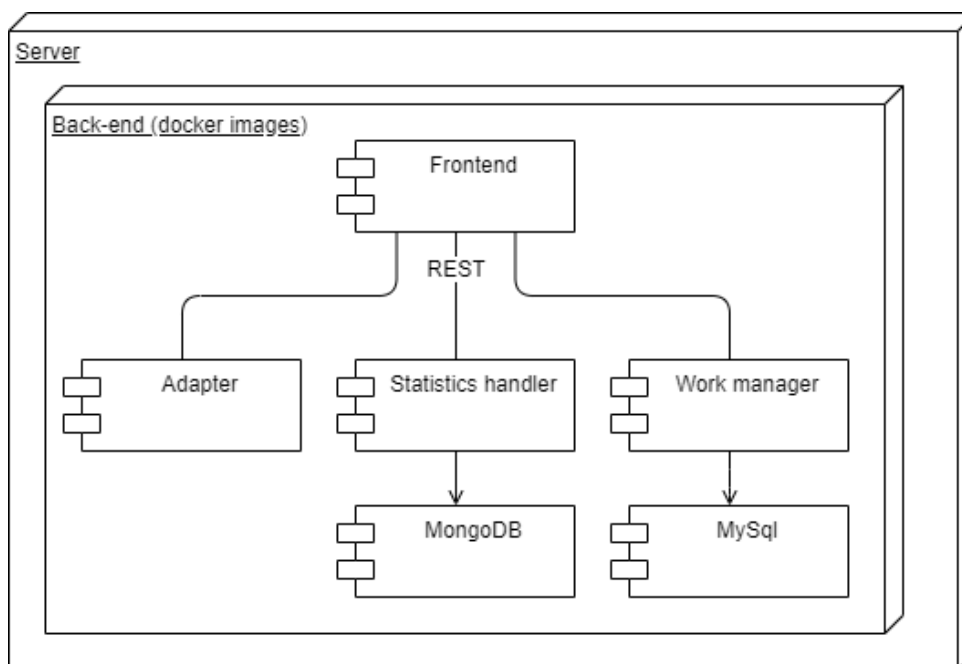


Рисунок 15 - Диаграмма развертывания

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

3. Анализ средств реализации

4. Реализация

4.1. Сущности

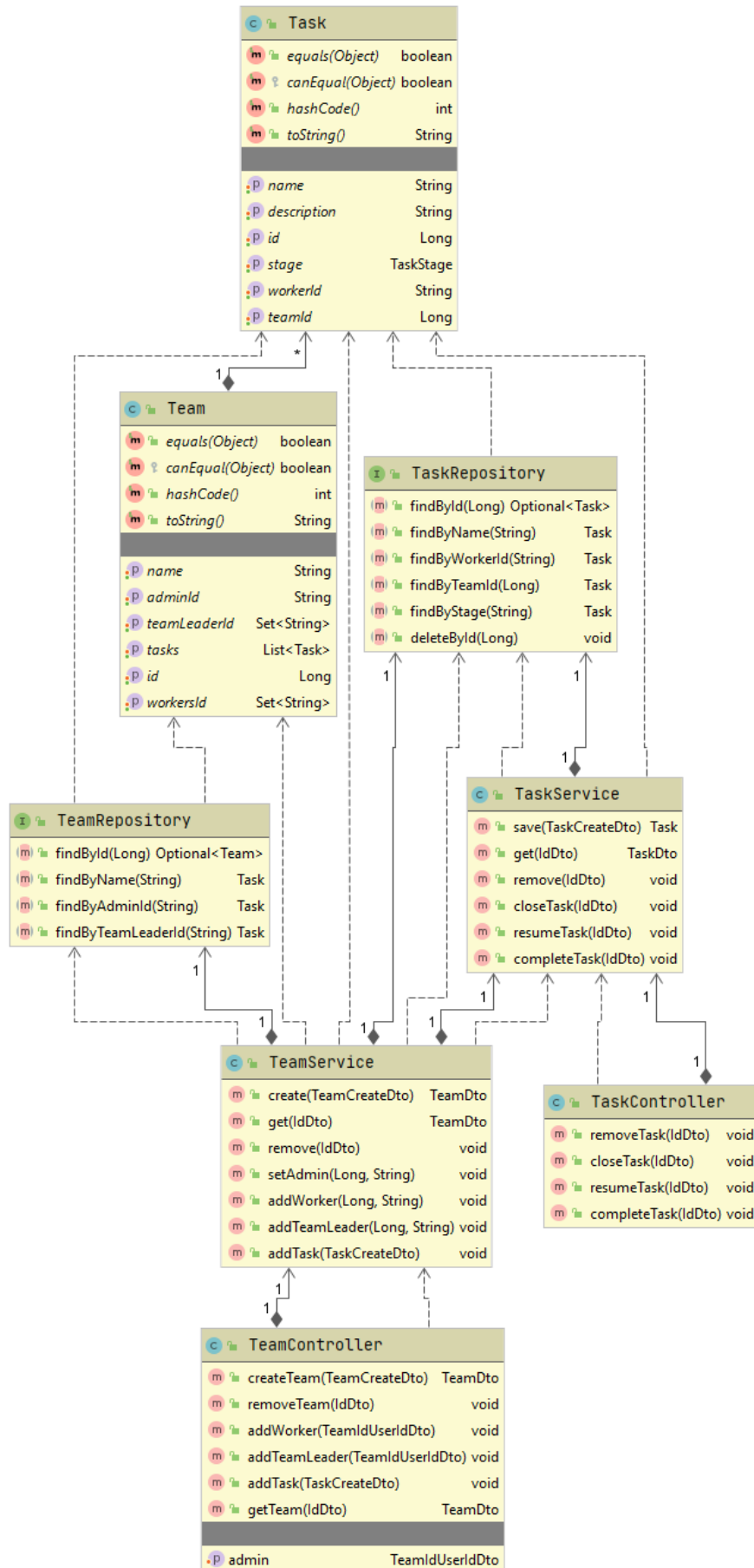


Рисунок 16 - Диаграмма классов Work Manager

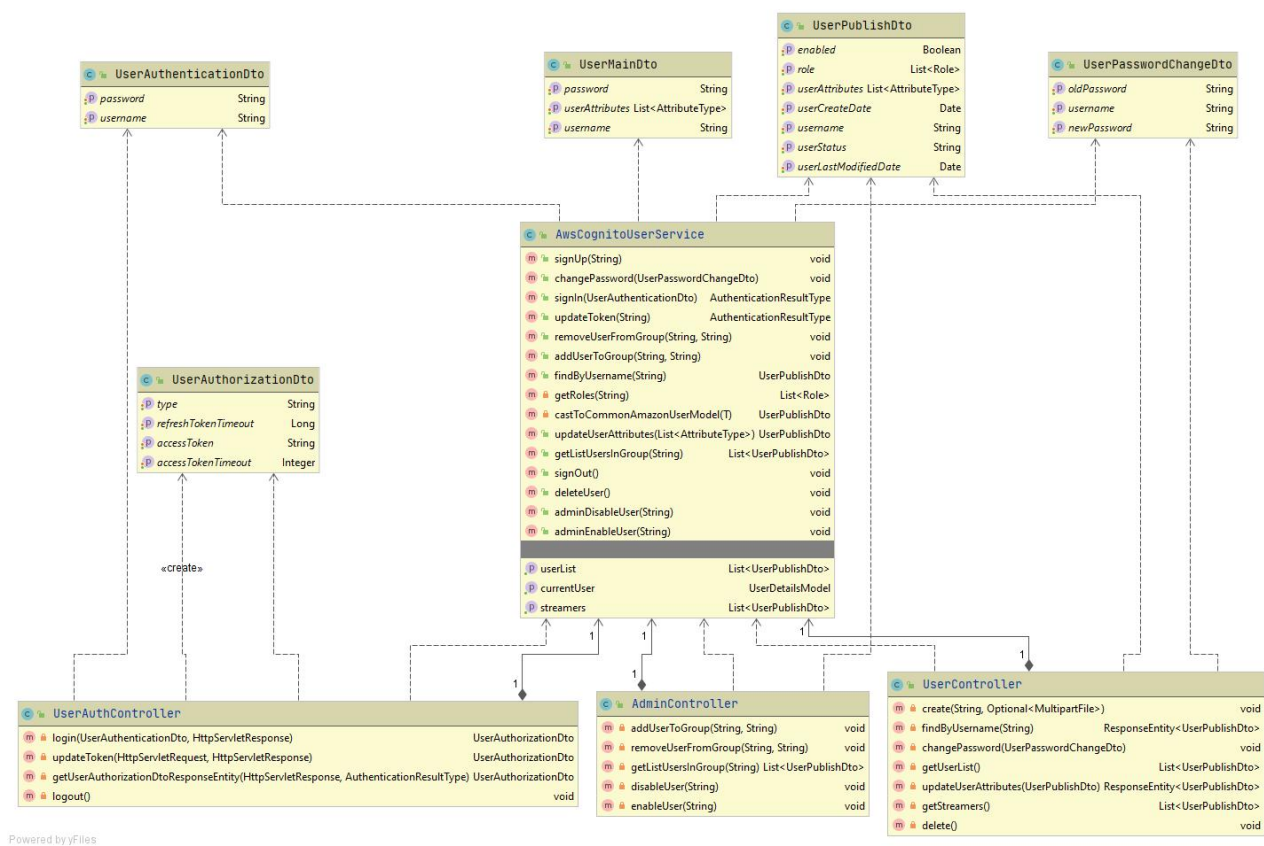


Рисунок 17 - Диаграмма классов Adapter

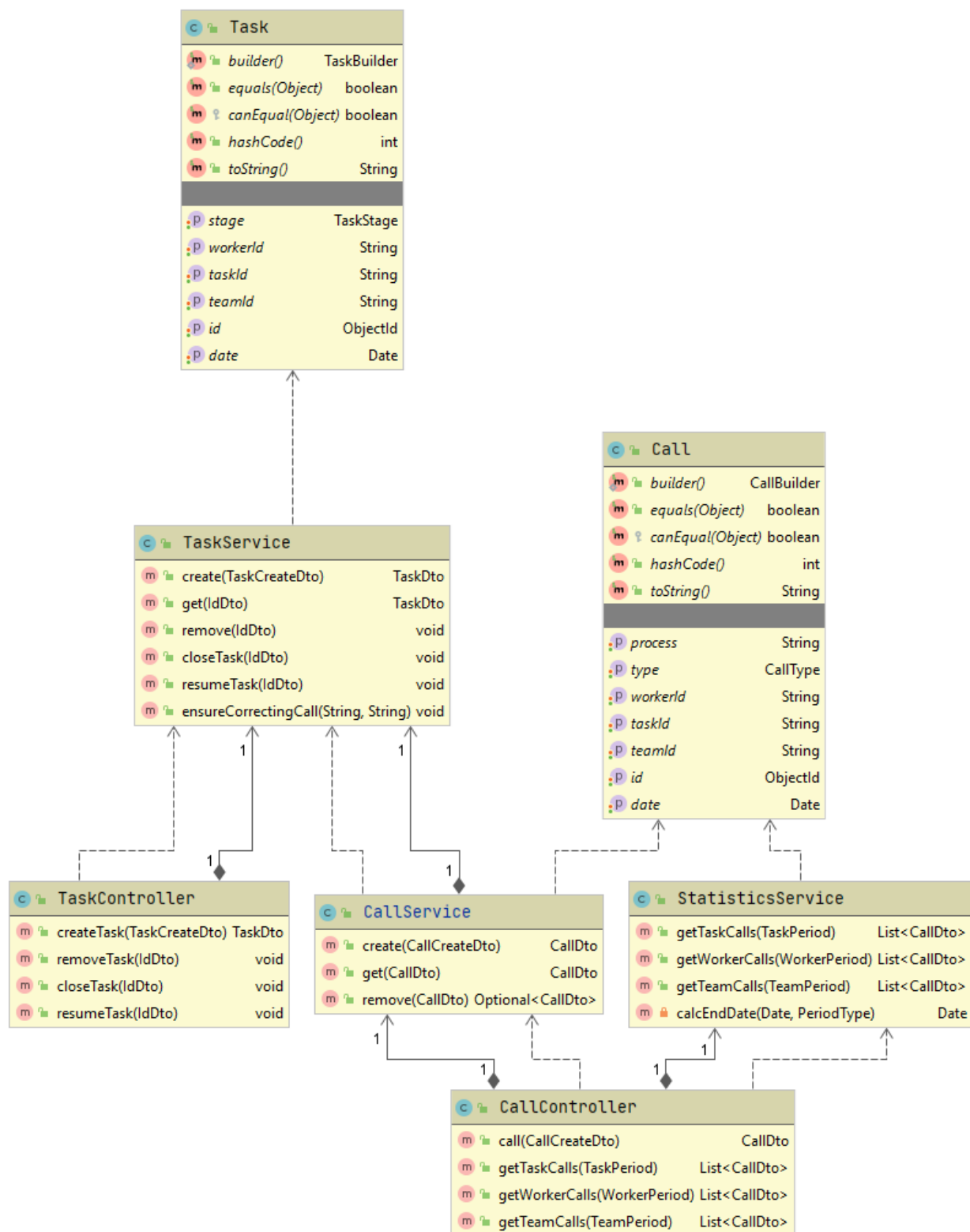


Рисунок 18 - Диаграмма классов Statistic Handler

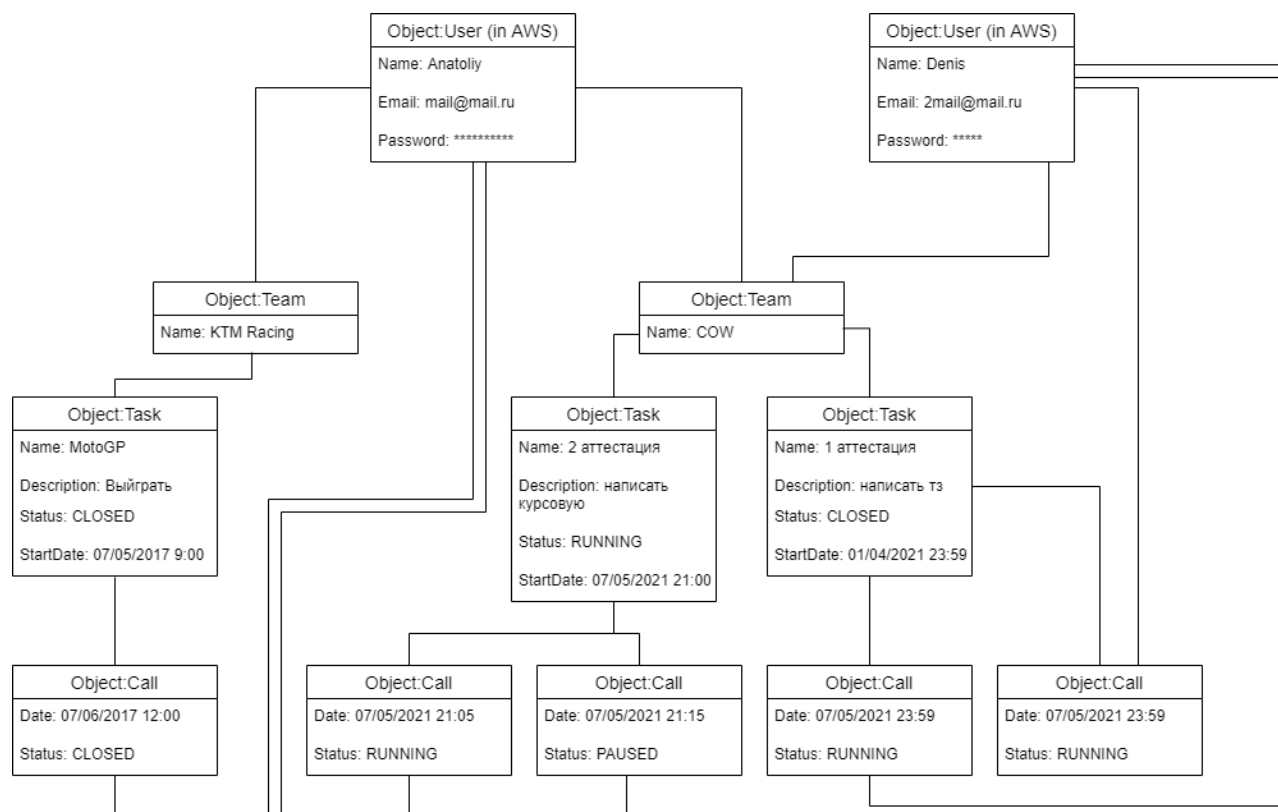


Рисунок 19 - Диаграмма объектов

4.2. Графический интерфейс

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.