

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

Кафедра Прикладная математика и информатика
(наименование)

02.03.03 Математическое обеспечение и администрирование информационных систем
(код и наименование направления подготовки)

Мобильные и сетевые технологии
(направленность (профиль))

**ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА
(БАКАЛАВРСКАЯ РАБОТА)**

на тему «Разработка мобильного приложения "Менеджер Задач" для
компании "Мегахенд"»

Обучающийся М.И. Вагапов
(Инициалы Фамилия) (личная подпись)

Руководитель к.т.н., доцент, Д.Г. Токарев
(ученая степень (при наличии), ученое звание (при наличии), Инициалы Фамилия)

Консультант к.п.н., доцент С.А. Гудкова
(ученая степень, звание, И.О. Фамилия)

Аннотация

Представленная квалификационная работа включает в себя три главы общим объемом 43 страниц. Состоит из введения, 24 рисунков, 3 таблиц, списка использованной литературы из 20 источников, включая литературу на английском языке.

Объектом исследования данной работы обозначен анализ деятельности планирования корпоративных задач и его реорганизация. Предметом исследования – программная разработка приложения для мобильных устройств, предназначенного для автоматизации процесса планирования задач. Цель выпускной работы представляет собой непосредственно создание мобильного приложения для планирования задач «Mhand Task».

Актуальность выпускной квалификационной работы обусловлена отсутствием на рынке приложений подходящего для организации по функционалу и интерфейсу планировщика задач, с помощью которого бы удалось улучшить планирование задач. Структура работы состоит из введения, трех глав, заключения и приложения.

В первой главе анализируется бизнес-процесс компании, после чего формируются требования к планировщику задач, проводится обзор аналогов и формулируется задача на разработку мобильного приложения.

Во второй главе рассматривается вопрос проектирования и разработки мобильного приложения для планирования задач «Mhand Task». Для этого выбрана платформа JetBrains WebStorm, а язык программирования JavaScript.

В третьей главе представлена реализация мобильного приложения, описывается процесс создания задач, оповещения пользователя и другие аспекты.

ABSTRACT

The title of the graduation project is Development of a Mobile Application for Task Planning «Mhand Task».

This graduation project is devoted to the analysis and reorganization of corporate task planning activities through the development of a mobile application aimed at automating the task planning process. The object of the research is the corporate task planning activities and their reorganization. The subject of the research is the software development of a mobile application designed for task planning automation. The aim of the work is to create a mobile application for task planning, named «Mhand Task».

The relevance of the graduation project is highlighted by the current market's lack of applications suitable for organizational task planning, which could significantly improve task planning processes.

This graduation project consists of an explanatory note on 43 pages, including an introduction, 24 figures, 3 tables, and a list of 20 references, some of which are in English. The structure of the work includes an introduction, three chapters, a conclusion, and an appendix.

In the first chapter, the author analyzes the business processes of the company, formulates requirements for the task planner, reviews existing analogs, and defines the task for developing the mobile application.

The second chapter is dedicated to the design and development of the mobile application «Mhand Task». The platform chosen for development is JetBrains WebStorm, and the programming language used is JavaScript.

The third chapter presents the implementation of the mobile application, detailing the process of task creation, user notifications, and other aspects.

In conclusion, the author emphasizes that the developed application can potentially fill the gap in the market for a suitable task planning application, thereby improving task planning efficiency.

Оглавление

Введение.....	Ошибка! Закладка не определена.
Глава 1 Исследование рабочих процессов организации.....	7
1.1 Краткая характеристика компании.....	7
1.2 Выбор инструментов CASE для моделирования бизнес-процессов.....	9
1.3 Оценка структуры бизнес-процессов.....	11
1.4 Оценка альтернативных приложений для планирования.....	15
1.5 Формулировка требований к мобильному приложению.....	18
1.6 Определение задач на разработку мобильного приложения.....	19
Глава 2 Проектирование мобильного приложения «Mhand Task».....	23
2.1 Определение платформы для разработки приложения.....	23
2.2 Определение средств для разработки.....	24
2.3 Разработка модели данных.....	Ошибка! Закладка не определена.
2.4 Структура программного обеспечения.....	27
Глава 3 Реализация мобильного приложения «Mhand Task».....	30
3.1 Сжатое изложение созданного решения.....	30
3.2 Реализация постановки задачи.....	31
3.3 Реализация редакции личного профиля пользователем.....	34
3.4 Тестирование мобильного приложения «Mhand Task».....	37
Заключение.....	40
Список используемой литературы.....	42
Приложение А Программный код класса DialogTask.vue.....	44

Введение

В наше время мобильное приложение представляет собой нечто большее, чем простая малофункциональная программа на смартфоне. Сейчас мобильные приложения представляют собой неотъемлемую часть кроссплатформенных сервисов, направленных на упрощение и улучшение взаимодействия с потребителями. В число таких сервисов входят и приложения по организации и планированию рабочего пространства внутри предприятий и корпораций.

Актуальность этого исследования продиктована необходимостью совершенствования методов планирования в современной цифровой среде. Большую популярность и востребованность демонстрируют приложения управления задачами на мобильные устройства. Включая большое разнообразие данных приложений, ООО «Море Сторе» не смогло для себя определить приложение с необходимым функционалом, из-за чего возможность сотрудников эффективно планировать задачи - отсутствовала.

Процесс организации планирования корпоративных задач представляется в данной работе как объект исследования, в свою очередь автоматизация озвученного процесса - ее предмет исследования. Целью данной работы является проектирование и разработка мобильного приложения «Mhand Task», которое улучшит управление рабочим процессом сотрудников и позволит им планировать корпоративные задачи на личных и рабочих мобильных устройствах.

Для достижения этой цели необходимо выполнить следующие задачи:

- осуществить полный анализ бизнес-процесса компании: в него входит оценка особенности самого процесса планирования и определение требований к разрабатываемому мобильному приложению;

- разработать проект мобильного приложения: выбор инструмента разработки, определение архитектуры приложения и разработка его модели данных;
- непосредственная реализация мобильного приложения: разработка функционала по созданию, ведению и редактированию корпоративных задач, а также редакции профиля пользователя;
- реализация тестирования мобильного приложения: организация условий проведения тестирования, сценариев тестирования и работоспособность основного функционала.

Работа состоит из трех основных глав.

Первая глава представляет собой анализ имеющихся на рынке приложений для планирования задач. Аргументируется выбор платформы для реализации мобильного приложения, а также представляются диаграммы использования для общего представления взаимодействия конечного пользователя с разработанным продуктом.

Вторая глава посвящена проектированию приложения, анализу и определению с наиболее оптимальным инструментом разработки, а также архитектуры мобильного приложения.

Третья глава демонстрирует процесс реализации мобильного приложения. Описана реализация основного функционала продукта, предназначенного для непосредственного взаимодействия пользователя с мобильным приложением. Рассматривается работа мобильного приложения, а также его тестирование.

Данная выпускная квалификационная работа для ООО «Море Сторе» представляет практическую ценность, так как предоставляет востребованную возможность в эффективном управлении процессом.

Глава 1 Исследование рабочих процессов организации

1.1 Краткая характеристика компании

«ООО «Море Сторе» - это общество с ограниченной ответственностью, реализующее розничную торговлю, осуществляемую непосредственно при помощи информационно-коммуникационной сети Интернет» [7]. Основная задача компании в организации сети магазинов и интернет пространства для розничной торговли вещами, импортированными из зарубежных стран.

Компания обеспечивает повседневной, качественной и комфортной одеждой все слои населения, позволяет это сделать во многих городах России по доступным ценам. Одна из основных задач компании в усовершенствовании интернет продуктов для активного взаимодействия с молодым населением.

Структура ООО «Море Сторе» описана на рисунке 1, ключевым лицом является генеральный директор компании, он отвечает за организацию работы компании, а также курирует работу всех отделов головного офиса в городе Тольятти, начиная от маркетинга, заканчивая IT-отделом. Он определяет путь развития компании на долгие годы, реализовывая свои цели. На данный момент действующим генеральным директором ООО «Море Сторе» является Александр Сергеевич Дюльдин.

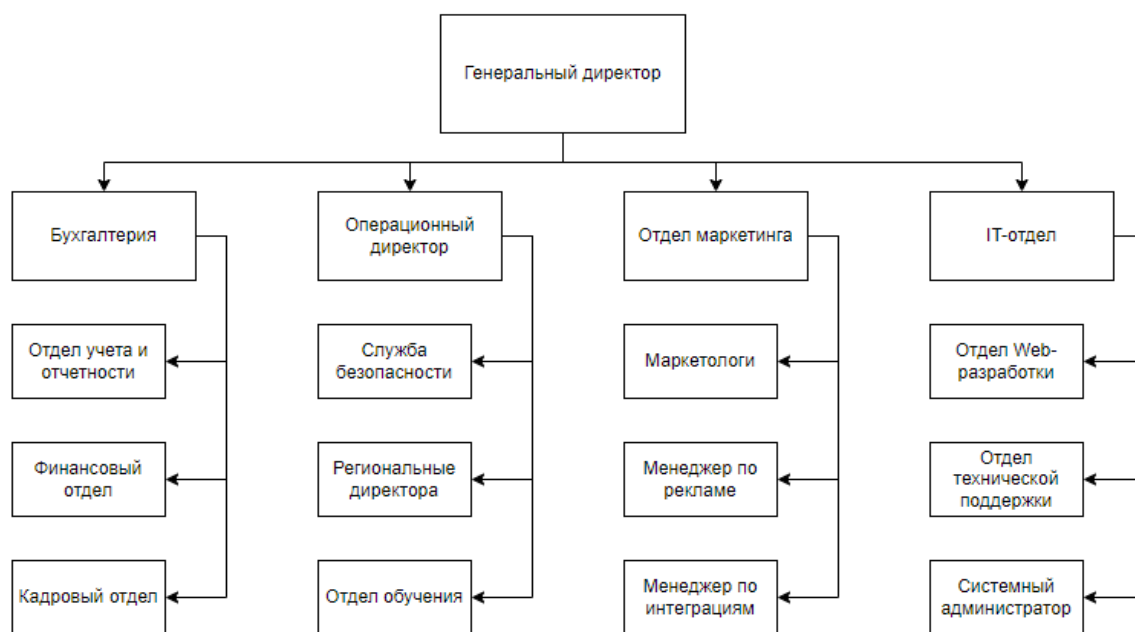


Рисунок 1 - Организационная структура ООО «Море Стор»

Бухгалтерия занимается финансовыми вопросами деятельности магазинов сети и интернет ресурса: кассовые отчеты, предоставление финансовой отчетности в налоговую службу, учетом имущества, организацией выплаты заработной платы, финансовая стратегия компании.

Операционный отдел занимается хозяйственными вопросами сети магазинов: организация помещений, учет товара, оформление договоров об аренде помещений, организация системы видеонаблюдения, предоставление учебных материалов для новых сотрудников.

Отдел маркетинга занимается вопросами по продвижению сети магазинов в информационном поле: уличная реклама, реклама на телевидении, реклама по радио, согласование бюджета на рекламу.

IT-отдел занимается в основном техническими вопросами: обслуживание оборудования, серверов, web-разработка интернет ресурса.

1.2 Выбор инструментов CASE для моделирования бизнес-процессов

«CASE средства используются при создании и разработке информационных систем управления предприятиями. Применительно к моделированию бизнес процессов они могут рассматриваться как инструментарий для совершенствования и непрерывного улучшения работы. CASE средства (Computer Aided Software Engineering) — это инструмент, который позволяет автоматизировать процесс разработки информационной системы и программного обеспечения. Разработка и создание информационных систем управления предприятием связаны с выделением бизнес-процессов, их анализом, определением взаимосвязи элементов процессов, оптимизации их инфраструктуры и т.д. Основной целью применения CASE средств является сокращение времени и затрат на разработку информационных систем, и повышение их качества. Многие современные CASE средства предоставляют возможности для моделирования практически всех предметных областей деятельности организаций. В составе этих средств существуют инструменты для описания моделей бизнес-процессов за счет различных диаграмм, схем, графов и таблиц» [11].

Одним из CASE-средством является «Microsoft Visio - включает в себя: создание диаграммы на основе данных и данные на основе диаграмм (в том числе экспортировать данные в отчеты); возможность создания диаграммы с внешними данными; возможность масштабирования диаграммы до нужного уровня точности; возможность создания интерактивные панели показателей; возможность создания отчетов» [2].

Еще одним CASE-средством можно выделить «Diagrams.net - это кроссплатформенное программное обеспечение для рисования графов, разработанное на HTML5 и JavaScript. Его интерфейс можно использовать для создания таких диаграмм, как блок-схемы, wireframes, UML-диаграммы, организационные диаграммы и сетевые диаграммы. Его автономное

приложение построено с использованием фреймворка Electron. Веб-приложение не требует онлайн-входа или регистрации и может открываться с локального жесткого диска и сохраняться на нем. Кроме того, приложение интегрируется с облачными сервисами для хранения данных, включая Dropbox, OneDrive, Google Drive, GitHub и GitLab.com» [15].

Стоит также упомянуть и о CASE-средстве «Ramus: Программа Ramus Educational позволяет создавать структуру классификации и кодирования всех объектов, которые участвуют в бизнес процессах предприятия. Эта особенность системы позволяет детально рассмотреть диаграммы потоков данных системы в нотации DFD» [3].

Для того чтобы детальнее рассмотреть CASE-средства и выбрать наиболее подходящее из них, были выбраны критерии для сравнительной оценки: импорт и экспорт в различные форматы, интеграции с другими программами, поддержка работы в разных нотациях, бесплатный доступ к программе.

В таблице 1 был представлен анализ по критериям для работы с программой и в качестве наиболее подходящего инструмента для описания бизнес-процессов, была выбрана платформа Diagrams.net, которая полностью соответствует всем требованиям для проведения сравнительного анализа.

Таблица 1 – Аналитическое сравнение CASE-средств

	Diagrams.net	Microsoft Visio	Ramus 2.0
Интеграция с другими программами.	+	+	-
Возможность работы в нотациях.	+	+	-

Продолжение таблицы 1

Импорт и экспорт в иные форматы.	+	-	+
Бесплатный доступ к программе.	+	-	+

В таблице было показано, что Diagrams.net удовлетворяет всем установленным критериям. Ramus 2.0 не соответствует таким критериям, как возможность работы в других нотациях и интеграция с другими приложениями. Microsoft Visio как CASE-средство не соответствует требованиям бесплатного доступа и возможности импорта и экспорта в другие форматы.

1.3 Оценка структуры бизнес-процесса

В данный момент для организации своей рабочей деятельности сотрудник головного офиса самостоятельно находит решение, подходящее под его требования. Контекстная диаграмма этого процесса продемонстрирована на рисунке 2.

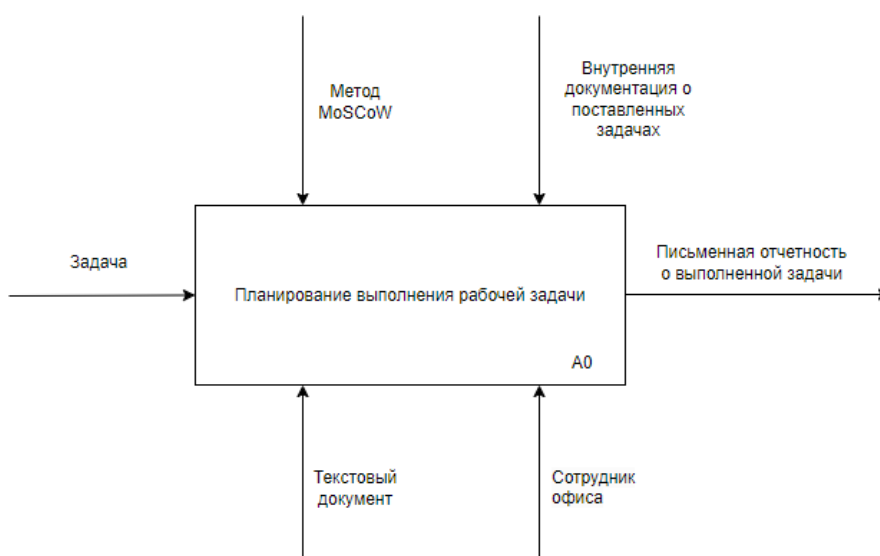


Рисунок 2 - Диаграмма контекста планирования выполнения корпоративной задачи

Ради более осязаемого понимания данного процесса, на рисунке 3 представлена декомпозиция блока А0.

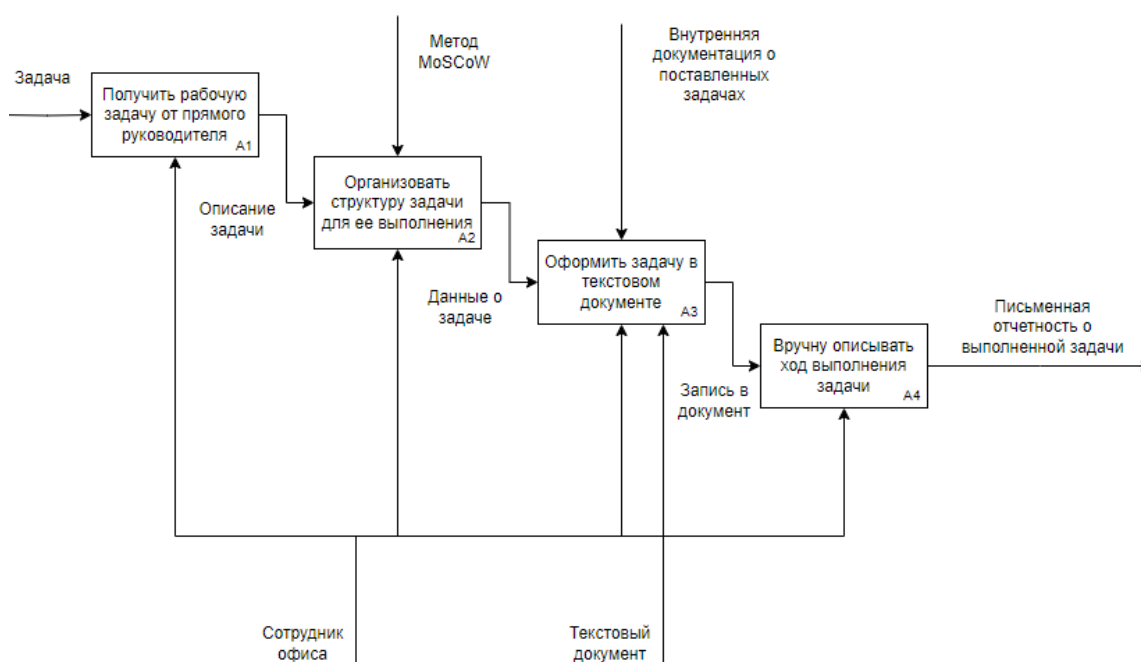


Рисунок 3 - Декомпозиция блока А0 процесса функциональной модели «Метод MoSCoW - это метод расстановки приоритетов, используемый в менеджменте, бизнес-анализе, управлении проектами и разработке программного обеспечения для достижения общего понимания с заинтересованными сторонами относительно важности, которую они придают выполнению каждого требования; он также известен как расстановка приоритетов MoSCoW или анализ MoSCoW. Сама аббревиатура означает следующее:

- m (Must) - являются критическими для успешного выполнения проекта в текущий период времени. Если хотя бы одно требование Must have не включено, проект следует считать неудачным;

- s (Should) - являются важными, но не необходимыми для поставки в текущий срок. Хотя требования Should have могут быть такими же важными, как и Must have, они часто не так критичны по времени, или может существовать другой способ удовлетворить требование, чтобы его можно было отложить до будущего срока поставки;
- c (Could) - желательны, но необязательны и могут улучшить пользовательский опыт или удовлетворенность клиентов за небольшую стоимость разработки. Обычно они включаются, если позволяют время и ресурсы;
- w (Would) - наименее критичные, с наименьшими затратами или не подходящие в данный момент. Не планируются в расписании на следующий срок поставки: либо отменяются, либо пересматриваются для включения в более поздний срок.

Методология MoSCoW обеспечивает систематическую классификацию требований и задач по их приоритетности, позволяя определить, какие элементы являются наиболее важными и необходимыми» [18].

Для более внимательного погружения в суть вопроса необходимо выполнить декомпозицию ключевых блоков, в нашем случае это А3 и А4, представленные на рисунках 4 и 5 соответственно.

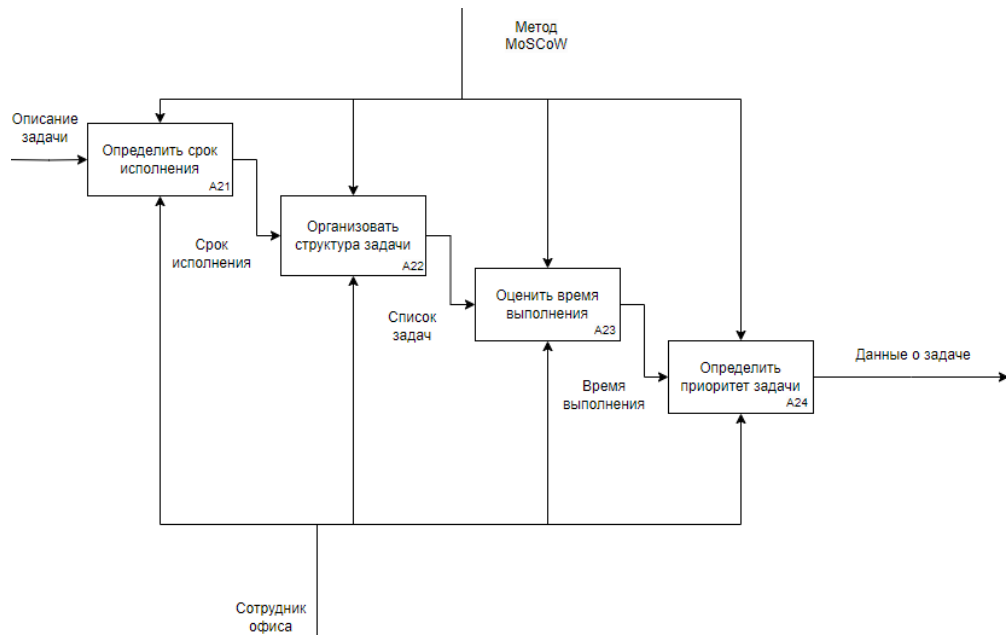


Рисунок 4 - Декомпозиция блока А

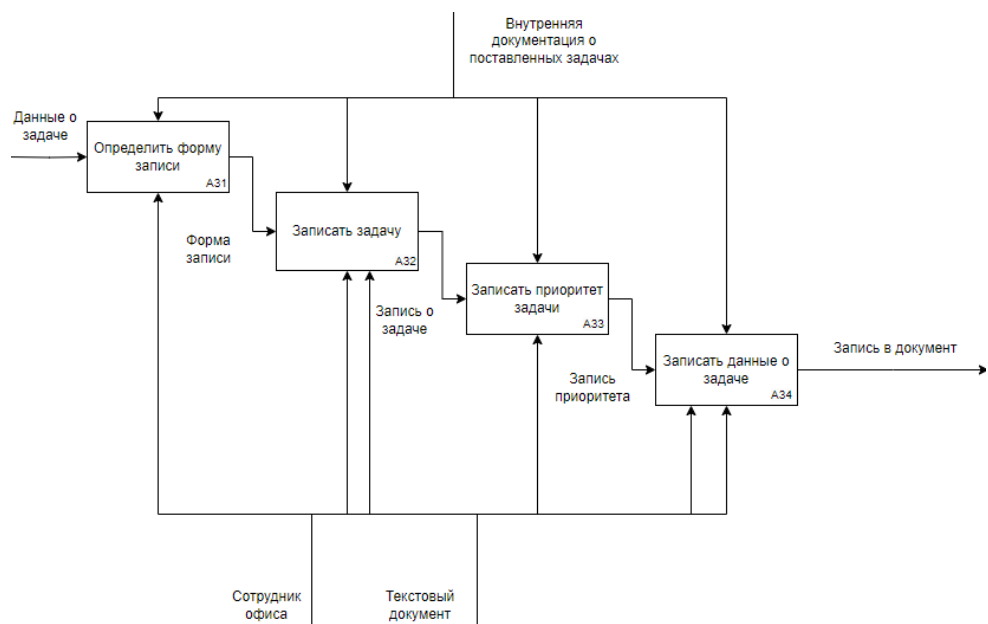


Рисунок 5 - Декомпозиция блока А

Визуальное ознакомления с процессом нам демонстрирует DFD-диаграмма на рисунке 6.

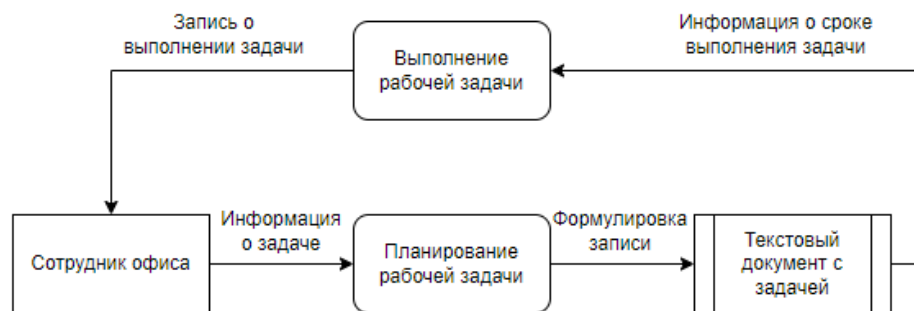


Рисунок 6 - DFD-диаграмма процесса планирования задач

В качестве результат можно обозначить детальное рассмотрение контекстной диаграммы, декомпозиции ее блоков и DFD-диаграмму данных.

1.4 Оценка альтернативных приложений для планирования

Нельзя не отметить, что на рынке мобильных приложений есть множество разновидностей данных планировщиков задач, мы рассмотрим наиболее популярные варианты из главного магазина приложений мобильной операционной системы Android - Google Play, а также главного магазина мобильной операционной системы IOS - App Store.

«Google Play — магазин приложений, а также игр и книг от компании Google, позволяющий сторонним компаниям предлагать владельцам устройств с операционной системой Android устанавливать и приобретать различные приложения. В магазине присутствуют платные и бесплатные приложения. На сентябрь 2017 года в Google Play содержится более 3 миллионов приложений для Android» [17].

«App Store — магазин приложений, разработанный и поддерживаемый компанией Apple для мобильных устройств на базе операционных систем iOS и iPadOS. App Store предлагает более 1,5 миллиона приложений для iPhone, iPod Touch и около 725 тыс. для iPad, число загрузок на июнь 2015 года превысило 100 миллиардов, а пользовательская база составляет порядка 575

миллионов человек. В числе приложений множество категорий, включая игры, приложения социальных сетей и многое другое» [10].

В ходе решения задачи по автоматизации планирования рабочей деятельности был проанализирован рынок мобильных приложений аналогичных по функционалу. В ходе анализа были отобраны следующие критерии:

- функция назначения задачи сотруднику;
- возможность отслеживания статуса задачи;
- возможность добавления описания задачи;
- аналитика выполнения задач.

Критерий назначения задачи сотруднику очень необходим, когда речь идет не просто о назначении задач себе лично, а также возможности делегировать задачу другому сотруднику, либо же поставить задачу сотрудникам из другого отдела. Возможность отслеживания статуса задачи также необходимо при работе с коллективом, так как многие задачи могут иметь не одного исполнителя. Добавление описания задачи поможет сориентироваться сотруднику по задаче, иначе полностью формулировку придется описывать в названии, что не может не сказаться на визуальном ощущении от использования приложения. Аналитика выполнения задач поможет как самому сотруднику, так и его руководителю отследить рабочую эффективность сотрудника, просроченные задачи, актуальные принятые и завершенные, все это поможет реализовать аналитика.

Pyrus в Google Play, App Store – планировщик, который позволяет ставить задачи другим сотрудникам, ставить задачи самому себе, а также значительный функционал: календарь, список дел, структура организации, простой вариант реализации аналитики. Непосредственно для планирования задач организован следующий функционал: комментарии, описание, прикрепление файлов, делегирования, назначение ответственного и наблюдателя, указание затраченного времени на задачу и срок выполнения.

Мегаплан в Google Play – это планировщик заданий, доступный на

разнообразных платформах, включая iOS, Android, Windows, а также имеющий веб-версию. Схожий по функционалу с Purgus, за исключением некоторых особенностей: в приложении есть возможность ведения внутренних документов, бронирование отпуска, а также календарь мероприятий организации.

YouGile в Google Play, App Store – планировщик задач с очень нагруженным интерфейсом, большинство функций которого доступны лишь после оплаты модулей, однако имеющий обширную аналитику и в отличие от вариантов описанных выше – самостоятельно отслеживающий затраченное время на выполнение тех или иных задач.

Приложения самые разные по визуальной составляющей, схожие по функционалу, однако имеющие свои недостатки. По приложениям выше было проведено сравнение, результаты которого можно наблюдать в таблице 2.

Таблица 2 – Сравнение мобильных приложений

Мобильные приложения	Назначение задачи другому сотруднику	Возможность отслеживания статуса задачи	Возможность добавления описание задач	Аналитика выполнения задач
Purgus	+	-	+	+
Мегаплан	+	+	-	-
YouGile	-	-	+	+

В таблице наглядно продемонстрировано, что не одно из приложений не смогло показать положительный результат по всем объявленным критериям, это стало ключевым аспектом в необходимости разработки собственного приложения.

1.5 Формулировка требований к мобильному приложению

Разрабатываемое мобильное приложение будет организовано по классификационным требованиям к программным системам FURPS. «Аббревиатура организована от первых букв слов и расшифровывается так:

- Functionality (функциональность);
- Usability (удобство использования);
- Reliability (надежность);
- Performance (производительность);
- Supportability (поддерживаемость).

Впервые требования были разработаны и представлены Hewlett-Packard. В настоящее время используется аббревиатура FURPS+. Символ «+» означает дополнительные факторы, добавленные HP для расширения понятия и выделения второстепенных, но важных атрибутов: интерфейс, операции, юридические вопросы и другие аспекты» [16].

Функциональность приложения обусловлена требованиями организации к самому приложению, если основные критерии к планированию задач уже были описаны ранее, то пожелания компании будут приниматься к сведению при проектировании и разработке мобильного приложения. Функциональность должна охватывать весь спектр потребностей организации

Удобство использования заключается в понятном для пользователя интерфейсе, без перегруженного фона, с основной информацией, без скрытых функций самого приложения. Необходимо помнить о том, что этим приложением будут пользоваться люди абсолютно разного возрастного контингента.

Надежность мобильного приложения заключается в возможности бесперебойной работы у пользователя на его устройстве, без критических ошибок самого приложения и с постоянным доступом к нему.

Производительность системы включает в себя скорость, с которой приложение обрабатывает данные, время отклика программного устройства, а также управление телефонными ресурсами. Разработанный продукт обязан включать в себя оптимальное время отклика, а также потребление ресурсов устройства.

Не стоит забывать и о поддержке программного продукта, в себя оно может включать плановые обновления по ходу разработки дополнительного функционала, а также устранение любых недоработок, обнаруженных в процессе использования приложения.

1.6 Определение задач на разработку мобильного приложения

Достижение эффективного процесса планирования корпоративных задач невозможно без модернизации бизнес-процесса компании, обозначив основные идеи модернизации, а именно включения в бизнес-процесс мобильного приложения, для оптимизации процесса планирования корпоративных задач. Контекстная диаграмма процесса с мобильным приложением продемонстрирована на рисунке 7.

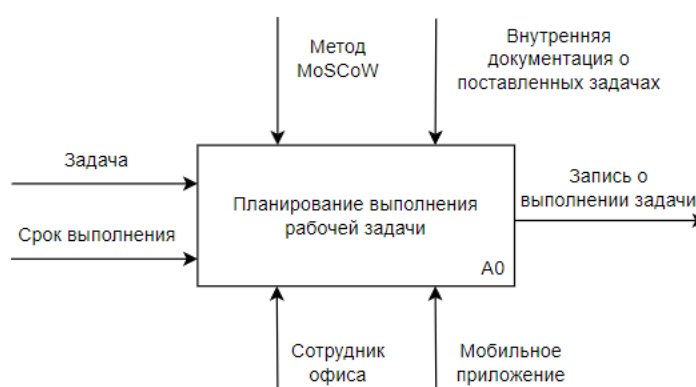


Рисунок 7 - Контекстная диаграмма процесса с мобильным приложением

Для изучения данной диаграммы более детально, на рисунке 8

представлена декомпозиция блока А0.

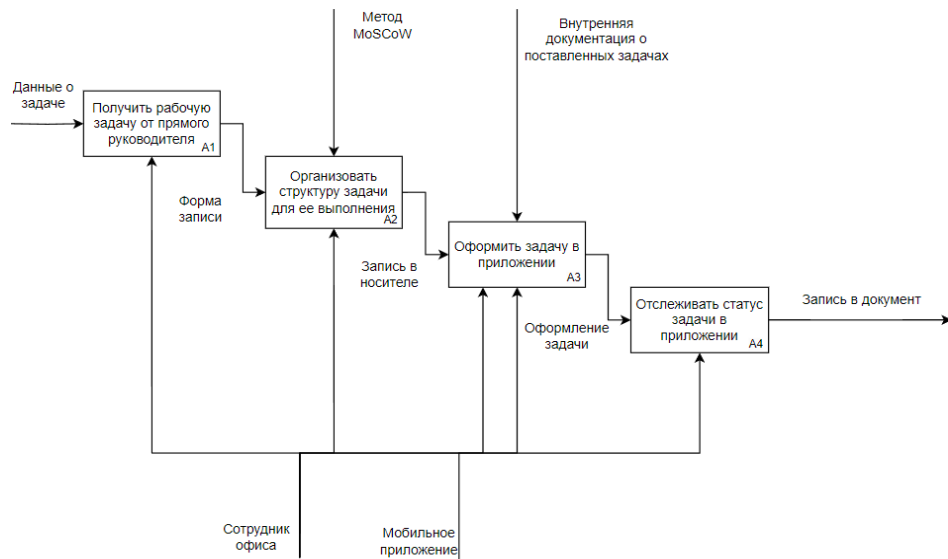


Рисунок 8 - Декомпозиция блока А0 модели «Планирование выполнения рабочей задачи»

Мы можем более детально изучить блоки А2 и А3, изображенные на рисунках 9 и 10 соответственно, если разложим их на составные части.

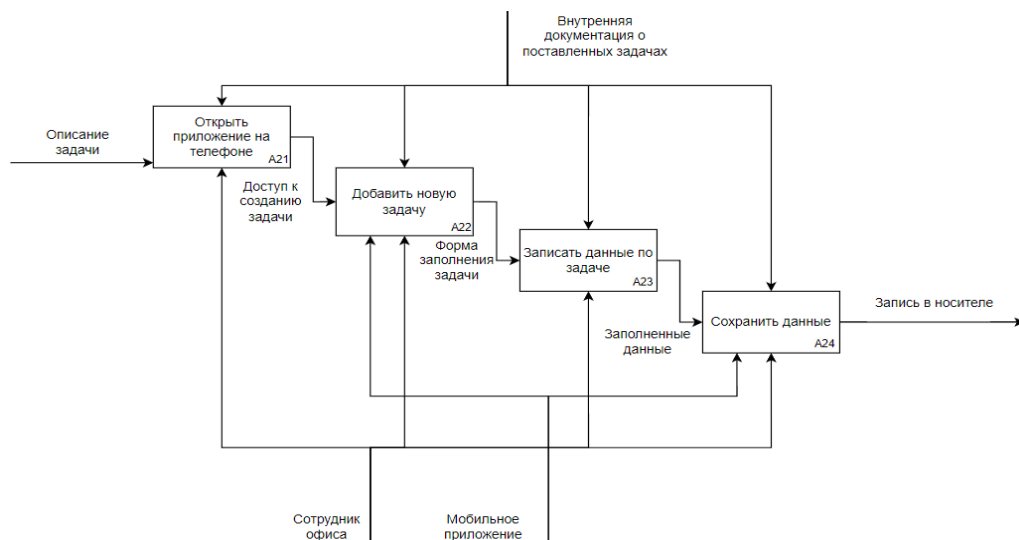


Рисунок 9 - Декомпозиция блока А2

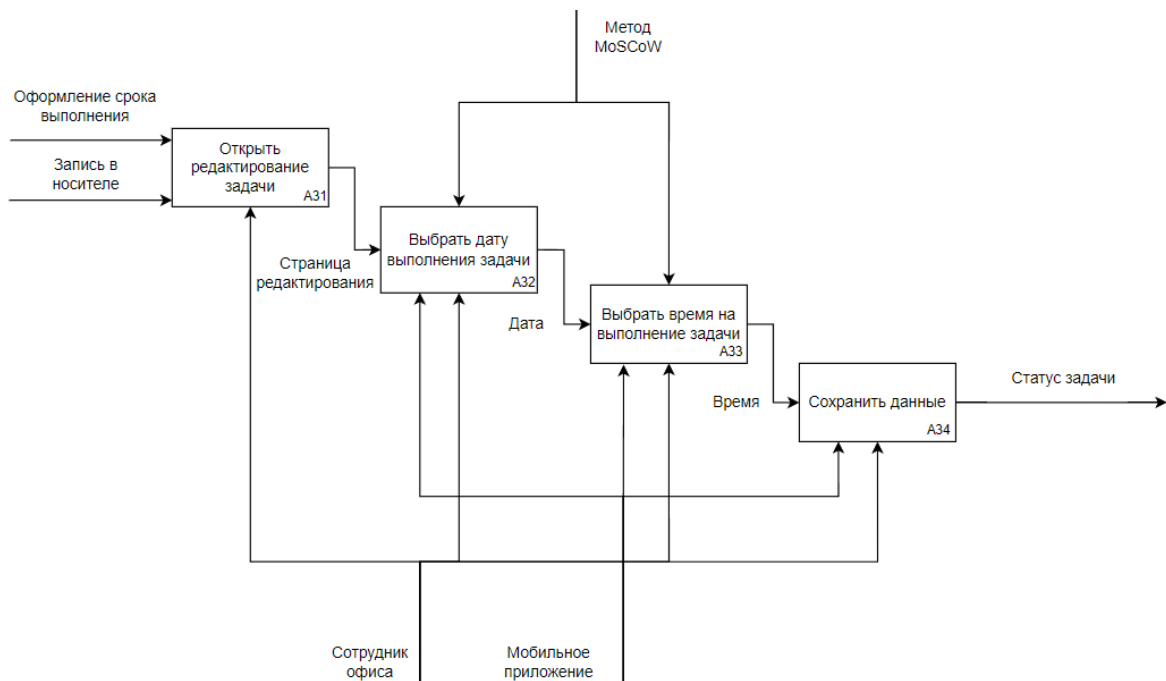


Рисунок 10 - Декомпозиция блока А3

Используя диаграмму потоков данных (DFD), можно исследовать визуальное взаимодействие данных в процессе, изображенном на рисунке 11.

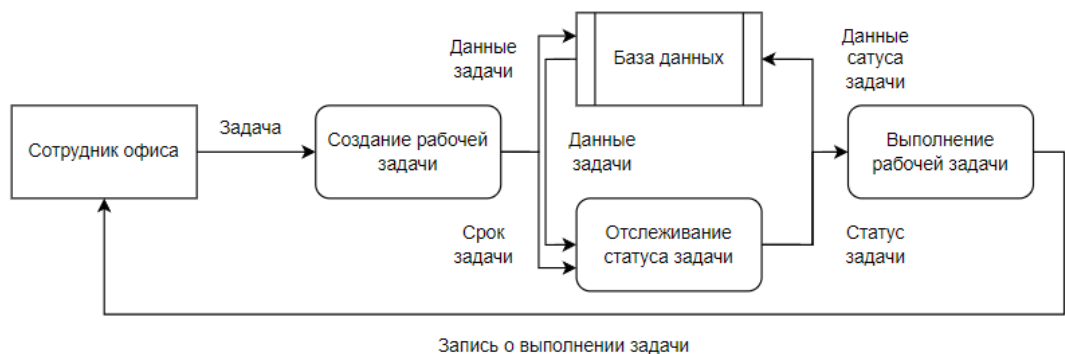


Рисунок 11 - DFD-диаграмма процесса планирования задач

Для идентификации пользователя приложения необходимо предоставить информацию об использовании приложения. Эту информацию можно представить с помощью диаграммы взаимодействия клиентов с приложением, изображенной на рисунке 12. Данная диаграмма наглядно

отображает основной функционал разрабатываемого программного обеспечения, который доступен пользователю.

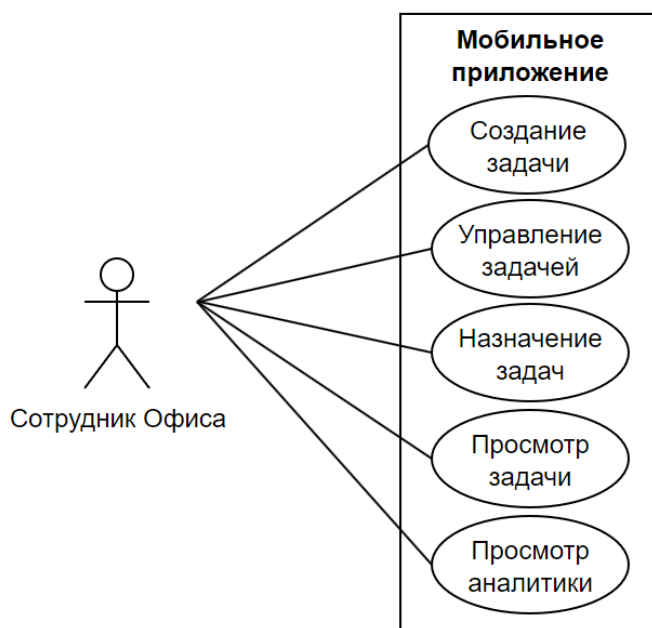


Рисунок 12 - Диаграмма взаимодействия клиентов с приложением.

Данная работа ставит перед собой цель разработки мобильного приложения для эффективного процесса управления корпоративными задачами. Основными требованиями приложения стала классификация требований к программным системам FURPS, изложенными в разделе 1.4.

Выводы по первой главе.

В ходе анализа первой главы можно отметить обозначение краткой характеристики организации, анализ, а также модернизация бизнес процесса организации, обозначение основных требований для приложения, обзор рынка мобильных задач для решения проблемы с эффективностью процесса планирования корпоративных задач.

Глава 2 Проектирование мобильного приложения «Mhand Task»

2.1 Определение платформы для разработки приложения

«В мае 2023 года вышло исследование, которое показало, что россияне почти полностью перешли на смартфоны под управлением Android, судя по статистике за I квартал 2023 г. Доля этих устройств растет огромными темпами за счет постепенного ухода с рынка их главных и, фактически, почти единственных конкурентов – IOS» [5]. Более подробные данные представлены на рисунке 13, где приведен график, отражающий соотношение мобильных устройств с различными операционными системами с мая 2023 года по май 2024 года.

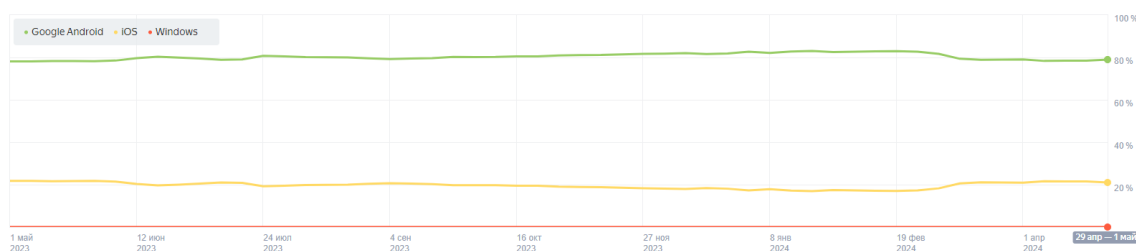


Рисунок 13 – График соотношения мобильных операционных систем.

На рисунках 13 мы можем проанализировать график с сайта Яндекс Радар [6]. Согласно графику, количество устройств с операционной системой iOS более чем в четыре раза меньше, чем устройств с Android. Это указывает на значительное преимущество распространенности смартфонов с операционной системой Android. Исходя из этого, для разработки приложения была выбрана платформа ОС Android, так как она наиболее распространена среди пользователей в Российской Федерации.

2.2 Определение средств для разработки

Понятие «средство разработки программного обеспечения» охватывает набор приемов, методов, методологий и инструментов, применяемых для создания программного кода.

«JetBrains WebStorm — интегрированная среда разработки на JavaScript, CSS & HTML от компании JetBrains, разработанная на основе платформы IntelliJ IDEA» [19].

«WebStorm обеспечивает автодополнение, анализ кода на лету, навигацию по коду, рефакторинг, отладку, и интеграцию с системами управления версиями. Важным преимуществом интегрированной среды разработки WebStorm является работа с проектами» [19].

Так как среда разработки рассчитана именно на JavaScript, приложение будет проектироваться именно на этом языке. Из явных преимуществ можно выделить то, что на JavaScript имеется огромное множество библиотек, которые позволяют закрыть потребность, как во внутренней, так и во внешней разработке оболочки. «Большинство современных интерпретаторов относятся к компилирующему типу, т.е. они сначала максимально приближают JavaScript к машинному коду, оптимизируют его, и лишь затем выполняют. Подобный подход позволил увеличить скорость работы скриптов в несколько раз» [8].

«Справедливо также будет отметить то, что изначально JavaScript предполагает собой именно WEB-разработку, однако с помощью взаимодействия с библиотеками - это решается максимально доступным и технически корректным способом» [20]. «Библиотека Capacitor - это кроссплатформенная нативная среда исполнения для приложений Web Native. На высоком уровне это означает, что Capacitor берет современное веб-приложение, а затем упаковывает его для запуска на iOS, Android и PWA с доступом к функциям родной платформы и элементам управления на уровне ОС. Затем Capacitor выступает в качестве среды выполнения, обеспечивая

взаимодействие между веб-приложением и базовой ОС» [14].

В качестве базы данных было решено использовать MongoDB. MongoDB представляет наиболее популярную на данный момент документо-ориентированную систему управления базами данных. По разным оценкам входит в десятку самых используемых баз данных в мире.

«Если реляционные базы данных хранят строки, то MongoDB хранит документы. В отличие от строк документы могут хранить сложную по структуре информацию. Документ можно представить как хранилище ключей и значений» [4].

Следовательно, для разработки была выбрана среда разработки JetBrains Webstorm, языком программирования определен JavaScript, а для реализации базы данных в мобильном приложении была выбрана библиотека MongoDB.

2.3 Разработка модели данных

«Проектирование модели данных невозможно без определения концептуальной и логической модели данных. Для того, чтобы спроектировать модель данных необходимо построить концептуальную и логическую модель данных. Концептуальная модель базы данных это наглядная диаграмма, нарисованная в принятых обозначениях и подробно показывающая связь между объектами и их характеристиками. Создается концептуальная модель для дальнейшего проектирования базы данных и перевод ее, например, в реляционную базу данных. На концептуальной модели в визуальном удобном виде прописываются связи между объектами данных и их характеристиками» [1]. На рисунке 14 продемонстрирована концептуальная модель базы данных.

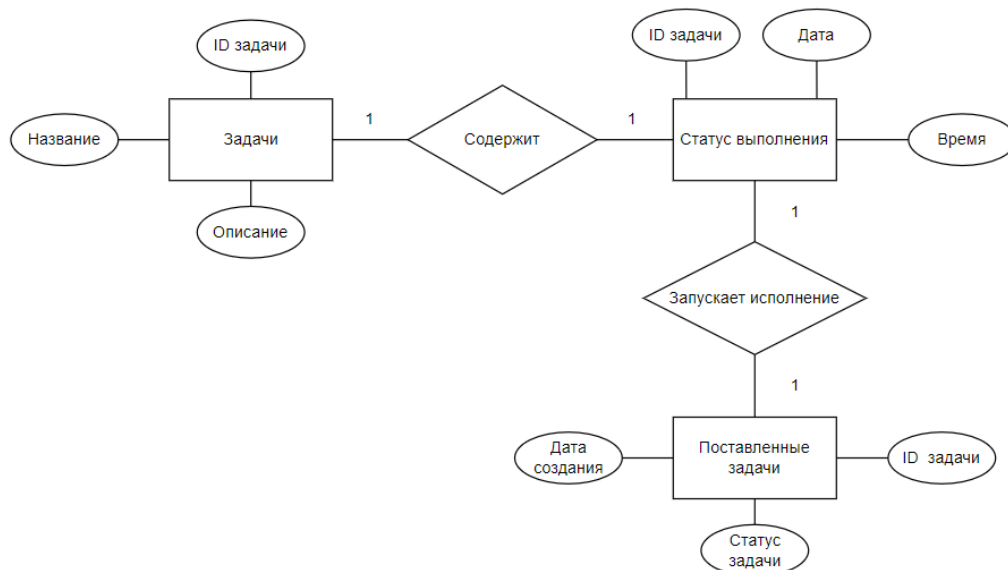


Рисунок 14 - Концептуальная модель данных

Из концептуальной модели данных понятно, что каждая сущность обладает своими атрибутами. Например, задачи содержат атрибуты, такие как ID, описание и название, а сущность статуса выполнения включает в себя атрибуты даты, времени и ID задачи. Еще одна сущность, «поставленные задачи», обладает атрибутами даты создания, статуса задачи и ID. На основе этой концептуальной модели данных строится логическая модель данных, представленная на рисунке 15.

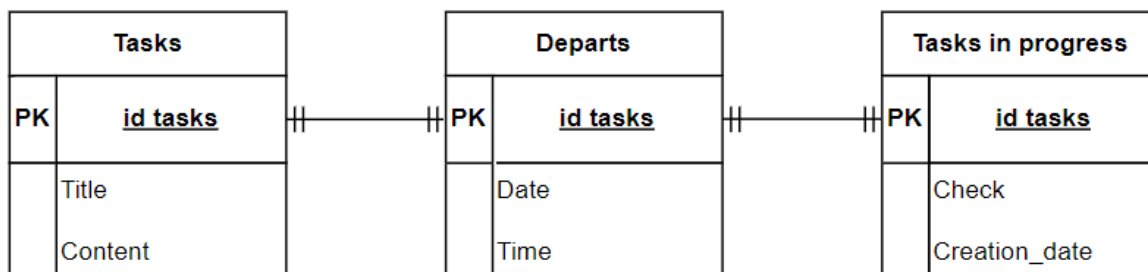


Рисунок 15 - Логическая модель данных.

В логической модели описываются основные понятия предметной области, устанавливается взаимосвязь между ними [9]. На логической модели данных можно видеть, что сущность «tasks» включает атрибуты id_tasks, title и content. Сущность «departs» содержит атрибуты id_tasks, date и time. Сущность «tasks in progress» состоит из атрибутов id_tasks, check и creation_date. Логическая модель данных завершает проектирование базы данных, демонстрируя, что все необходимые объекты для реализации базы данных мобильного приложения «Mhand Task» были учтены

2.4 Структура программного обеспечения

Мобильное приложение «Mhand Task» состоит из программного кода JavaScript, реализуется как web-приложение и с помощью библиотеки Capacitor импортируется как APK приложение для Android. На рисунке 16 представлена схема Capacitor [13].

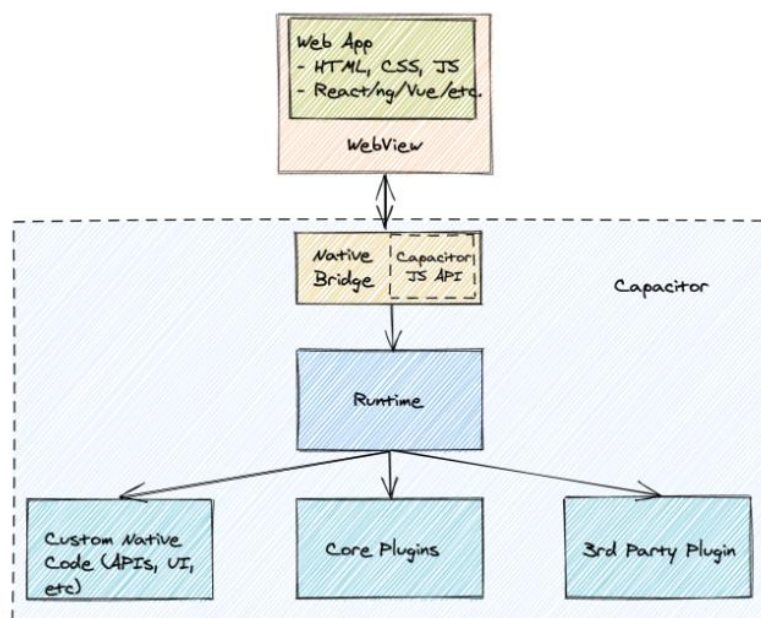


Рисунок 16 - Схема модели Capacitor.

На этой схеме мы видим, что ваше веб-приложение работает внутри

Web View. «Web View - это нативный элемент управления ОС, который обеспечивает оптимизированный экземпляр браузера без хрома. Если вы представите себе типичный веб-браузер, то тонна накладных расходов приходится на хром и работу вокруг фактического кадра просмотра. Web View - это всего лишь один из экземпляров этого фрейма, поэтому он очень легкий» [12].

Архитектура мобильного приложения определена и теперь следует указать диаграмму классов, которая обозначит основные классы и их взаимосвязь, это внесет упрощенное понимание о сложной системе. Диаграмма классов мобильного приложения показана на рисунке 17.

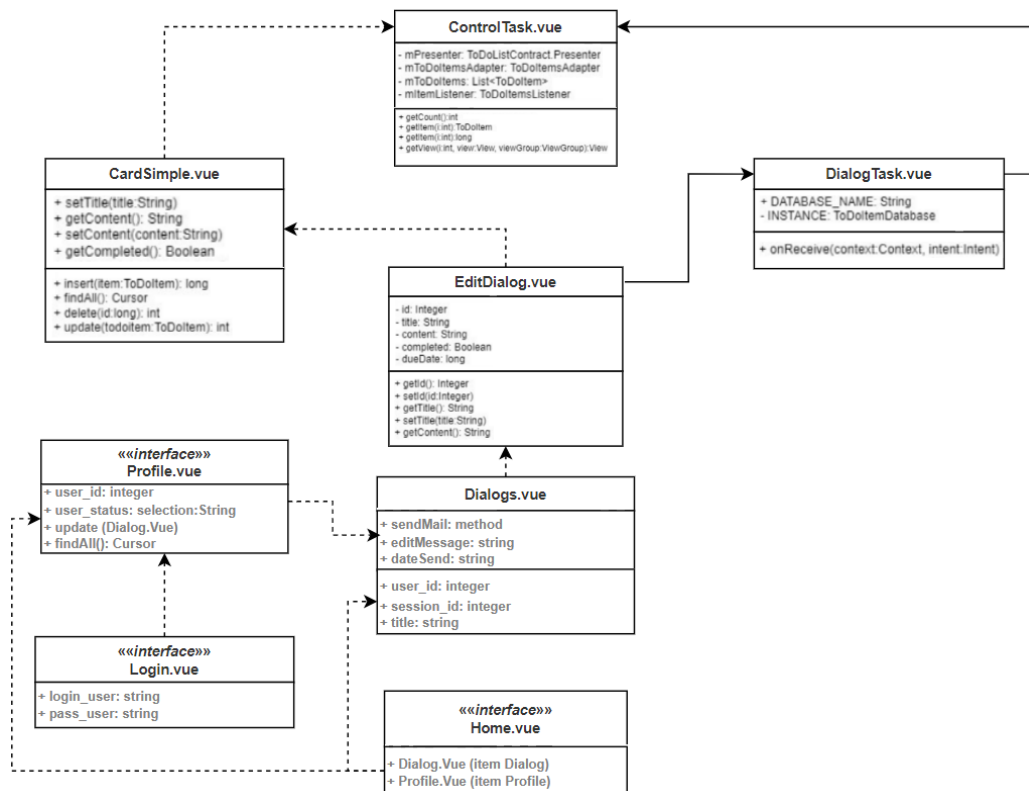


Рисунок 17 – Диаграмма классов мобильного приложения.

Диаграмма развертывания, представленная на рисунке 18, иллюстрирует физическое размещение артефактов на узлах системы.

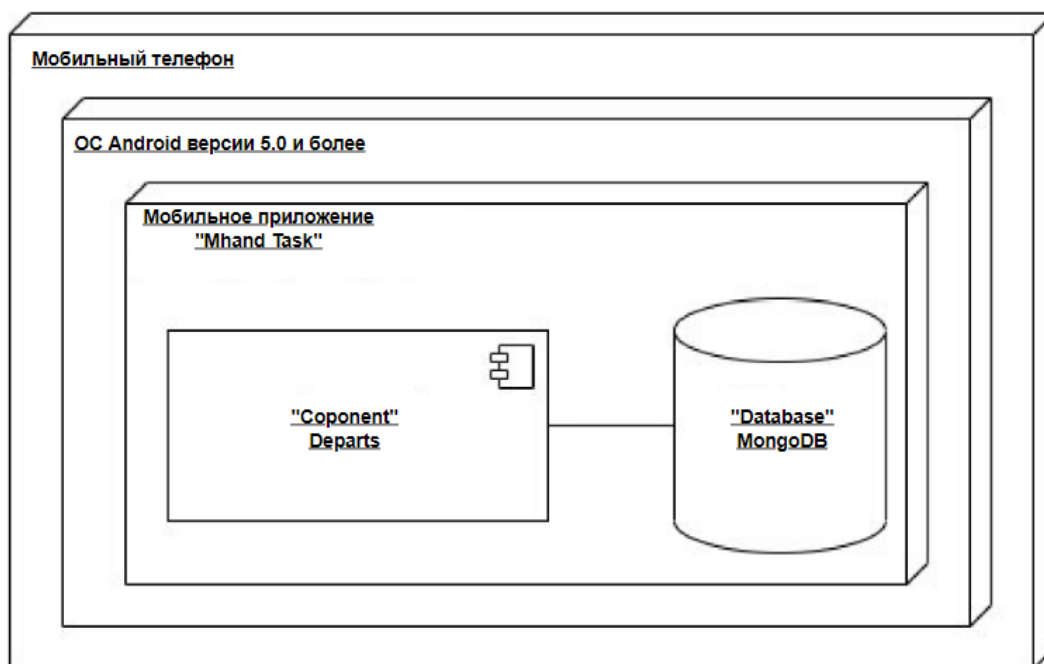


Рисунок 18 - Диаграмма развертывания мобильного приложения.

Рисунок 18, на котором представлена диаграмма развертывания, завершает этап проектирования мобильного приложения. Теперь начинается основной этап — реализация и последующее тестирование приложения.

Выводы по второй главе.

Результатом второй главы можно отметить ознакомление с процессом проектирования приложения «Mhand Task», в которое входит как выбор средства разработки с платформой, так и проектирование модели данных и архитектуры приложения. В разработке эффективного и функционального приложения - данный этап является одним из самых важных.

Глава 3 Реализация мобильного приложения «Mhand Task»

3.1 Сжатое изложение созданного решения

В ходе проектирования и реализации мобильного приложения «Mhand Task» были разработаны девять классов и три интерфейса. Классы хранятся в трех папках: components, modules и panel, это классы:

- ApplicationPage.vue (Реализация редактирования профиля пользователя);
- CardSimple.vue (Реализация интерфейса приложения);
- ControlTask.vue (Реализация аналитики и статуса задач);
- Dialogs.vue (Реализация создания задачи);
- DialogTask.vue (Реализация окна с выводом параметров задачи) (Приложение А);
- EditDialog.vue (Реализация управления задачами);
- MessageComponent.vue (Реализация комментариев и описаний к задачам);
- departs.js (Реализация обмена данными с базой данных);
- main.js (Основное тело программного кода);
- Home.vue (Интерфейс домашней страницы);
- Profile.vue (Интерфейс профиля пользователя);
- Login.vue (Реализация окна логина и регистрации).

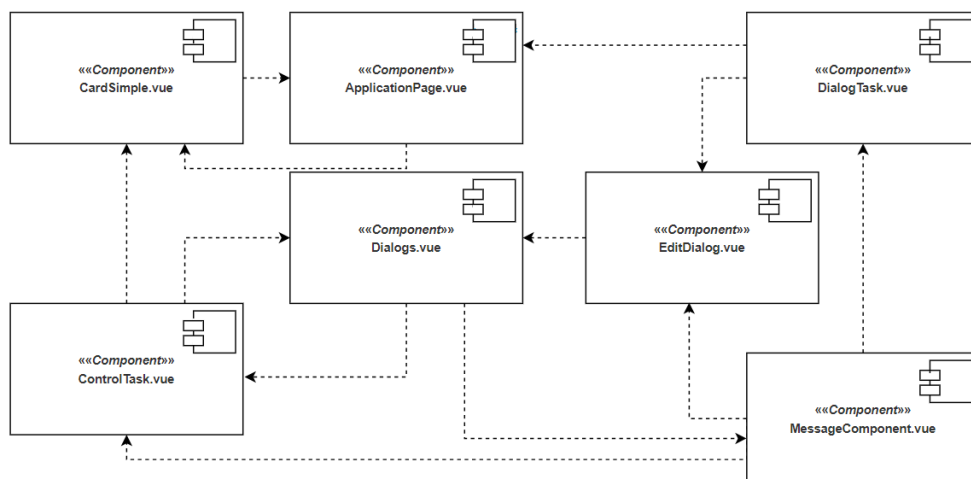


Рисунок 19 – Структура классов мобильного приложения.

Для наглядной демонстрации на рисунке 19 показана структура классов мобильного приложения «Mhand Task».

3.2 Реализация постановки задачи

Рассматривая в этом пункте часть кода, представляет собой компонент Vue.js, который отвечает за процесс создания задачи в приложении.

Подробно разберем, как этот процесс реализован:

В Template компонента определены различные элементы пользовательского интерфейса (UI), необходимые для создания задачи. Кнопка «Поставить задачу» (w-button) инициирует открытие диалогового окна для ввода информации о задаче. В диалоговом окне (w-dialog) предусмотрены различные поля для ввода информации о задаче: краткое описание, подробное описание, дата дедлайна, выбор исполнителей и добавление контролирующих лиц. Используются также компоненты для ввода текста (w-input, w-textarea), выбора даты (w-input type="date"), и выбора пользователей из списка (w-table с чекбоксами w-checkbox). Кнопка «Поставить задачу» завершает процесс создания задачи и закрывает диалоговое окно, это можно увидеть на рисунке 20.

```

1 <template>
2 <w-button color="primary" @click="addTaskDialog = true">
3   Поставить задачу
4 </w-button>
5 <w-dialog v-model="addTaskDialog" :width="width">
6   <template #title>
7     <w-icon>mdi mdi-pen-plus</w-icon> <span>Постановка задачи</span>
8     <div class="spacer"></div>
9     <w-button lg bg-color="red" color="white" @click="addTaskDialog = false"><w-icon>mdi mdi-close</w-icon></w-button>
10  </template>
11  <template #default>
12    <div class="contentDialog">
13      <h3>Выберите кому хотите поставить задачу</h3>
14      <w-table :headers="tableHeaders" :items="users2">
15        <template #item-cell.checkbox="{item, label, header, index}">
16          <div class="text-center">
17            <w-checkbox v-model="item.checkbox"></w-checkbox>
18          </div>
19        </template>
20      </w-table>
21      <w-input class="mt2" v-model="task" :color="taskColor" :label-color="taskColor" outline>Краткое описание задачи</w-input>
22      <w-textarea class="mt2" outline v-model="taskDescription" label="Опишите задачу подробнее"></w-textarea>
23      <div class="mt2" style="...">
24        <w-input type="date" outline label-position="left" :color="deadlineColor" :label-color="deadlineColor" v-model="date">
25          Дата дедлайна
26        </w-input>
27      </div>
28      <div style="...">
29        <w-checkbox v-model="visitor" label="Добавить контролирующего/наблюдающего за задачей"></w-checkbox>
30        <div v-if="visitor">
31          <w-table :headers="tableHeaders" :items="visitorUsers">
32            <template #item-cell.checkbox="{item, label, header, index}">
33              <div class="text-center">
34                <w-checkbox v-model="item.checkbox"></w-checkbox>
35              </div>
36            </template>
37          </w-table>
38        </div>
39      </div>
40    </div>
41  </template>

```

Рисунок 20 – Реализация блока Template класса Dialogs.vue

В блоке скрипта определены данные, методы и вычисляемые свойства компонента. В данных (data) хранится состояние компонента, такое как введенная пользователем информация о задаче, выбранные пользователи и дата дедлайна. Методы (methods) представляют функциональность компонента, включая валидацию данных, отправку запросов на сервер для создания задачи, отправку уведомлений и обработку событий взаимодействия пользователя с интерфейсом. Используются также методы mapGetters и mapActions для доступа к данным из хранилища Vuex и выполнения асинхронных операций, это показано на рисунке 21.


```

121   computed: {
122     ...mapGetters(["users", "userCrm"]),
123   },
124   async mounted() {
125     await this.getUsers()
126     console.log(this.users)
127     this.users2 = []
128     for (let i = 0; i < this.users.length; i++) {
129       console.log(this.users[i].firstname)
130       let name = this.users[i].firstname + ' ' + this.users[i].lastname
131       let email = this.users[i].workMail
132       let position = this.users[i].jobTitle
133       let depart = this.users[i].depart.name
134       let id = this.users[i].idProfile
135       let tgChat = this.users[i].tgChat
136       this.users2.push({name: name, email: email, position: position, depart: depart, id: id, checkbox: false, tgChat: tgChat})
137     }
138     for (let i = 0; i < this.users.length; i++) {
139       let name = this.users[i].firstname + ' ' + this.users[i].lastname
140       let email = this.users[i].workMail
141       let position = this.users[i].jobTitle
142       let depart = this.users[i].depart.name
143       let id = this.users[i].idProfile
144       let tgChat = this.users[i].tgChat
145       this.visitorUsers.push({name: name, email: email, position: position, depart: depart, id: id, checkbox: false, tgChat: tgChat})
146     }
147
148     await this.getUserCrm(this.userId)
149     console.log('Cons iz diag', this.userCrm)
150
151   },
152   methods: {
153     ...mapActions(["getUsers", "getUserCrm"]),
154     saveDeadline() {
155       this.date = this.datePick.toLocaleString().substr(0, 10)
156     },
157     async logDeparts() {
158       console.log(this.departs)
159       console.log(this.users2)
160       let userArr = []
161       for (let i = 0; i < this.users2.length; i++) {
162         if (this.users2[i].checkbox == true) {
163           userArr.push(this.users2[i])
164         } else continue
165       }
166       let visitorArr = []
167       for (let i = 0; i < this.visitorUsers.length; i++) {
168         if (this.visitorUsers[i].checkbox == true) {
169           visitorArr.push({id: this.visitorUsers[i].id, name: this.visitorUsers[i].name})
170         } else continue
171       }
172       console.log(userArr)
173       let count = 0
174       console.log(this.date)
175     }
176   }

```

Рисунок 21 – Реализация блока Script класса Dialogs.vue

В блоке Style определены CSS-стили, которые применяются к элементам внутри компонента. Стили используются для определения внешнего вида диалогового окна и его содержимого, а также для обеспечения адаптивности интерфейса на различных устройствах. Таким образом, данный компонент обеспечивает пользователю удобный интерфейс для создания задачи в мобильном приложении, предоставляя возможность ввода необходимой информации и взаимодействия с приложением для управления задачами, что показано на рисунке 22.

```

258   <style scoped>
259   .dialogClass{
260     width: 1000px;
261     justify-content: center;
262   }
263   .contentDialog{
264     overflow: scroll;
265   }
266   @media screen and (max-width: 768px){
267     .contentDialog{
268       height: 60vh;
269     }
270   }
271
272 </style>

```

Рисунок 22 – Реализация блока Style класса Dialogs.vue

В результате получилось разработать функционал создания задачи пользователем, назначение ответственного, а также реализован интерфейс.

3.3 Реализация редакции личного профиля пользователем

Этот класс представляет компонент Vue.js, который отвечает за процесс редактирования профиля пользователя в мобильном приложении. Подробно разберем, как этот процесс реализован: В Template компонента определены различные элементы пользовательского интерфейса (UI) для редактирования данных профиля пользователя. Используются компоненты w-input для ввода текста и w-button для кнопки «Подтвердить». Для загрузки аватара пользователя используется блок .box, содержащий изображение профиля и скрытый input типа file. Пользователь может нажать на изображение для выбора нового файла изображения, это продемонстрировано на рисунке 23.

```

1 <template>
2 <div v-if="type === 'departs'" >
3   <div>
4     <w-card>
5       <template #title>
6         Подтверждение отдела
7       </template>
8       <template #default>
9         <w-table :headers="label" :items="items" @row-click="onRowClick"> </w-table>
10
11
12       </template>
13     </w-card>
14     <w-dialog persistent v-model="departsDialog" transition="bounce" width="400">
15       <template #title>
16         {{cardContent.name}}
17       </template>
18       <template #default>
19         <w-flex class="row justify-start align-center">
20           <h3>Выбрали отдел:</h3><span>{{cardContent.depart}}</span>
21         </w-flex>
22         <w-flex class="row justify-start align-center">
23           <w-checkbox v-model="changeDepart" name="checkbox1">
24             <span>Сменить отдел</span>
25           </w-checkbox>
26           <w-select class="ml2" v-if="changeDepart" v-model="selectedIndexDepart" outline :items="departs2" item-value-key="_id" item-label-key="name">
27             Выберите отдел
28           </w-select>
29         </w-flex>
30       </template>
31
32       <template #actions>
33
34         <w-flex class="row justify-end">
35           <w-button class="error--bg mr2" @click="departsDialog=false">Отклонить</w-button>
36           <w-button class="success--bg" @click="changeDepartFunc();departsDialog=false">Подтвердить</w-button>
37         </w-flex>
38       </template>
39     </w-dialog>
40   </div>
41 </div>
42 </div>
43 </div>
44
45 </template>

```

Рисунок 23 – Реализация блока Template класса ApplicationPage.vue

В блоке скрипта определены данные, методы и вычисляемые свойства компонента. В данных (data) хранится текущая информация о профиле пользователя и выбранный файл изображения. В методах (methods) реализована логика обновления профиля пользователя: В зависимости от того, был ли выбран новый файл изображения, отправляется либо только текстовая информация о профиле, либо текстовая информация вместе с загруженным изображением. Метод `getFormData` отправляет выбранный пользователем файл на сервер для загрузки на удаленное хранилище (например, Amazon S3) и получает обратно ссылку на загруженное изображение. Метод `uploadFile` вызывается при выборе нового файла изображения и сохраняет его в data для отображения в UI и отображен на рисунке 24.

```

99   onClick(item, index){
100     let data = item
101     this.cardContent.name = data.item.name
102     this.cardContent.depart = data.item.depart
103     this.cardContent.id = data.item.id
104     this.departsDialog = true
105     console.log(this.cardContent)
106   },
107   async changeDepartFunc(){
108     if(this.changeDepart == false){
109       console.log("Id Profile",this.cardContent.id)
110       let response = await fetch( input: process.env.VUE_APP_BACK_HTTP+'crm/acceptDepart', init: {
111         method:'POST',
112         headers:{
113           Accept: "application/json",
114           "Content-Type": "application/json",
115         },
116         body:JSON.stringify( value: {
117           id:this.cardContent.id
118         })
119       })
120       if (response.status == 200){
121         await this.fetchData()
122       }else{
123         alert('Ошибка')
124       }
125     }else{
126       console.log("Id Depart",this.selectedIndexDepart)
127       let tempDepart = null
128       for(let i=0; i<this.departs2.length;i++){
129         if(this.departs2[i]._id == this.selectedIndexDepart){
130           tempDepart = this.departs2[i]
131           break
132         }else continue
133       }
134       let response = await fetch( input: process.env.VUE_APP_BACK_HTTP+'crm/changeDepartAdmin', init: {
135         method:'POST',
136         headers:{
137           Accept: "application/json",
138           "Content-Type": "application/json",
139         },
140         body:JSON.stringify( value: {
141           id:this.cardContent.id,
142           departId:tempDepart._id,
143           departName:tempDepart.name,
144         })
145       })
146       if (response.status == 200){
147         await this.fetchData()
148       }else{

```

Рисунок 24 – Реализация блока Script класса Dialogs.vue

В этом блоке определены CSS-стили, задающие внешний вид элементов UI компонента. Стили позволяют красиво отобразить изображение профиля и добавить эффект наведения для замены фотографии. Таким образом, данный компонент обеспечивает пользователю удобный интерфейс для редактирования профиля в мобильном приложении.

3.4. Тестирование мобильного приложения «Mhand Task»

Функциональное тестирование играет ключевую роль в разработке программного обеспечения, помогая выявить и устранить возможные ошибки и обеспечивая высокое качество работы приложения. Его цель заключается в проверке корректного выполнения заданных функций программы в различных условиях, которые важны для пользователей.

Для проведения функционального тестирования мы определим четыре тест-кейса, каждый из которых будет иметь следующую структуру:

- цель: описание того, что именно мы пытаемся протестировать;
- ожидаемый результат: какой результат мы ожидаем получить от тестирования;
- входные данные: какие данные или условия будут использоваться для проведения теста;
- процедура тестирования: шаги, которые мы выполним, чтобы протестировать функцию;
- полученный результат: результат тестирования и любые обнаруженные проблемы или наблюдения.

Такой подход поможет нам структурировать и провести тестирование мобильного приложения эффективно, обеспечивая проверку его функциональности в различных сценариях использования.

Исходя из описанного выше, необходимо сформировать таблицу с результатом тестирования. В таблице 3 представлены результаты тестирования функционала приложения «Mhand Task».

Таблица 3 – Тестирование функционала приложения «Mhand Task»

Цель	Ожидания	Исходные данные	Процесс проведения тестирования	Результат
Заведение корпоративной задачи в приложении.	Заведение задачи на отдельной вкладке для создания корпоративной задачи.	Пользователь переходит во вкладку создания задачи.	Пользователь заполняет наименование задачи, поля для конфигурации задачи, описание и нажимает кнопку «Сохранить».	Результат оправдал ожидания
Управление и редактирование задачей сотрудником.	На задаче возможно самостоятельно но редактировать статус, исполнителя и время выполнения.	Пользователь переходит в окне редактирования задачи.	Пользователь вводит дату и время выполнения задачи, статус и делегирует задачу коллеге, нажимает кнопку «Сохранить».	Результат оправдал ожидания
Редактирования профиля пользователя.	Пользователь имеет возможность самостоятельно но изменить свои данные в личной карте профиля, поменять почту, пароль.	Пользователь находится во вкладке управления профилем.	Пользователь нажимает кнопку изменить возле своего ФИО, вводит новые данные, меняет аватар. почту, номер телефона, нажимает кнопку «Сохранить».	Результат оправдал ожидания
Проверка функций удаления задачи	После выбора функции удаления задачи на главном экране приложения задача будет удалена	Пользователь находится в окне редактирования задачи.	При нажатии кнопки «Удалить» пользователь видит появившееся окно подтверждением удаления, нажимает «Подтверждаю», задача удаляется.	Результат оправдал ожидания

В первой строке таблицы проверяется создание задачи. Чтобы это проверить, необходимо перейти во вкладку создания задачи, нажать кнопку «Создать»», после заполнить данные по задаче и нажать кнопку сохранить. По итогу задача была создана и отображена в списке задач пользователя, из этого следует, что функция работает исправно.

Во второй строке таблицы происходит тестирование редактирования и управления задачами. Для этого мы переходим в окно редактирования задачи, далее меняем данные на необходимые нам, добавляем исполнителя, меняем статус задачи и меняем время выполнения и дату, нажимается кнопка сохранения. Полученный результат совпал с ожидаемым и данные успешно сохранились в задаче и были обновлены в описании задачи.

В третьей строке проверяется функция редактирования карточки профиля пользователя. Для этого необходимо перейти в свой профиль, нажать кнопку редактирования, заполнить новые данные ФИО, возраста, электронной почты и номера телефона, выбрать новый аватар и сохранить изменения. Как итог мы получили обновленный профиль, что подтверждает успешное исполнение функции.

В четвертой строке проверяется функция удаления. Для ее проверки необходимо перейти во вкладку с имеющимися задачами, выбрать необходимую и нажать кнопку «Удалить». Результат был согласно ожиданиям, при нажатии подтверждения удаления задачи - задача удаляется из списка и не отображается у пользователей.

Процесс тестирования функционала разработанного продукта продемонстрировал успех в итоговом результате процесса разработки функциональных действий с задачами и приложением в целом.

Выводы по третьей главе.

Третья глава наглядно продемонстрировала процесс реализации мобильного приложения «Mhand Task», включая разработку элементов графической части проекта, а также основного функционала, было проведено тестирования на работоспособность приложения.

Заключение

В процессе выполнения выпускной квалификационной работы было спроектировано и разработано мобильное приложение для планирования корпоративных задач «Mhand Task» под операционную систему Android. Была объявлена цель данной работы - проектирование и разработка мобильного приложения для управления процессом планирования корпоративных задач.

В ходе выполнения работы были достигнуты следующие результаты:

- был осуществлен анализ бизнес-процессов компании, исходя из которого были обозначены требования к функционалу мобильного приложения;
- было спроектировано мобильное приложение, включая в себя выбор средства разработки, модели данных, определение архитектуры мобильного приложения;
- разработано мобильное приложение с обширным функционалом, включающим в себя управление задачами и профилем пользователя;
- осуществлено тестирование приложения, показавшее положительный результат работы функционала приложения и удобство во взаимодействии с пользователем.

По итогу работы можно отметить, что было спроектировано и реализовано полноценное мобильное приложение «Mhand Task», предоставляющее возможность эффективного управления корпоративным пространством для его пользователей.

Однако у приложения «Mhand Task» есть потенциал для дальнейшего развития и добавления новых функций. Среди них могут быть:

Добавление календаря для организации совместных мероприятий или конференций.

Добавление документации сотрудникам в отдельную вкладку, это

может быть документация с обучением, либо же внутренние документы компании.

Добавление общего чата сотрудников в приложение, для обсуждения задач и новостей внутри компании.

Помимо этого, важно отметить, что в ходе разработки приложения были учтены особенности интерфейса пользователя и обеспечена его простота в использовании. Также были проведены тщательные тестирования приложения на различных устройствах и сценариях использования для гарантированного качества и надежности. Полученные результаты подтверждают успешную реализацию задач и функциональности приложения, а также показывают его потенциал для дальнейшего улучшения и развития.

Дальнейшая работа по приложению определена в обновлениях и улучшениях приложения, внедрение функционала согласованных с пользователями и руководством.

Список используемой литературы

1. Андиева Е. Ю., Сидоренко В. С. Анализ критериев выбора case–средств //Редакционная коллегия. – 2015. – С. 164.
2. Белоусов А. В., Толкачева О. С., Баева В. Г. Разработка концептуальной модели базы данных информационно-справочных СППР //Системный анализ в проектировании и управлении: сборник научных трудов XXI Международной научной практической конференции. – 2017. – С. 298.
3. Быстренина И.Е., Case-средства анализа и проектирования информационных систем в содержании обучения бакалавров прикладной информатики // Новая наука: опыт, традиции, инновации. - 2022. - С. 4.
4. Введение в MongoDB. [Электронный ресурс] / Режим доступа: URL: <https://metanit.com/nosql/mongodb/1.1.php> (дата обращения 01.05.2024).
5. В России стремительно растет доля смартфонов на ОС Android. [Электронный ресурс] / Режим доступа: URL: <https://www.tssonline.ru/news/v-rossii-stramitelno-rastiot-dolia-smrtfonov-na-os-android> (дата обращения 01.05.2024).
6. Мобильные ОС в России. [Электронный ресурс] / Режим доступа: URL: <https://radar.yandex.ru/mobile> (дата обращения 01.05.2024).
7. ООО «Море Сторе». [Электронный ресурс] / Режим доступа: URL: <https://www.tinkoff.ru/business/contractor/legal/1217700613635/> (дата обращения 01.05.2024).
8. МЦИИ Омега Сайнс, Информационное обеспечение науки как двигатель прогресса // Международная научно-практическая конференция. - 2020. - С. 6.
9. Черкесова К.И., Об особенностях проектирования логической и физической модели базы данных портала образовательного учреждения // Наука. Искусство. Культура. - 2015. - С. 3.
- 10.App Store. [Электронный ресурс] / Режим доступа: URL: https://ru.wikipedia.org/wiki/App_Store (дата обращения 01.05.2024).

11. CASE-средства. [Электронный ресурс] / Режим доступа: URL: https://www.kpms.ru/Automatization/CASE_tools.htm (дата обращения 01.05.2024).
12. Capacitor. [Электронный ресурс] / Режим доступа: URL: <https://capacitorjs.jp/blog/how-capacitor-works> (дата обращения 01.05.2024).
13. Capacitor JS. [Электронный ресурс] / Режим доступа: URL: <https://github.com/ionic-team/capacitor> (дата обращения 01.05.2024).
14. Capacitor JS Application. [Электронный ресурс] / Режим доступа: URL: <https://medium.com/@SmileFX/a-complete-guide-building-a-capacitorjs-application-using-pure-javascript-and-webpack-37d00f11720d> (дата обращения 01.05.2024).
15. Diagrams.Net. [Электронный ресурс] / Режим доступа: URL: <https://en.wikipedia.org/wiki/Diagrams.net> (дата обращения 01.05.2024).
16. FURPS. [Электронный ресурс] / Режим доступа: URL: <https://ru.wikipedia.org/wiki/FURPS> (дата обращения 01.05.2024).
17. Google Play. [Электронный ресурс] / Режим доступа: URL: https://ru.wikipedia.org/wiki/Google_Play (дата обращения 01.05.2024).
18. MoSCoW method. [Электронный ресурс] / Режим доступа: URL: https://en.wikipedia.org/wiki/MoSCoW_method (дата обращения 01.05.2024).
19. WebStorm. [Электронный ресурс] / Режим доступа: URL: <https://ru.wikipedia.org/wiki/WebStorm> (дата обращения 01.05.2024).
20. WebStorm external libraries. [Электронный ресурс] / Режим доступа: URL: <https://stackoverflow.com/questions/27850763/webstorm-external-libraries-explanation>

Приложение А

Программный код класса DialogTask.vue

```
<script>
import {mapActions, mapGetters} from "vuex";
import MessageComponent from "../MessageComponent";

export default {
  name: "DialogTask",
  components:{
    MessageComponent
  },
  props:{
    task:{
      value:Object,
      default(){
        return {}
      }
    },
    taskControl:{
      value:Boolean,
      default() {
        return false;
      }
    }
  },
  data: ()=>({
    selected:[],
    progress:50,
    textComment:'',
    addVisitor:false,
    text:'dadadsasad mjadni abd akshdb habd sajhhdh kjksbd ksndk a',
    items: [
      { label: 'Choice 1', value: 1 },
      { label: 'Choice 2', value: 2 },
      { label: 'Choice 3', value: 3 }
    ],
    header:[
      {label:"Выбрать", key:"checkbox"},
      {label: "ФИО", key:"name"},
      {label:"Отдел", key:"departName"}
    ]
  })),
```

Продолжение приложения А

```
computed:{
  ...mapGetters(["users","userCrm"])
},
async mounted() {
  await this.getUsers()
  console.log(this.users)
  console.log(this.$props.task)
  for(let i=0;i<this.users.length;i++){
    this.users[i].checkbox = false
    this.users[i].name = this.users[i].firstname + ' ' + this.users[i].lastname
    this.users[i].departName= this.users[i].depart.name
  }
  if(this.$props.task.visitor){
    for(let i=0; i< this.$props.task.visitor.length; i++){
      for(let j=0;j<this.users.length; j++){
        if(this.$props.task.visitor[i].id == this.users[j].idProfile){
          this.users[j].checkbox = true
        }
      }
    }
  }
},
methods:{
  ...mapActions(["getUsers","getUserCrm"]),
  async addComment(){
    let response = await fetch(process.env.VUE_APP_BACK_HTTP+'crm/addComment/'+this.$props.task._id,{
      method:'PATCH',
      headers:{
        Accept: "application/json",
        "Content-Type": "application/json",
      },
      body:JSON.stringify({
        id:this.userCrm.idProfile,
        avatar:this.userCrm.linkAvatar,
        time:new Date(),
        name:this.userCrm.firstname + ' '+this.userCrm.lastname,
        text:this.textComment,
      })
    })
    if(response.status == 200){
      alert('Комментарий для задачи добавлен')
      window.location.reload()
    }else alert('Ошибка')
  },
}
```

Продолжение приложения А

```
async addVisitors(task){
  let arr = []
  for(let i = 0; i<this.users.length; i++){
    if(this.users[i].checkbox){
      let name = this.users[i].firstname + ' ' + this.users[i].lastname
      arr.push({id:this.users[i].idProfile, name:name, tg:this.users[i].tgChat})
    }
  }
  console.log(task)
  console.log(arr)

  let response = await fetch(process.env.VUE_APP_BACK_HTTP+'crm/taskPushVisitor', {
    method:'PATCH',
    headers:{
      Accept: "application/json",
      "Content-Type": "application/json",
    },
    body: JSON.stringify({
      idTask: task,
      arrVisitors: arr
    })
  })
  if (response.status == 200){
    this.addVisitor = false
    alert('Контролирующие добавлены')
    for(let i=0; i<arr.length; i++ ){
      console.log(arr[i])
      let message = 'Вас добавили контролирующем за задачей\n' +
        'Задача: '+this.$props.task.nameTask+'\n' +
        'Задача от: '+this.$props.task.author.name+'\n' +
        'Задача для: '+this.$props.task.userName+'\n' +
        'Описание: '+this.$props.task.discriptionTask+'\n' +
        'Чтобы следить за ходом выполнения задачи, перейдите на сайт http://185.82.24.227'
      let url = encodeURI('https://api.telegram.org/bot5058763471:AAE5IYPYmQJU0h4dr25_EZfngyUoQ1Ck1j0/sendmessage?chat_id='+arr[i].tg+'&text='+message)
      let response = await fetch(url)
    }
  }
}
```

```
},
checkType(id){
  if(id == this.userCrm.idProfile){
    return false
  }else{
    return true
  }
},
checkFlex(id){
  if(id == this.userCrm.idProfile){
    return "padding: 10px; display: flex; justify-content: end"
  }else{
    return "padding: 10px; display: flex; justify-content: start"
  }
}
}
}
</script>
```