

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 / специальности)

Мобильные и сетевые технологии
(направленность (профиль) / специализация)

ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА (БАКАЛАВРСКАЯ РАБОТА)

на тему Разработка мобильного приложения для учёта сотрудником Центра цифровых компетенций статуса движения документов

Обучающийся

И.В. Приходько

(Инициалы Фамилия)

(личная подпись)

Руководитель

канд. пед. наук, доцент, Т.А. Агошкова

(ученая степень (при наличии), ученое звание (при наличии), Инициалы Фамилия)

Консультант

канд. пед. наук, доцент, С.А. Гудкова

(ученая степень (при наличии), ученое звание (при наличии), Инициалы Фамилия)

Тольятти 2024

Аннотация

Тема: «Разработка мобильного приложения для учёта сотрудником Центра цифровых компетенций статуса движения документов».

Выпускная квалификационная работа состоит из пояснительной записки на 42 страницах, введения, трех глав, заключения, 5 таблиц, 18 рисунков, списка литературы из 20 источников, включая 9 зарубежных источников и 1 приложения.

Основной сущностью данной работы является проектирование и создание мобильного приложения, которое имеет функционал, позволяющий отслеживание статуса движения документов.

Во введении описывается актуальность данной бакалаврской работы, которая заключается в возможности увеличить эффективность и скорость работы сотрудника Центра цифровых компетенций с помощью мобильного приложения.

В первой главе проводится анализ предметной области, описание бизнес-процессов, анализ бизнес-модели, анализ аналогов мобильных приложений и постановка задачи на разработку.

Во второй главе проводится проектирование мобильного приложения, выбор средств разработки, языка программирования, создание диаграммы вариантов использования и моделирования базы данных.

В третьей главе приведена реализация мобильного приложения, а также проведено тестирование разработанного мобильного приложения.

Заключение состоит из перечня выполненных задач в рамках данной бакалаврской работы и описание полученного результата.

Основным результатом данной выпускной квалификационной работы является готовое приложение, которое способно выполнять установленные функции по отслеживанию статуса движения документов.

Abstract

The title of the graduation work is « Development of a mobile application for accounting by an employee of the Center for Digital Competencies of the status of document movement».

The senior paper consists of an introduction, three parts, a conclusion, tables, list of references including foreign sources and the graphic part on 1 A1 sheet.

The key issue of the thesis is the design and creation of a mobile application that has functionality that allows tracking the status of document movement. We touch upon the problem of tracking mistake which can occur with many different documents going to different places and details about the status of those documents not being properly recorded anywhere.

The aim of the work is to give some information about the development of a mobile application for accounting by an employee of the Center for Digital Competencies of the status of document movement.

The graduation work may be divided into several logically connected parts which are: analysis of the Center for Digital Competencies work; analysis of the existing similar apps; setting a task for future development; justification of the choice of tools for development; development of the app; demonstration; testing.

Finally, we present the work on the successful development of the app, the use of which can help workers of the Center for Digital Competencies track status of documents and better manage their time for other tasks.

In conclusion we'd like to stress this work is relevant in solving the problem of tracking status of documents as well as helping manage work time better and avoid human error in the matter of moving documents between departments.

Оглавление

Введение.....	5
Глава 1 Анализ предметной области.....	7
1.1 Характеристика деятельности Центра цифровых компетенций....	7
1.2 Выбор case-средств для описания бизнес-процессов.....	8
1.3 Анализ модели бизнес-процесса	11
1.4 Обзор и анализ существующих аналогов мобильных приложений	13
1.5 Постановка задачи	15
Глава 2 Проектирование мобильного приложения	18
2.1 Выбор средств разработки	18
2.2 Выбор языка программирования.....	20
2.3. Диаграмма вариантов использования	22
2.4 Моделирование базы данных	24
Глава 3 Реализация мобильного приложения	26
3.1 Разработка мобильного приложения	26
3.2 Реализация базы данных	29
3.3 Реализация основного функционала.....	31
3.4 Тестирование приложения.....	35
Заключение	38
Список используемой литературы	40
Приложение А Блок-схема создания новой записи.....	42

Введение

С развитием смартфонов все больше и больше развиваются и мобильные приложения. Почти для любого занятия можно найти приложение, которое упростит или улучшит процесс выполнения задачи. Есть приложения для заказа еды, отслеживания новостей и погоды и много других. Также планирование и организация времени стали намного проще с появлением смартфонов, а время — это очень ценный ресурс в жизни человека.

Актуальность данной работы заключается в необходимости улучшения отслеживания трудовых задач в Центре Цифровых Компетенций Тольяттинского государственного университета, т.к. в данный момент такой инструмент отсутствует, что делает возможности сотрудников более ограниченными и может привести к непредвиденным ошибкам.

Объектом исследования данной работы является процесс отслеживания передачи документов между отделами.

Предметом исследования является автоматизация отслеживания процесса передачи документов.

Целью данной работы является разработка мобильного программного обеспечения, которое позволит пользователю отслеживать перемещение документов, заполняя список задач, что приведет к повышению трудоспособности.

Для достижения поставленной цели будут выполнены следующие задачи:

- анализ предметной области;
- анализ бизнес-процессов Центра Цифровых Компетенций;
- обзор существующих на рынке решений;
- определить требования к будущему приложению;
- обосновать выбор средств для разработки;
- произвести моделирование базы данных;
- разработать мобильное приложение;

- реализовать модули приложения;
- протестировать мобильное приложение.

В первой главе данной работы осуществляется анализ предметной области и дается ее краткая характеристика. Проводится анализ одного из бизнес-процессов с помощью выбранного на основе сравнения кейс-средства. Приводятся существующие продукты на рынке и проводится их сравнение на соответствие подобранным критериям, что позволяет отобрать основные требования и функции, которые должны быть использованы в разрабатываемом приложении.

Во второй главе выбирается основное программное обеспечение, которое будет использовано при программировании, а также основной язык. Также проводится проектирование базы данных мобильного приложения.

В третьей главе представлены основные элементы и функции, реализованные в мобильном приложении, а также проведено тестирование.

Глава 1 Анализ предметной области

1.1 Характеристика деятельности Центра цифровых компетенций

Центр цифровых компетенций(ЦЦК) является одним из отделов Тольяттинского Государственного Университета(ТГУ). В структуре Университета он подчиняется Проректору по цифровизации. Ключевой целью отдела является создание системы практикоориентированной ИТ-подготовки в Тольяттинском государственном университете. Основные функции, за которые отвечает отдел:

- проектирование образовательных программ по ИТ-направлению, формулировка требований к учебным курсам;
- организация наполнения образовательных программ информацией, посредством разработки, перевода, закупки либо же аренды;
- обеспечение образовательных программ современными образовательными технологиями постановки практических навыков (совместно с отделом технологий онлайн-образования ТГУ);
- обеспечение кадрами(сотрудниками) процессы разработки информации и обучения;
- участие в создании востребованного и продаваемого на рынке образовательного продукта в составе команды проекта Росдистант;
- анализ обратной связи от потребителей и партнеров, постоянная актуализация образовательных программ;
- работа в рамках подразделения IT Student.

Также сотрудники отдела занимаются оформлением различных документов, для осуществления всех вышеперечисленных функций. Эти документы передаются между лицами или отделами. Они могут различаться по типу: периодичные, которые оформляются в определенный период времени или же не периодичные, которые оформляются по мере надобности.

К периодическим документам можно отнести таблицы за первую половину месяца, оформляется 12-13 числа, и за вторую половину месяца, оформляется 27-29 числа, а также договора о почасовой оплате, такие как заявление на почасовую оплату труда, которое оформляется в начале семестра преподавателем, и карточка нагрузки, которая оформляется в конце семестра.

К не периодическому типу документов можно отнести служебные записки (на оформление доступа к ресурсам, на создание ЭЦП, на размер оплаты труда, и т.д.), официальные письма в органы власти и компаниям, договора на оказание услуг (на перевод с английского на русский, на разработку контентов или части контентов и т.д.), договора гражданско-правового характера (на оказание преподавательских услуг, на выполнение работ), договора о практической подготовке обучающихся, лицензионный договор о передаче неисключительного права пользования Программы для ЭВМ, лист согласования, дополнительное соглашение к договору, соглашение о расторжении договора, а также акты выполненных работ.

Таким образом мы рассмотрели характеристику деятельности центра Цифровых компетенций.

1.2 Выбор case-средств для описания бизнес-процессов

Бизнес-процессом можно называть последовательность взаимосвязанных действий и событий, направленных на достижение какой-либо цели. При проектировании их рассматривают с целью их анализирования и дальнейшего улучшения.

Есть несколько способов, с помощью которых можно описать бизнес-процесс:

- текстовый способ, который последовательно описывает словами бизнес-процесс;
- табличный способ, разбивает бизнес-процесс по ячейкам структурированной таблицы;

- графический подход, информация подается в виде графических образов.

В последнее время больше всего в разработке используется именно графический подход, так как он позволяет визуализировать информацию. У данного подхода есть большое количество методологий, например, как IDEF0, IDEF3, DFD, BPMN, UML и другие, а также большое количество CASE-средств, таких как Vpwin, ERwin и другие. [10]

CASE-средства (Computer Aided System/Software Engineering) - это набор средств и подходов, который способствует созданию программного обеспечения высокого качества, свободного от ошибок и удобного в обслуживании.

Рассмотрим некоторые из CASE-средств: Ramus 2.0, Microsoft Visio, Vpwin, Draw.io.

Ramus 2.0 – это программное обеспечение, которое предоставляет средства для моделирования бизнес-процессов с использованием методики IDEF0((Integration Definition for Function Modeling) и DFD (data flow diagrams). Оно позволяет анализировать и проектировать функциональные модели с помощью графических инструментов, обеспечивая удобный способ визуализации и документирования бизнес-процессов. Есть функция формирования отчетности по моделям и системе классификации. [6,9]

Microsoft Visio - это программа для создания диаграмм и визуализации информации. Широко используется для моделирования бизнес-процессов, создания организационных схем, схем баз данных, сетевых диаграмм, диаграмм потоков данных и многого другого. В Microsoft Visio есть набор инструментов, который позволяет удобно создавать, редактировать и представлять информацию в виде графических диаграмм различных типов, а также работать вместе с другими пользователями в режиме реального времени. Программа содержит классические функции MS Office Word и Excel, а также позволяет импорт данных прямо из MS Excel и Access. [1]

Врwin - это программное обеспечение, которое предоставляет инструменты для моделирования бизнес-процессов и проектирования баз данных. Позволяет анализировать и оптимизировать бизнес-процессы, создавать ER-диаграммы для проектирования структуры баз данных, а также выполнять другие задачи, связанные с управлением данными и бизнес-анализом. Работает с методологиями в IDEF0, IDEF3 и DFD.

Draw.io – это бесплатный Интернет-ресурс, для создания различных схем и диаграмм, например: организационные схемы, диаграммы потоков данных, сетевые диаграммы. Имеет большой каталог шаблонов и фигур. Возможность сохранения в облачных ресурсах и импортирования в форматах JPG, PNG, SVG, VDSX. Есть функция совместной работы, что удобно для работы с продуктами Confluence и Jira.

Для сравнительного анализа будем использовать следующие критерии оценки: доступность, функционал, интуитивность интерфейса, экспорт в различные форматы, количество различных методологий. В таблице 1 отображены результаты сравнительного анализа.

Таблица 1 – Сравнительный анализ CASE-средств на основе выделенных критериев

Критерий	Ramus 2.0	Microsoft Visio	Врwin	Draw.io
Доступность	+	-	-	+
Функционал	+	+	+	+
Интуитивность интерфейса	-	+	-	+
Экспорт в различные форматы	+	+	+	+
Количество различных методологий	-	-	-	+

Таким образом таблица показывает, что Ramus 2.0 не соответствует двум критериям, Microsoft Visio также не соответствует по двум критериям, Врwin не соответствует по трем критериям, а Draw.io соответствует всем критериям оценки.

1.3 Анализ модели бизнес-процесса

Рассмотрим процессы, которые связаны с документами в ЦЦК на примере процесса согласования договора ГПХ. Сотрудник ЦЦК заполняет шаблон для договора, согласовывает его с лицом, с которым он оформляется, подготавливает дополнительные документы, согласовывает договор с руководителем, передает в юридический отдел, по возвращению передает в материальный отдел, отправляет договор по почте России, отправляет трек-номер лицу, с которым договор оформляется, и отслеживает возвращение подписанного договора, сдает оригинал договора на хранение, сделав копию. Контекстная диаграмма данного процесса представлена на рисунке 1.

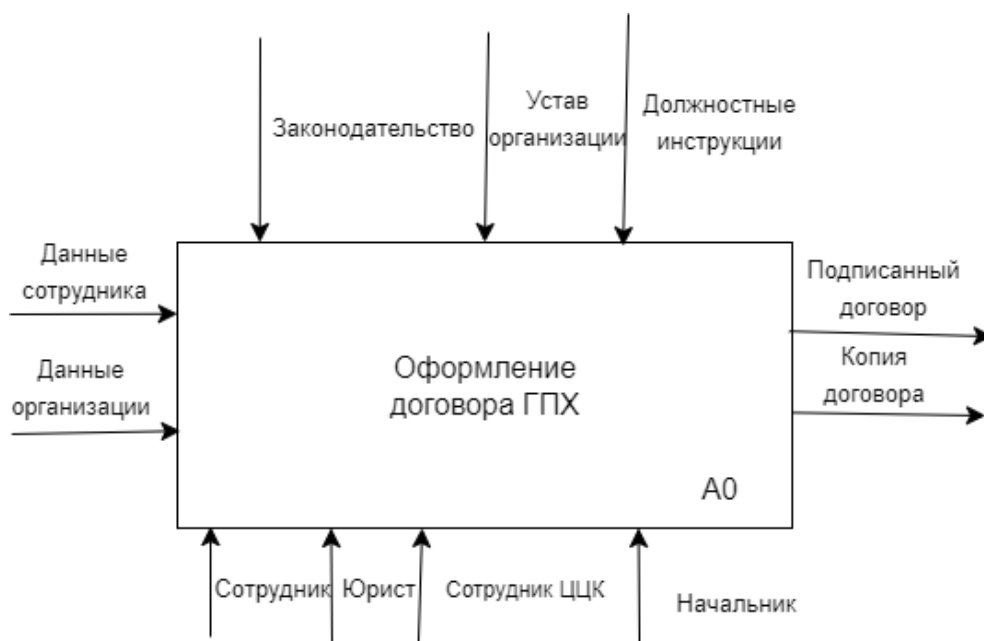


Рисунок 1 – Контекстная диаграмма процесса оформления договора ГПХ

На рисунке 2 представлена декомпозиция данного процесса.

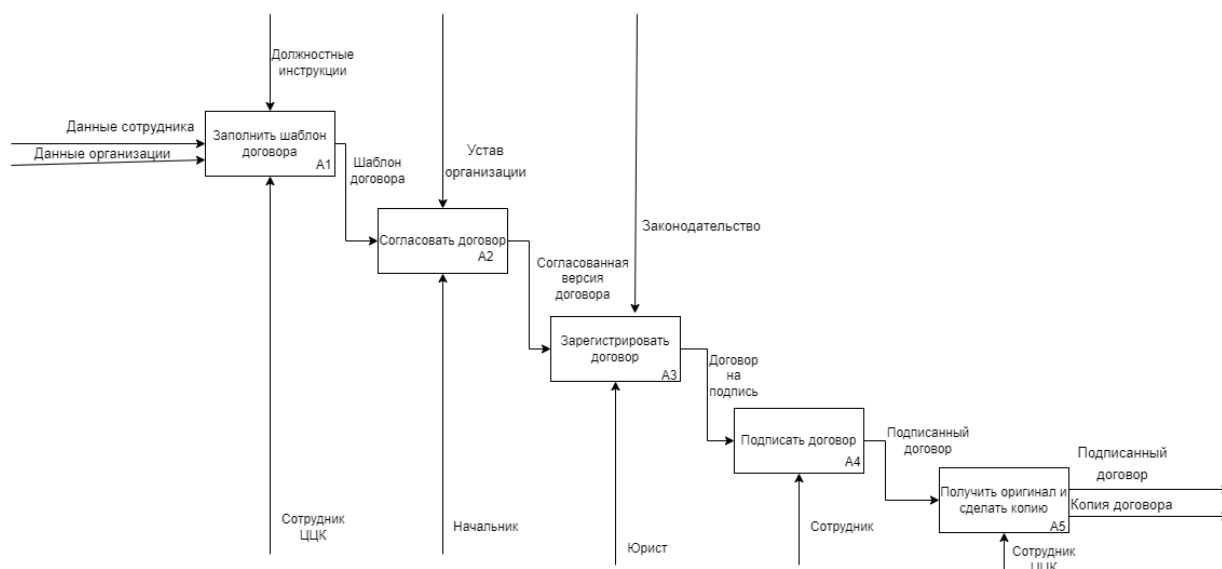


Рисунок 2 - Декомпозиция процесса оформления договора ГПХ

Рассмотрев процесс оформления договора ГПХ можно прийти к выводу что большинство действий этого процесса подразумевает что работник ЦЦК будет знать и помнить, где в какой момент находится документ, т.к. это его задача получить подписанный договор. Все действия по перемещению документа осуществляются непосредственно сотрудником ЦЦК и соответственно нельзя исключить человеческий фактор и возможность наличия ошибок при невозможности вспомнить дату, когда документ будет готов или отдел, в который был отдан документ.

Можно сделать вывод о том, что отслеживание перемещений документа можно упростить при помощи приложения, которое будет позволять отслеживать кому отдан был документ и когда его нужно получить обратно.

Сформируем функциональные требования для разрабатываемого приложения, исходя из потребностей заказчика:

- должно работать без доступа в интернет;
- должно выполнять основную функцию создания записи о движении документа;
- должно иметь перечень возможных документов;
- должно позволять редактуру и удаление существующей записи;

- должно затрачивать минимальные ресурсы для установки и поддержания работоспособности;
- должно иметь возможность дальнейшего развития с помощью добавления нового функционала, который может потребоваться в будущем.

Таким образом мы провели анализ процесса согласования договора и сформировали функциональные требования для приложения.

1.4 Обзор и анализ существующих аналогов мобильных приложений

С каждым годом мобильные телефоны и приложения для них развиваются все больше и больше, поэтому у решения любой проблемы обязательно найдется одно или несколько решений. Рассмотрим похожие приложения из Google Play, которые взаимодействуют с документами.

NoPaper – мобильное приложение для работы электронной подписью со сканированием и обменом документами. Основные возможности:

- работает со всеми видами электронной подписи;
- позволяет выпустить электронную подпись онлайн;
- пользователь может подписать документ со смартфона;
- позволяет увидеть статус всех присланных документов;
- позволяет создавать маршруты подписания и согласования;
- имеет архив документов;
- позволяет хранить документы в облаке или на сервере компании.

В основном приложение NoPaper ориентированно на работу с электронными документами и подписанием их с помощью электронной подписи, что явно видно из его основных функций. Приложение имеет хороший рейтинг в 4,5 звезд, а также пробный период, который является бесплатным, однако в основном функции скрыты за оплачиваемой подпиской.

СЭД «Практика» - мобильное приложения для документооборота.

Основные возможности приложения:

- отслеживание заданий от руководства;
- позволяет просмотр документов в приложении;
- поиск по существующим документам.

Авторизация в приложении осуществляется через основной сайт компании, разработавшей мобильное приложение. Приложение имеет рейтинг в 3,7 звезд. Нет никакой информации о существовании оплаты. Есть существенная разница между работой на сайте и в приложении, отсутствует часть функций.

1С: Документооборот МК - мобильное приложение для доступа к возможностям системы 1С: Документооборота. Основные возможности:

- отслеживание задач;
- просмотр почты;
- обсуждения на форуме, в чате, видеозвонке;
- просмотр документов;
- есть различные виджеты;
- отслеживание отсутствий;
- сканер документов;
- учет рабочего времени;
- уведомления различного характера (например, о новой задаче, сообщении).

Мобильное приложение в основном ориентированно на работу между сотрудниками компании, выдаче распоряжений и отслеживанию графика работы. Является платным решением как сайт, так и мобильное приложение. Соединение происходит через базу данных.

1С: Документооборот 2.2 - также мобильное приложение для доступа к возможностям системы 1С: Документооборота. Является другой версией приложения 1С: Документооборот МК, в основном имеет те же функции и также является платным решением.

Рассмотрев все вышеперечисленные мобильные приложения, проведем их сравнительный анализ, по следующим критериям: добавление задачи, доступность, наличие регистрации, добавление заметок к задаче, количество ресурсов, затрачиваемое на установку, работа с интернетом. Результаты представлены в таблице 2.

Таблица 2 - Анализ представленных приложений на соответствие выставленным критериям

Название приложения	Добавление задачи	Доступность	Регистрация	Заметки к задаче	Ресурсы	Интернет
NoPaper	+	+/-	+	+	+	+
СЭД «Практика»	+	+	+	-	-	+
1С:Документооборот МК	+	-	+	+	-	+
1С:Документооборот 2.2	+	-	+	+	-	+

Таким образом можно прийти к выводу, что ни одно из приложений не соответствует всем критериям, что приводит к необходимости разработки собственного приложения. Основными причинами, приводящими к данному выводу, являются: конкретные требования заказчика, удовлетворение которых приведет к повышению эффективности работы, экономии времени и большей организованности при передаче документов.

1.5 Постановка задачи

Требуется разработать мобильное приложение, которое позволит работнику Центра цифровых компетенций отслеживать куда и кому переданы документы, а также дату, когда их нужно получить обратно. Сформулируем требования, которым должно соответствовать разработанное в будущем мобильное приложение. Сформированные требования позволят нам упростить процесс разработки, давая точное представление, что требуется. Существует

стандартная классификация требований, разработанная Робертом Грэйди в 1992 году. [11] Она включает в себя:

- functionality, функциональность;
- usability, удобство использования;
- reliability, надежность;
- performance, производительность;
- supportability, поддерживаемость.

Также есть и несколько возможных ограничений, следуя этой классификации:

- design, ограничения проектирования;
- implementation, ограничения разработки;
- interface, ограничения, накладываемые на интерфейсы;
- physical, физические ограничения.

Теперь же составим требования к нашей будущей системе основываясь на данной классификации.

Требования к функциональности:

- возможность добавления новой записи;
- редактирование и удаление записи;
- поиск по записям.

Требования к удобству использования:

- интуитивно понятный интерфейс;
- эстетически привлекательный дизайн.

Требования к надежности:

- доступность в реальном времени 24 часа в сутки 7 дней в неделю;
- отказоустойчивость.

Требования к производительности системы:

- высокое время отклика системы;
- минимальное время запуска и завершения работы;
- минимальное потребление ресурсов системы мобильного телефона;
- простота установки.

Технические требования:

- совместимость с Android версии 5.0 и выше;
- возможность работы без соединения с интернетом;
- защита несанкционированного доступа.

Рассмотрим основные модули, которые необходимо реализовать;

- модуль, который позволит просматривать все существующие записи и осуществлять по ним поиск;
- модуль, который позволит пользователю создать новую запись и отредактировать ее. Должен содержать тип документа, основной документ, исполнителя, которому отдан документ, сроки исполнения и установление напоминания. Модуль должен быть организован посредством заполнения пользователем полей и функцией установить напоминание;
- модуль базы данных, который позволит сохранять записи.

Мы сформировали все требования, придерживаясь которых мы сможем получить приложение, которое будет отвечать все принятой классификации.

Выводы по первой главе

В данной главе был произведен анализ бизнес-процесса работы ЦЦК, были произведены обзор и анализ существующих решений на рынке мобильных приложений и выявлено что ни одно из них не соответствует запрашиваемым требованиям. На основе данного вывода произведена дальнейшая разработка требований к будущему приложению на основе существующих стандартов.

Глава 2 Проектирование мобильного приложения

2.1 Выбор средств разработки

За последнее десятилетие развитие мобильных устройств и мобильных приложений вышло на новый уровень благодаря прогрессу. Основной операционной системой, которой пользуются люди, является ОС Android. Процент устройств на рынке составляет около 80%, благодаря открытому коду, который позволяет установить операционную систему на любое устройство. Таким образом система используется не только непосредственно производителем, но и другими компаниями, которые используют ОС Android при выпуске своих устройств на рынок. Операционная система поддерживает обширный каталог приложений, которые может создать каждый при помощи доступного программного обеспечения. Android каждый год получает обновление версии. На рисунке 3 представлен график обновлений.

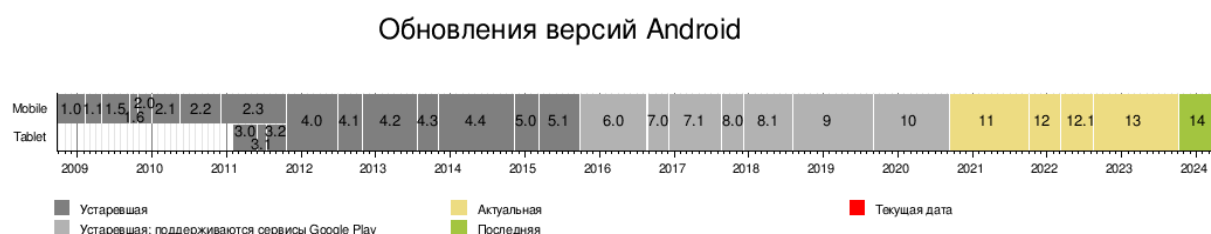


Рисунок 3 – График обновления версий Android

Для разработки под систему Android есть несколько различных сред которые мы рассмотрим, таких как: Eclipse, IntelliJ IDEA, NetBeans IDE, Android Studio.

Eclipse – это интегрированная среда разработки, которая в основном позволяет работать с языком Java, на котором она и написана. Данная среда является open-source и свободно интегрируется. Eclipse поддерживает

подключение плагинов, что дает ему возможность использоваться и с другими языками программирования. Таким же образом Eclipse имеет плагин и для разработки на Android называемый Android Development Tools (ADT). [5]

Intelij IDEA является более профессиональной средой разработки, которой часто пользуются в больших компаниях. У нее много полезных функций, которые упрощают и ускоряют написание кода, например автоматическая подстановка символов. Для сборки Intelij IDEA предлагает Gradle и Maven. Основным языком для работы также является Java, но, как и Eclipse позволяет установку плагинов, что может добавить новые функции и поддержку других языков. Для разработки на Android используется плагин.

NetBeans IDE также является интегрированной средой разработки, поддерживаемой такие языки как Java, Python, PHP, JavaScript, C, C++, Ада и ряда других. Поддерживается компанией Oracle, а изначально разработана компанией Sun Microsystems, которую позже поглатила компания Oracle. Существует плагин для упрощения разработки на ОС Android.

Android Studio отличается от остальных сред разработки тем, что была разработана непосредственно для работы с ОС Android. [3, 13] Она является бесплатным программным продуктом, который разработала компания Google. Своим дизайном и функционалом напоминает Intelij IDEA, что делает переход на данный продукт более комфортным. Android Studio доступна для всех компьютерных операционных систем. Имеет предустановленные макеты под Android приложения, поддерживает разработку под Tv, Watch и Wear. Сборка осуществляется с помощью Gradle, поддерживает Java и Kotlin. Для работы необходимо установить Android SDK. Есть возможность подключить устройство на Android для выполнения написанного кода либо же использовать встроенный эмулятор.

Проведем сравнительный анализ вышеперечисленных сред разработки по следующим признакам: доступность, является ли плагином, возможность разработки без установки плагина, удобство установки, удобство среды, шаблоны. Результаты приведены в таблице 3.

Таблица 3 – Сравнительный анализ средств разработки

Среда разработки	Доступность	Плагин	Разработка без плагина	Удобство установки	Удобство среды	Шаблоны
Eclipse	+	+	-	-	-	-
Intelij IDEA	-/+	+	-	+	+	+
NetBeans IDE	+	+	-	-	-	-
Android Studio	+	-	+	+	+	+

Исходя из результатов сравнительного анализа из всех рассмотренных средств разработки будем использовать Android Studio [13], так как это единственное программное обеспечение, которое было написано непосредственно для работы с Android, оно подходит по наибольшему количеству параметров и также оно имеет наиболее удобный функционал для разработки мобильного приложения.

2.2 Выбор языка программирования

Как было сказано ранее Android Studio [13] позволяет использовать два языка программирования: Kotlin, Java.

Kotlin это кроссплатформенный язык, статически типизированный, разработанный компанией JetBrains в 2016 году чтобы помочь разработчикам сэкономить время и уменьшить количество кластеризации и шаблонного кода. В 2018 году компания Google объявила его официальным языком для Android разработки. В основном он непосредственно используется для разработки Android приложений, когда Java используется повсеместно. Синтаксис Kotlin направлен на сокращение написания кода, что делает его лаконичным, а также он ориентирован на совместимость с Java-кодом. Благодаря статической типизации тип каждого выражения известен при компиляции. [12]

Основные особенности Kotlin:

- null safety: обеспечивает предотвращение ошибок, обрабатывая Null pointers на уровне системы типов. Он знает, когда переменная может быть Null, а когда нет;
- coroutines: асинхронная работа, подобие threads, но более легкие;
- поддержка функционального стиля программирования: позволяет использовать лямбда-выражения, функции высшего порядка и другие функциональные концепции; [20]
- краткий синтаксис: позволяет писать более чистый и понятный код;
- безопасность типов, позволяет предотвратить ошибки типизации.

Java объектно-ориентированный язык программирования выпущенный в 1995 году компанией Sun Microsystems. Он не компилируется в машинный код, а полагается на виртуальную машину, которая принимает байт-код. Данная виртуальная машина Java должна быть установлена для корректной работы. Для работы с Android виртуальная машина называется Dalvik, ее работа это интерпретация байт-кода найденного в CPU и исполнение программы на процессоре. Java используется для разработки так как большинство разработчиков владеют этим языком и он привлекает своей простотой. [16]

Из основных преимуществ Java можно выделить следующие:

- объектно-ориентированность: включает наследование, полиморфизм, абстракцию и инкапсуляцию, которые помогают структурировать блоки программы и определять взаимодействие между ними;
- независимость от платформ: байт-код языка Java не привязан ни к какой платформе и пока вы используете Java Virtual Machine (JVM) он может быть выполнен где угодно;
- защищенность: использование собственной среды выполнения, проверка типов при компиляции, проверка во время выполнения;
- автоматическое удаление элементов, которые не используются, отлов исключений;

- многопоточность: позволяет создавать отзывчивые приложения, которые работают с большим количеством задач одновременно;
- большое количество созданных библиотек, которые предоставляют различные инструменты разработки. [17]

Проведем сравнение в форме таблицы по следующим признакам: скорость работы, компактность кода, преимущественное покрытие рынка, полнота документации. Результаты сравнения представлены в таблице 4.

Таблица 4 – Сравнение языков программирования

Язык программирования	Скорость работы	Компактность кода	Покрытие рынка	Полнота документации
Java	+	-	+	+
Kotlin	+	+	-	-

Естественно, и у Java и у Kotlin есть свои сильные и слабые стороны. Для разработки приложения был выбран язык программирования Java на основе опыта работы ранее, а также то, что Kotlin по сравнению с Java более новый язык, что делает более затруднительным поиск документации.

2.3. Диаграмма вариантов использования

Унифицированный язык моделирования (The Unified Modelling Language(UML)) предлагает стандартную языковую спецификацию для поддержки объектно-ориентированного подхода к системному анализу и проектированию. Диаграммы использования имеют строго формализованный синтаксис, для документирования сценариев, описывающих поведение системы. Диаграмма вариантов использования состоит из актеров и методов использования.

Актеры — это объекты, которые взаимодействуют с системой, они окружение системы, обозначается человеком.

Методы использования - это сервисы и функциональные возможности, которые используют актеры, они должны подходить параметрам актеров, обозначается овалом.

Диаграмма вариантов использования определяет, как ожидания внешних объектов в отношении системы, так и ее специфические свойства. Взаимодействия между актерами и методами показываются с помощью соединения их линией. Основным актером является работник ЦЦК. [8,18]

Для разрабатываемого приложения определим основные варианты использования:

- создание новой задачи;
- редактирование задачи;
- удаление задачи;
- просмотр задачи;
- поиск задачи.

На рисунке 4 представлена диаграмма вариантов использования разрабатываемого мобильного приложения.

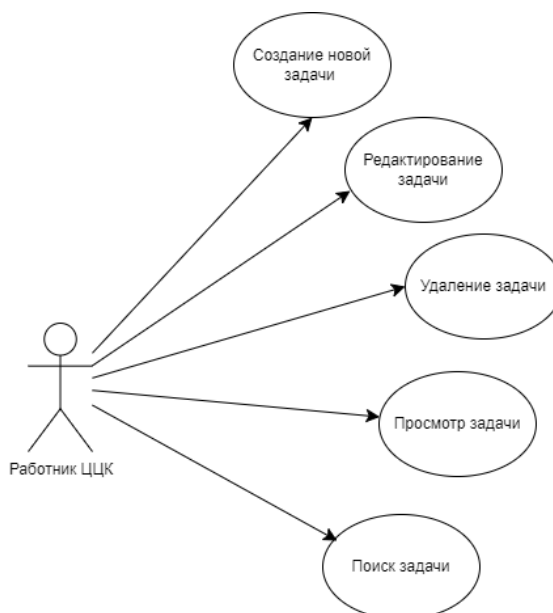


Рисунок 4 - Диаграмма вариантов использования приложения

Таким образом, мы определили основные варианты использования и спроектировали диаграмму вариантов использования приложения.

2.4 Моделирование базы данных

Моделирование данных это процесс построения диаграмм для потоков данных, при создании или редактировании баз данных. Такие диаграммы помогают определить характеристики для форматов данных, структуры и функций, которые понадобятся для работы с базами данных.

Есть несколько уровней моделей:

- концептуальная модель данных: первый уровень детализации, представляет общую структуру и содержание;
- логическая модель данных: второй уровень детализации, описывает поток данных и содержимое базы данных, добавляет детали к концептуальной модели, но не включает спецификации;
- физическая модель данных: третий уровень детализации, включает особенности реализации логической модели, детали структуры базы данных. [15]

Концептуальная модель показывает связь между сущностями и их характеристиками. [4] Выделим такие сущности как документы, задачи, рассылка напоминаний. Концептуальная модель базы данных представлена на рисунке 5.

Логическая модель раскрывает концептуальную, на основе которой и строится. У сущности документы есть атрибуты: тип документа, основной документ и сопроводительные документы. У сущности задача есть такие атрибуты как: id задачи, сведения о документе, исполнитель, дата возврата, место нахождения документа и дополнительное описание. Сущность рассылка напоминаний содержит атрибуты: время дата и id задачи. Теперь построим логическую модель данных, которая представлена на рисунке 6.

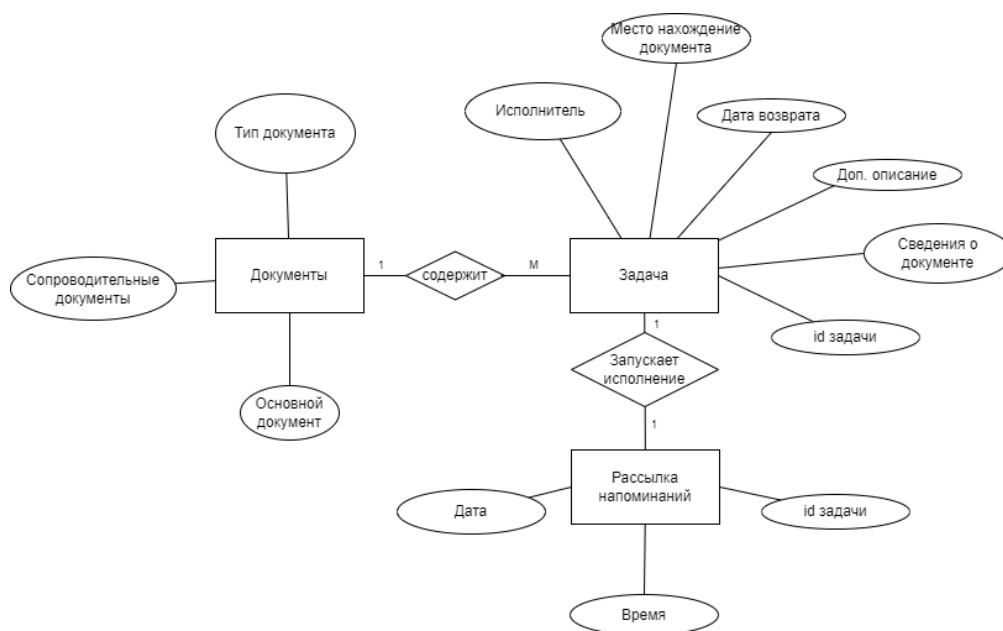


Рисунок 5 - Концептуальная модель базы данных

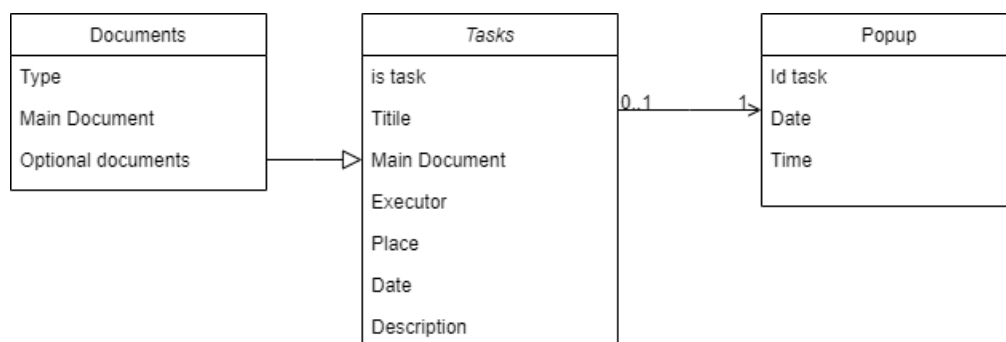


Рисунок 6 - Логическая модель базы данных

На данном этапе мы рассмотрели все возможные уровни моделей базы данных, которые помогут в реализации базы данных мобильного приложения.

Выводы по второй главе

В этой главе мы определились со средством разработки, языком программирования, спроектировали диаграмму вариантов использования и смоделировали базу данных приложения при помощи концептуальной и логической модели данных.

Глава 3 Реализация мобильного приложения

3.1 Разработка мобильного приложения

При создании проекта в программе Android Studio прилагается множество шаблонов для работы, они позволяют начать работу с уже подготовленными элементами, которые могут помочь в разработке. Есть множество шаблонов, они представлены на рисунке 7, при разработке данного приложения использовался шаблон Empty Activity, который создает основное пустое окно.

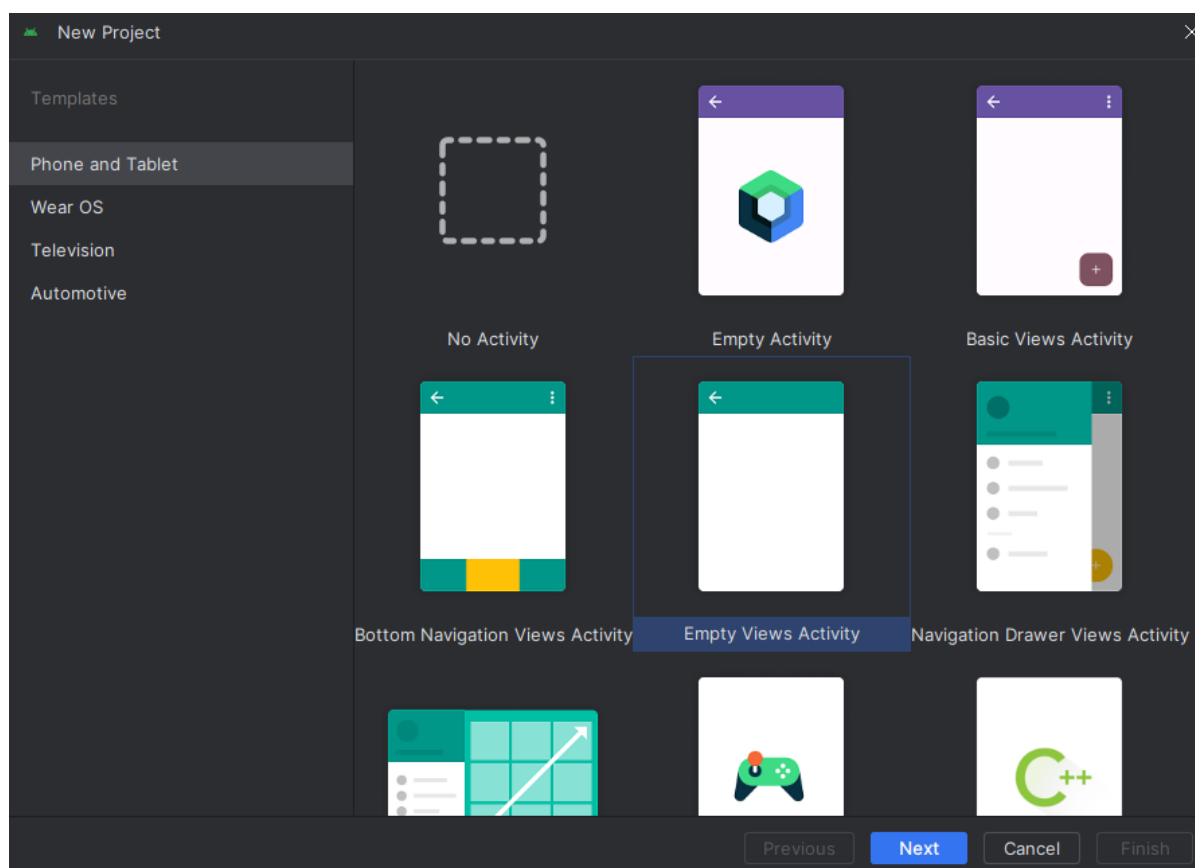


Рисунок 7 – Шаблоны программы Android Studio

Далее перед началом разработки должен быть выбран язык разработки и минимальная версия software development kit (SDK), которая определяет,

начиная с какой версии операционной системы будет возможность запустить данное приложение. Была выбрана версия SDK 23, как оптимальная, т.к. по статистике большинство устройств работает на Android 7 и выше. Статистика наиболее используемых версий Android представлена на рисунке 8.

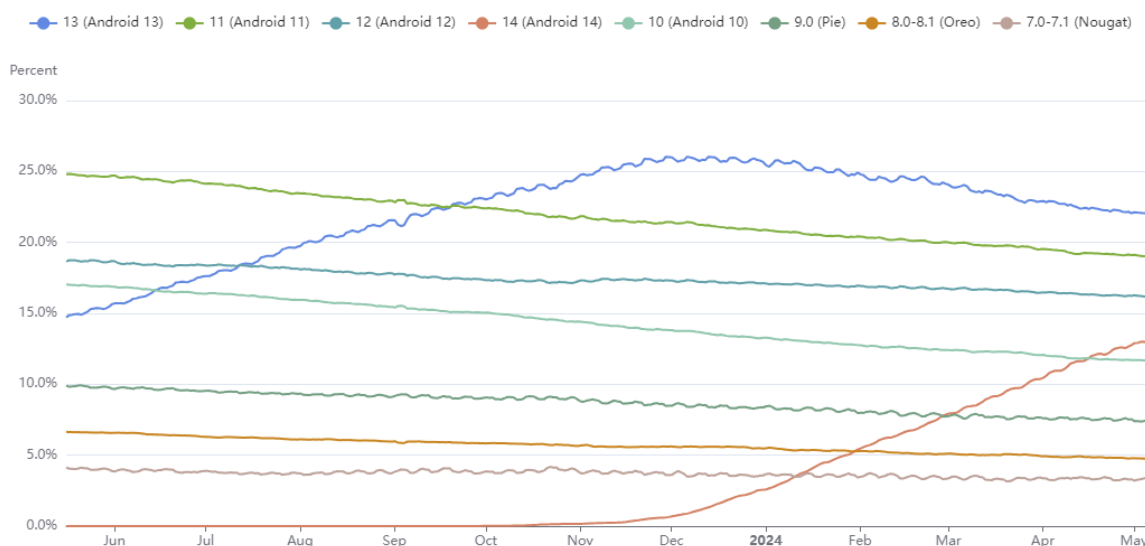


Рисунок 8 – Статистика используемых версий Android за 2023-2024 год

Сборка приложения будет происходить при помощи Gradle. В файле «gradle. build» собрана информация, которая используется при сборке. Он содержит основную информация, такую, например, как SDK, информацию о приложении, зависимости, которые необходимы для работы приложения.

При реализации мобильного приложения под названием «OverlookDoc» было создано 5 классов и 2 интерфейса. Ко всем классам, которые реализуют взаимодействие пользователя с приложением, прилагается также файл xml, который организывает внешний вид экрана для пользователя. Для удобства организации было создано три папки в дополнение к основной. Структура разработанного приложения представлена на рисунке 9.

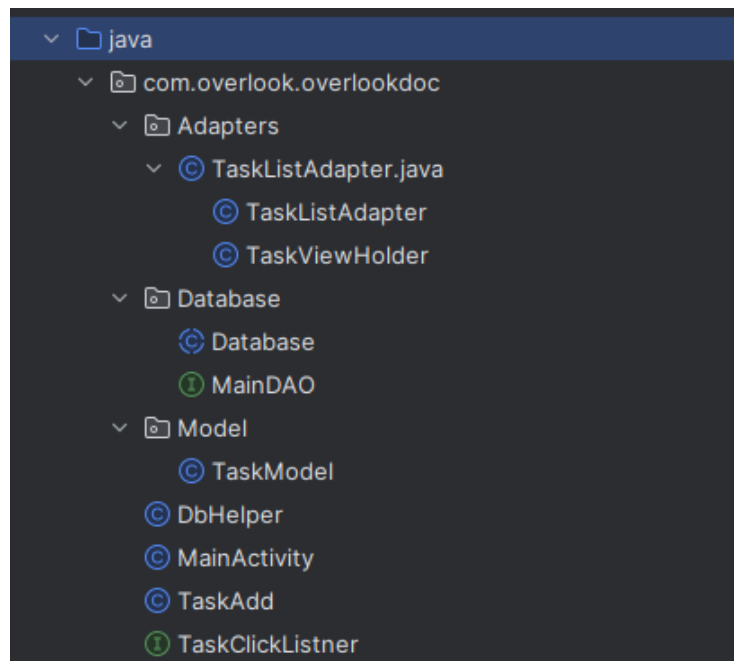


Рисунок 9 – Структура проекта

В папке Model хранится файл: TaskModel.java – реализует основную модель приложения.

В папке Database хранятся:

- Database.java – конфигурация базы данных;
- MainDao.java – интерфейс для взаимодействия приложения и базы данных.

В папке Adapters находятся файлы:

- TaskListAdapter.java – вывод списка задач;
- TaskViewHolder.java – адаптер для списка задач.

В основной папке находятся файлы:

- MainActivity.java – основной класс приложения, с которого происходит запуск;
- DbHelper.java – обновление, создание базы данных;
- TaskAdd.java – класс добавления новой задачи;
- TaskClickListener.java – интерфейс для обработки нажатий на задачу.

На рисунке 10 представлена диаграмма данных классов. На ней представлены классы с их атрибутами и операциями.

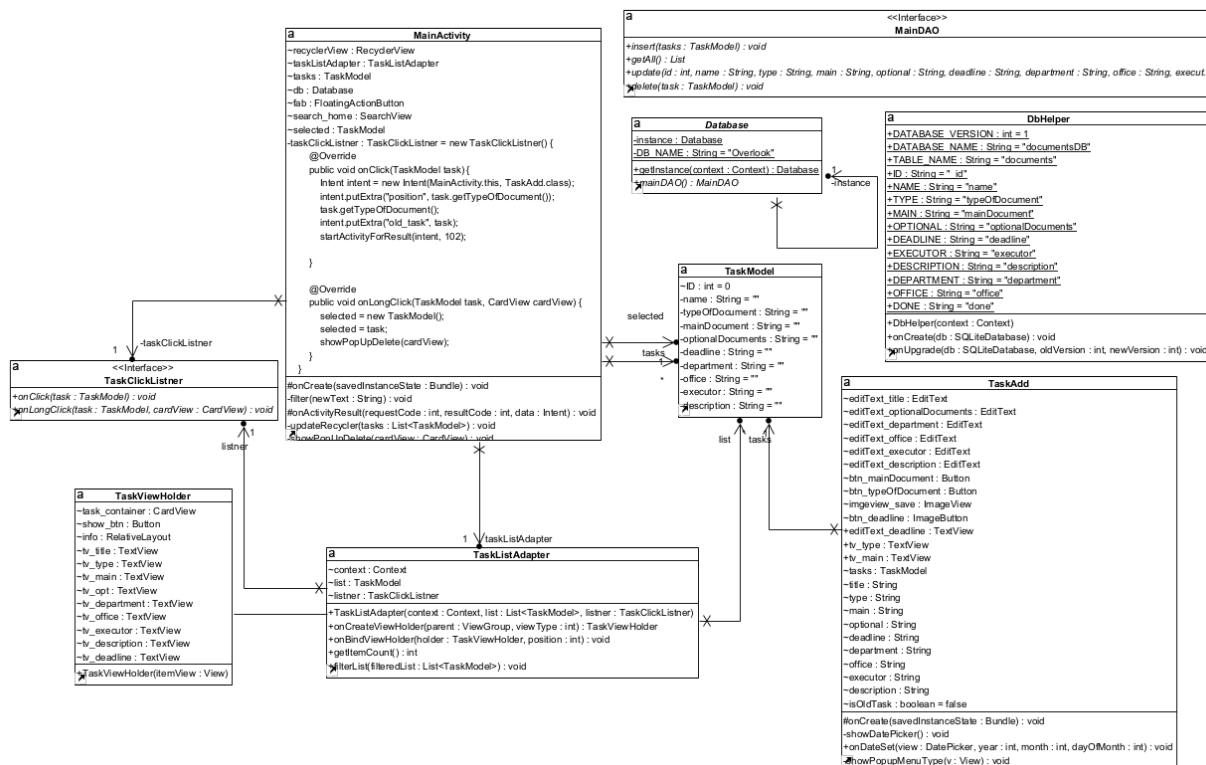


Рисунок 10 – Диаграмма классов мобильного приложения

На диаграмме представлены классы с их атрибутами и операциями. Рассмотрев структуру проекта, перейдем к реализации базы данных.

3.2 Реализация базы данных

Приложение позволяет пользователю добавлять новые записи о документах, которые в последствии появляются на основном экране приложения и сохраняются при выходе и входе обратно.

Для работы с данными используется библиотека Room. Данная библиотека использует в основе SQLite, поверх которого он создает API – интерфейс. Данная библиотека имеет три основных компонента, которые

необходимо реализовать - Entity, DAO и Database. Entity, по-другому сущность, которая реализовывает таблицу базы данных. Для использования добавляется аннотация наверху класса. Поля таблицы создаются по полям класса помеченного данной аннотацией. Ключ таблицы помечается аннотацией PrimaryKey [2]. В приложении за сущность базы данных отвечает класс TaskModel.java, часть кода которого представлена на рисунке 11. В данном классе также присутствуют методы получения и передачи данных в данные поля. Они используются т.к. поля имеют модификатор доступа private.

```
@Entity(tableName = "documents")
public class TaskModel implements Serializable {
    4 usages
    @PrimaryKey(autoGenerate = true)
    int ID = 0;
    3 usages
    @ColumnInfo(name = "name")
    private String name = "";
```

Рисунок 11 – Фрагмент класса TaskModel.java

Data Access Objects (DAO) – класс, который определяет методы для взаимодействия между приложением и базой данных. [14] Используется либо интерфейс, либо абстрактный класс, и помечается соответственно аннотацией DAO. Также уже внутри данного класса используются аннотации Insert, Update, Delete, которые представляют методы для работы с базой данных и соответственно расставляются над методами, реализующими эти действия. Есть и аннотация Query, которая позволяет прописывать запросы к базе данных на языке SQL прямо в коде приложения. DAO в данном приложении представлено интерфейсом MainDao.java фрагмент кода которого представлен на рисунке 12.

```

@Dao
public interface MainDAO {
    1 usage 1 implementation
    @Insert(onConflict = REPLACE)
    void insert(TaskModel tasks);

    3 usages 1 implementation
    @Query("SELECT * FROM documents ORDER BY id DESC")
    List<TaskModel> getAll();

```

Рисунок 12 – Фрагмент кода MainDao.java

Последним необходимым классом является Database.java, который дает возможность получить доступ для подключения к базе данных. Он помечается аннотацией Database и является абстрактным. [2] Содержит версию базы данных и сущности для работы. Фрагмент представлен на рисунке 13.

```

@androidx.room.Database(entities = {TaskModel.class}, version = 4, exportSchema = false)
public abstract class Database extends RoomDatabase {
    3 usages
    private static Database instance;
    1 usage
    private static String DB_NAME = "Overlook";

    1 usage
    public synchronized static Database getInstance(Context context) {
        if (instance == null) {

```

Рисунок 13 – Фрагмент кода Database.java

Таким образом была реализована база данных для разработанного мобильного приложения, что позволяет пользователю иметь возможность выполнять различные операции с данными непосредственно с телефона.

3.3 Реализация основного функционала

Основной функцией приложения можно назвать просмотр информации в формате списка задач. Для этого реализован сам список, добавление в него

записей, их редактирование и удаление. Для того, чтобы пользователь мог использовать приложение и взаимодействовать с ним используется визуальный интерфейс части, которого, пишутся при помощи файлов с разметкой на языке разметки XML (eXtensible Markup Language). Данные файлы написаны в стиле похожем на язык HTML, что делает работу с ними более удобной, так как их просто читать. В структуре файлов они находятся в папке res/layout, если они определяют экраны, либо же в res/menu, если относятся к меню. [19]

В файле task_list.xml представлена разметка того, как будет выглядеть единственная запись. С помощью компонента RecyclerView в файле activity_main.xml будут выводиться новые записи в виде списка. Для того чтобы записи отображались используется класс TaskListAdapter, который при помощи TaskViewHolder получает данные из вне и передает их как текст элементам. Фрагмент TaskListAdapter представлен на рисунке 14 ниже.

```
@Override
public void onBindViewHolder(@NonNull TaskViewHolder holder, int position)
    holder.tv_title.setText(list.get(position).getName());

    holder.tv_type.setText(list.get(position).getTypeOfDocument());
    holder.tv_main.setText(list.get(position).getMainDocument());

    holder.tv_opt.setText(list.get(position).getOptionalDocuments());

    holder.tv_department.setText(list.get(position).getDepartment());
```

Рисунок 14 – Фрагмент кода TaskListAdapter.java

При добавлении новой записи она может быть свернута или же развернута.

При свернутом положении можно увидеть только название, при развернутом доступен просмотр всех полей записи. Реализация дизайна представлена на рисунке 15.

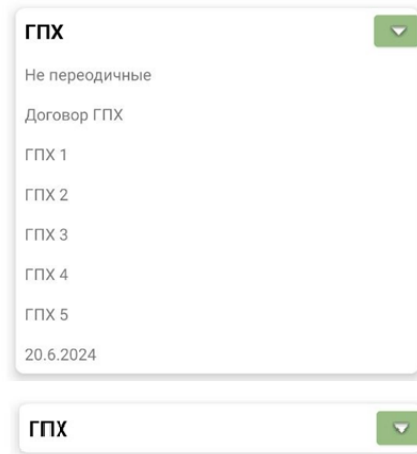


Рисунок 15 – Внешний вид добавленной записи

В activity_task_add.xml представлена разметка экрана, на котором пользователь создает новую запись. Открывается она при нажатии на кнопку с главного экрана. Логику работы экрана добавления обрабатывает класс TaskAdd.java. В данном классе происходит проверка заполнения основных полей, передача данных при помощи Intent в файл основной Активности, обработка таких кнопок, как выбор типа документа, основного документа, выбора даты, сохранения записи. В зависимости от выбора типа документа выводится разный список при выборе основного документа. Фрагмент кода с проверкой заполнения полей представлен на рисунке 16.

```
if (title.isEmpty()) {
    Toast.makeText(context: TaskAdd.this, text: "Добавьте название", Toast.LENGTH_LONG).show();
    return;
} else if (executor.isEmpty()) {
    Toast.makeText(context: TaskAdd.this, text: "Добавьте исполнителя", Toast.LENGTH_LONG).show();
    return;
} else if (type.equals("Тип документа")) {
    Toast.makeText(context: TaskAdd.this, text: "Выберите тип документа", Toast.LENGTH_LONG).show();
    return;
} else if (type.equals("Основной документ")) {
    Toast.makeText(context: TaskAdd.this, text: "Выберите основной документ", Toast.LENGTH_LONG).show();
    return;
}
```

Рисунок 16 – Проверка на заполнение полей в классе TaskAdd.java

В приложении А представлена блок-схема алгоритма создания и сохранения новой записи. Также на рисунке 17 можно увидеть дизайн страницы добавления и всплывающее сообщение, что тип документа не выбран.

Документ Гпх

Тип документа

Выбрать

Основной документ

Выбрать

Сопроводительные документы

Дата окончания

Отдел

Кабинет

Иванов

Доп.заметки

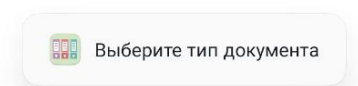


Рисунок 17 – Экран добавления новой записи

Главный класс реализует основной экран и логику приложения. Он содержит методы для обработки кнопки добавления, поиска по записям, показа списка записей, длинного и короткого нажатия на отдельную запись. При коротком нажатии происходит переход на страницу с редактированием записи, куда данные передаются при помощи Intent, а при долгом нажатии показывается меню с кнопкой удаления. Пример работы представлен на рисунке 18.

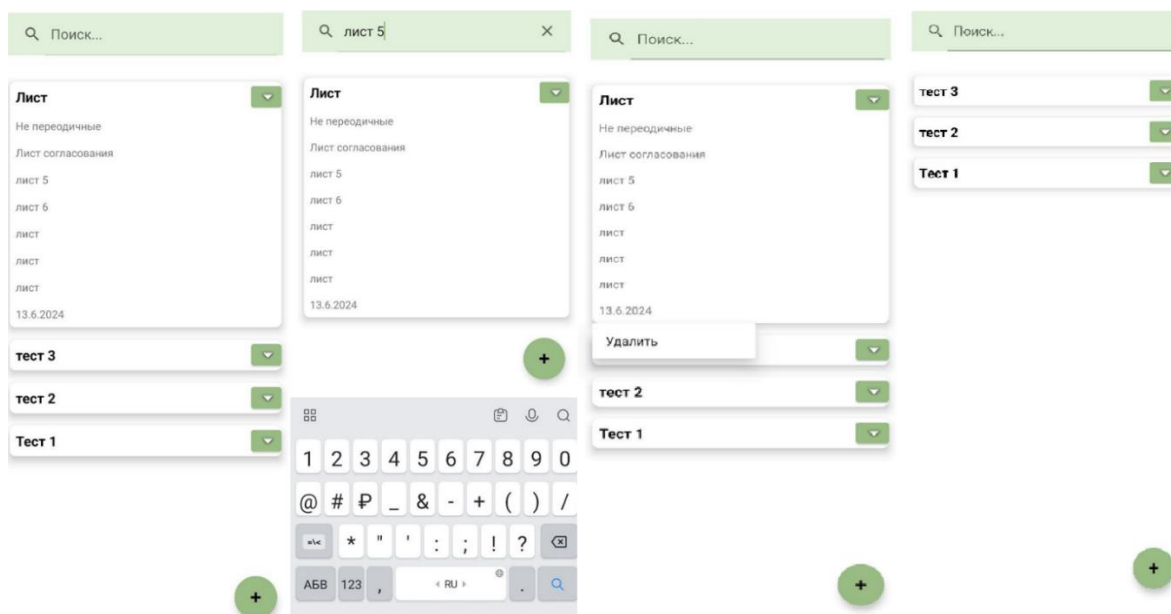


Рисунок 18 – Основной экран приложения и его функции

Таким образом мы рассмотрели все основные элементы и функции разработанного мобильного приложения.

3.4 Тестирование приложения

Одним из основных этапов разработки является тестирование, которое позволяет найти ошибки в работе программы и устранить их. Для тестирования будет использоваться функциональное тестирование.

Функциональное тестирование – тестирование, которое позволяет проверить корректность работы функционала приложения. [7] При использовании данного вида тестирования проводится проверка того, как ведет себя приложение в различных ситуациях работы пользователя.

Проведем несколько тестов, в ходе которых проверим работу функционала и сравним с ожидаемым результатом.

При запуске приложения должны появиться все основные компоненты и приложение не должно упасть с ошибкой. При добавлении новой записи должны быть доступны все необходимые поля, список выбора основного

документа должен меняться в соответствии с выбранным типом документа, обязательно должны быть заполнены поля названия, типа, основного документа и исполнителя, допускается не заполнять остальные поля, при нажатии на сохранение должен быть произведен переход на основной экран, на который должна добавиться новая запись.

Сохранение в базу данных можно считать рабочим при условии, что все данные сохраняются корректно и при выходе и новом входе в приложение все ранее сохраненные данные присутствуют.

Чтобы считать корректным удаление записи, при долгом нажатии на запись должно появляться подменю с кнопкой удалить, при нажатии на которую запись должна быть удалена из списка и базы данных. При редактировании записи по нажатию на существующую запись должен быть осуществлен переход на страницу редактирования, на которой любое изменение, при нажатии на кнопку сохранения, должно отобразиться на основной странице приложения в соответствующей записи. Поиск записи можно считать корректно работающим при условии, что запись возможно найти по любому полю существующей записи и результаты поиска при этом отображаются корректно, без не подходящих по условию записей.

Таким образом результаты функционального тестирования представлены ниже в таблице 5.

Таблица 5 – Результаты тестирования

Действие	Ожидаемый результат	Действительный результат
Добавление новой записи	<ul style="list-style-type: none"> -При не заполнении пунктов названия, типа документа, основного документа и исполнителя выводится сообщение об ошибке -При нажатии на кнопку выбрать типа документа выдается список из двух вариантов -При нажатии на кнопку выбрать основного документа варианты выбора меняются в соответствии с выбранным типом -сохранение происходит по нажатию кнопки -новая задача появляется на основном экране 	Ожидаемый результат соответствует реальному

Продолжение таблицы 5

Действие	Ожидаемый результат	Действительный результат
Открыть приложение	-Приложение открывается -Сверху расположена строка для поиска -Снизу расположена кнопка добавления -Кнопка добавления доступна	Ожидаемый результат соответствует реальному
Сохранение в базу данных	-При сохранении задача появится на основном экране -При входе и выходе из приложения останутся все сохраненные задачи	Ожидаемый результат соответствует реальному
Удаление записи	-При долгом нажатии на задачу появится допменю с функцией удаления -При нажатии «удалить» задача будет удалена -При входе и выходе из приложения задача не появится снова	Ожидаемый результат соответствует реальному
Редактирование записи	-При коротком нажатии на задачу произойдет переход на страницу редактирования -При редактировании поля и сохранении, данные на основном экране поменяются	Ожидаемый результат соответствует реальному
Поиск записи	- При вводе слова в поисковую строку найдутся все задачи, которые будут подходить по параметру поиска	Ожидаемый результат соответствует реальному

По итогам тестирования разработанного мобильного приложения можно сделать вывод, что все заявленные функции работают исправно. Это значит, что любой из возможных сценариев использования будет отработан корректно.

Выводы по третьей главе

По результатам работы проведенной в третьей главе данной работы были представлены основная структура проекта, реализация основного функционала приложения «OverlookDoc», а также проведено функциональное тестирование, по результатам которого можно убедиться в корректной работе приложения и его функций.

Заключение

В процессе выполнения данной бакалаврской работы было спроектировано и создано мобильное приложение под названием «OverlookDoc» использование которого возможно на устройствах под операционной базой Android. Целью данной работы было создание приложения, которое повысит трудоспособность и исключит фактор непредвиденных ошибок работника Центра Цифровых Компетенций Тольяттинского государственного университета.

В процессе выполнения работы были выполнены следующие задачи:

- была произведена работа по анализу Центра Цифровых Компетенций, его бизнес-процессов и исходя из данного анализа составлены функциональные требования для разработки;
- произведен сравнительный анализ аналогичного рода приложений и выявлены их сильные и слабые стороны;
- произведено проектирование мобильного приложения, в которое входит выбор средств разработки, определение вариантов использования, моделирование базы данных;
- проведена разработка мобильного приложения, удовлетворяющего ранее установленным требованиям, таким как создания записей под заказчика, редактирование, удаление и поиск по существующим записям;
- проведено функциональное тестирование по всевозможным сценариям использования, в результате которого подтвердилось корректное поведение приложения и всех его функций.

Результатом работы является мобильное приложение, которое является катонизированным средством для отслеживания перемещения документов в стенах Тольяттинского Государственного Университета и его партнеров. Данное приложение позволит исключить возможность утери информации о

документе и исключить фактор непреднамеренной ошибки, а также поможет более комфортно организовывать время.

В ходе работы все поставленные задачи были решены и достигнута изначальная цель. Разработанное приложение отвечает потребностям и запросам заказчика. Тем не менее есть ряд возможных улучшений и функций, которые могут быть добавлены позже. Примером данных улучшений являются:

- добавление бокового меню с категориями основных документов, которое позволит четко разграничить записи. Это может улучшить опыт пользователя при использовании приложения;
- разработка функционала уведомлений, которые будут оповещать о том, что какой-либо документ нужно забрать и где.

При этом нужно помнить о том, что дальнейшие улучшения должны быть добавлены с учетом мнения непосредственных пользователей и выполнять функции, которые они считают необходимыми.

Список используемой литературы

1. Андиева Е. Ю., Сидоренко В. С. Анализ Критериев Выбора Case–Средств //Редакционная коллегия. – 2015. – 164 с.
2. Аннин П. А. Краткий Обзор Room Persistence Library, 2018 [Электронный ресурс]. URL.: <https://interactive-plus.ru/e-articles/545/Action545-470878.pdf>
3. Ахметов А. К. Операционная система Android: история создания и развития. Разработка приложений для платформы Android [Электронный ресурс]. URL: <https://cyberleninka.ru/article/n/operatsionnaya-sistema-android-istoriya-sozdaniya-i-razvitiya-razrabotka-prilozheniy-dlya-platformy-android>
4. 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 научнопрактической конференции. – 2017. – 298 с.
5. Бурнет Э. Привет, Android! Разработка мобильных приложений. 2-е издание. – СПб.: Издательство «Питер», 2016 – 256 с.
6. Гузилов А. В., Мышенков К. С., Симонов М. Ф. Анализ качества case-средств для функционального моделирования систем //Информационно-аналитические и интеллектуальные системы для производства и социальной сферы. – 2021. – 29 с.
7. Куликов С. С., Данилова Г. В., Смолякова О. Г., Меженная М. М. Тестирование программного обеспечения - 2019 – 277с.
8. Лешек А. Мацяшек. Анализ и проектирование информационных систем с помощью UML 2.0 / Лешек А. Мацяшек. – М.: Вильямс, 2016. - 816 с.
9. Федоров Н.В. Проектирование информационных систем на основе современных CASE-технологий. – М.: МГИУ, 2008 – 287 с.
10. Цуканова О.А. Методология И Инструментарий Моделирования Бизнес-Процессов Учебное пособие Университет ИТМО – 2015 – 101 с.

11. Требования к системе: классификации FURPS+ / [Электронный ресурс] – URL: <https://sysana.wordpress.com/2010/09/16/furps/>
12. Alireza Sahami Shirazi, Niels Henze, Albrecht Schmidt, Robin Goldberg, Benjamin Schmidt, Hansjorg Schmauder Insights into Layout Patterns of Mobile User Interfaces by an Automatic Analysis of Android Apps – 2013 - 10с.
13. Android Studio [Электронный ресурс]. URL: <https://developer.android.com/studio>
14. Dao [Электронный ресурс] URL: <https://developer.android.com/reference/androidx/room/Dao>
15. Data modeling overview [Электронный ресурс] URL: <https://www.sap.com/central-asia-caucasus/products/technology-platform/datasphere/what-is-data-modeling.html>
16. Matheus Flauzino, Júlio Veríssimo, Ricardo Terra, Elder Cirilo, Vinicius H. S. Durelli, Rafael S. Durelli Are you still smelling it? A comparative study between Java and Kotlin language. - 2018 - 10 с.
17. Mrs. Selina Khoirom, Moirangthem Sonia, Borishphia Laikhuram, Jaeson Laishram, Tekcham Davidson Singh Comparative Analysis of Python and Java for Beginners – 2020 - 24 с.
18. Niclas Everlönn and Stylianos Gakis Java and Kotlin, a performance comparison – 2020 – 70 с.
19. Radosław Klimek, Piotr Szwed Formal Analysis of Use Case Diagrams-2010-17с.
20. Subham Bose, Aditi, Madhuleena Mukherjee, Madhurima Banerjee A Comparative Study: Java Vs Kotlin Programming In Android Application Development – 2018 – 5 с.

Приложение А

Блок-схема создания новой записи

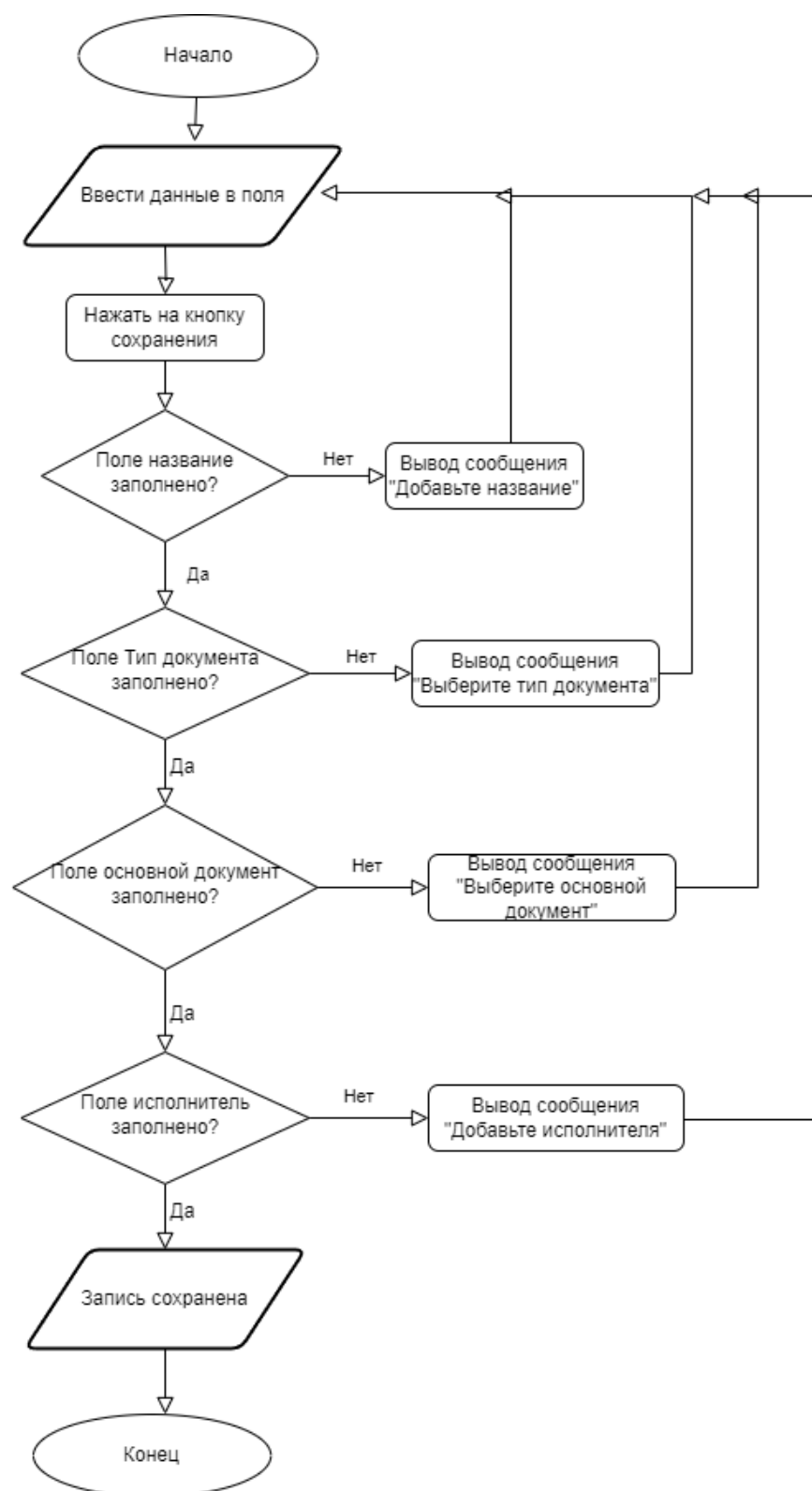


Рисунок А.1 - Блок-схема создания новой записи