

НЕГОСУДАРСТВЕННОЕ ОБРАЗОВАТЕЛЬНОЕ ЧАСТНОЕ
УЧЕРЕЖДЕНИЕ
ВЫСШЕГО ОБРАЗОВАНИЯ
«МОСКОВСКИЙ УНИВЕРСИТЕТ «СИНЕРГИЯ»

Факультет Информационных технологий

Специальность

09.02.07

(код)

Кафедра

ЦЭ

(аббревиатура)

ДИПЛОМНЫЙ ПРОЕКТ

На тему Автоматизация управления проектами "Почта России"

Обучающийся

Заболотный Давид Александрович

(Фамилия, Имя, Отчество)



(Подпись)

Руководитель

Сибирев Иван Валерьевич

(Фамилия, Имя, Отчество)



(Подпись)

Москва 2025г.

Негосударственное образовательное частное учреждение
высшего образования
«Московский университет «Синергия»



УТВЕРЖДАЮ
Декан факультета
Информационных технологий
Университета «Синергия»

Захаров
А.В. Захаров

Факультет _____ Информационных технологий _____
Кафедра _____ ЦЭ _____

ЗАДАНИЕ
на дипломный проект
обучающемуся
Заболотный Давид Александрович

1. Тема дипломного проекта: Автоматизация управления проектами "Почта России"

2. Структура дипломного проекта:

Введение

Глава 1. Аналитическая часть

1.1. Техничко-экономическая характеристика предметной области и предприятия

1.1.1. Характеристика предприятия и его деятельности

1.1.2. Организационная структура управления предприятием

1.1.3. Программная и техническая архитектура ИС предприятия

1.2. Характеристика комплекса задач, задачи и обоснование необходимости автоматизации

1.2.1 Выбор комплекса задач автоматизации и характеристика существующих бизнес-процессов

1.2.2 Определение места проектируемой задачи в комплексе задач и ее описание

1.2.3 Анализ информационных потоков проектируемой задачи

1.2.4 Анализ системы обеспечения информационной безопасности и защиты информации Анализ существующих разработок и выбор стратегии автоматизации

1.3 Анализ существующих разработок для автоматизации задачи

1.3.1 Выбор и обоснование стратегии автоматизации задачи

1.3.2 Выбор и обоснование способа
приобретения ИС для автоматизации задачи

Глава 2. Проектная часть

2.1 Разработка проекта автоматизации

2.1.1 Этапы жизненного цикла проекта автоматизации

2.1.2 Ожидаемые риски на этапах жизненного цикла и их
описание

2.2 Информационное обеспечение задачи

2.2.1 Характеристика нормативно-справочной, входной и
оперативной информации

2.2.2 Характеристика результатной информации

2.3 Программное обеспечение задачи

2.3.1 Сценарий диалога

2.3.2 Характеристика базы данных

2.3.3 Описание программных модулей

2.4 Испытания разработанного решения

2.4.1 Перечень объектов и функций, подлежащих испытаниям

2.4.2 Методы проведения испытаний

2.4.3 Проведение проверочных испытаний и их результаты

Глава 3. Обоснование экономической эффективности проекта

3.1. Выбор и обоснование методики расчёта экономической
эффективности

3.2. Расчёт показателей экономической эффективности проекта


Заключение

Список использованной литературы

Приложения

Руководитель: _____

Обучающийся задание получил «09» февраля 2025г.

Обучающийся:  _____
(Подпись)

Заболотний Давид Александрович

(Расшифровка)

Содержание

ВВЕДЕНИЕ.....	4
Глава 1. Аналитическая часть.....	8
1.1 Техничко-экономическая характеристика предметной области и предприятия (Анализ деятельности «КАК ЕСТЬ»).....	8
1.1.1. Характеристика предприятия и его деятельности	8
1.1.2. Организационная структура управления предприятием.....	10
1.1.3. Программная и техническая архитектура ИС предприятия ..	12
1.2 Характеристика комплекса задач, задачи и обоснование необходимости автоматизации.....	17
1.2.1. Выбор комплекса задач автоматизации и характеристика существующих бизнес-процессов.....	17
1.2.2. Определение места проектируемой задачи в комплексе задач и ее описание	18
1.2.3. Анализ информационных потоков проектируемой задачи....	19
1.2.4. Анализ системы обеспечения информационной безопасности и защиты информации	21
1.3 Анализ существующих разработок и выбор стратегии автоматизации («КАК ДОЛЖНО БЫТЬ»).....	22
1.3.1. Анализ существующих разработок для автоматизации задачи	23
1.3.2. Выбор и обоснование стратегии автоматизации задачи	25
1.3.3. Выбор и обоснование способа приобретения ИС для автоматизации комплекса задач	25
Глава 2. Проектная часть.....	30
2.1. Разработка проекта автоматизации	30
2.1.1 Этапы жизненного цикла проекта автоматизации	30
2.1.2. Ожидаемые риски на этапах жизненного цикла и их описание	31
2.2. Информационное обеспечение задачи.....	33
2.2.1. Характеристика нормативно-справочной, входной и оперативной информации	33
2.2.2. Характеристика результатной информации	35
2.3. Программное обеспечение задачи.....	37
2.3.1. Сценарий диалога	37

2.3.2. Характеристика базы данных	39
2.3.4. Описание программных модулей.....	43
2.4. Испытания разработанного решения	48
2.4.1. Перечень объектов и функций, подлежащих испытаниям.....	48
2.4.2. Методы проведения испытаний	49
2.4.3. Проведение проверочных испытаний и их результаты	49
Глава 3. Обоснование экономической эффективности проекта	55
3.1. Выбор и обоснование методики расчета экономической эффективности.....	55
3.2. Расчет показателей экономической эффективности проекта.....	56
Заключение	59
Список использованной литературы	61
Приложение	65

ВВЕДЕНИЕ

В современном мире эффективное управление проектами является важным фактором успешной реализации бизнес-процессов, особенно в крупных организациях с высокой нагрузкой и распределенной структурой. Автоматизация этого процесса помогает сократить рутинные операции, уменьшить количество ошибок и повысить прозрачность на всех этапах работы. Особенно это актуально для государственных компаний, таких как «Почта России», где масштабы деятельности требуют четкой координации и контроля.

Актуальность темы связана с необходимостью оптимизации управления проектами в «Почте России». В настоящее время компания сталкивается с рядом трудностей, включая недостаточную автоматизацию планирования, сложности в распределении ресурсов, задержки в отчетности и дублирование данных. Эти проблемы приводят к снижению эффективности управления проектами, увеличению сроков их реализации и росту затрат. Решением может стать внедрение специализированной информационной системы, которая обеспечит централизованное управление, автоматизацию контроля сроков и ресурсов, а также улучшит взаимодействие между подразделениями.

Целью дипломного проекта является разработка системы автоматизации управления проектами для «Почты России», направленной на повышение эффективности планирования, контроля и отчетности, сокращение сроков выполнения проектов и снижение операционных издержек.

Для достижения поставленной цели необходимо выполнить ряд задач. Во-первых, требуется провести анализ существующих систем управления проектами и оценить их применимость для «Почты России». Во-вторых, важно исследовать бизнес-процессы компании и выявить ключевые проблемы. Далее необходимо определить функциональные и нефункциональные требования к разрабатываемой системе.

Следующим этапом является разработка архитектуры и структуры системы автоматизации управления проектами, а также проектирование базы

данных для хранения информации о проектах, задачах, ресурсах и исполнителях. После этого предстоит реализовать пользовательский интерфейс, обеспечивающий удобное взаимодействие с системой.

Завершающими шагами станут тестирование системы на соответствие требованиям и расчет экономической эффективности внедрения предложенного решения.

Практическая значимость работы заключается в создании системы, которая позволит «Почте России» автоматизировать ключевые процессы управления проектами, сократить временные и финансовые затраты, повысить прозрачность выполнения задач и улучшить контроль за реализацией проектов. Внедрение такой системы способствует росту производительности, снижению рисков и укреплению конкурентоспособности компании на рынке логистических и почтовых услуг.

Глава 1. Аналитическая часть

1.1 Техничко-экономическая характеристика предметной области и предприятия (Анализ деятельности «КАК ЕСТЬ»)

1.1.1 Характеристика предприятия и его деятельности

ФГУП "Почта России" представляет собой крупнейшую национальную почтовую компанию, которая оказывает универсальные почтовые, логистические и финансовые услуги на всей территории Российской Федерации. Основанная в 2002 году в результате реорганизации федеральной почтовой службы, компания занимает важное место среди стратегических предприятий страны. В сферу ее деятельности входят почтовая связь, включая обработку и доставку корреспонденции, посылок и экспресс-отправлений, а также финансовые услуги, такие как пенсионные выплаты, денежные переводы и банковские операции через "Почта Банк". Кроме того, компания активно развивает логистические услуги, доставку интернет-заказов и предоставление государственных услуг, включая выдачу документов и распространение печатных изданий.

Компания располагает обширной инфраструктурой, насчитывающей свыше 40 000 отделений по всей России, включая труднодоступные регионы. Численность сотрудников превышает 350 000 человек, а ежедневный поток клиентов достигает 1,5 млн человек. Годовой оборот компании исчисляется сотнями миллиардов рублей. Организационная структура предприятия построена по иерархическому принципу и включает центральный аппарат в Москве, отвечающий за стратегическое управление, управления макрорегионов, координирующие работу в федеральных округах, а также региональные подразделения, обеспечивающие операционную деятельность.

История компании началась в 2002 году, когда на базе Федеральной почтовой службы было создано единое государственное предприятие. Эта трансформация стала ответом на необходимость модернизации устаревшей почтовой системы, унаследованной с советских времен. В 2003 году "Почта России" присоединилась к Всемирному почтовому союзу, что способствовало развитию международного сотрудничества. Знаковым событием стало

принятие в 2008 году Федерального закона "О почтовой связи", закрепившего за компанией статус национального оператора.

Масштабная реформа почтовой отрасли стартовала в 2013 году и включала обновление инфраструктуры, внедрение автоматизированных систем и расширение перечня услуг. В 2016 году компания была включена в список системообразующих организаций страны, а в 2019 году завершилась ее интеграция с "Почта Банком", что значительно усилило финансовый сектор деятельности. В последние годы предприятие активно развивает цифровые технологии, логистические направления и услуги электронной коммерции, сохраняя свою роль ключевого элемента инфраструктуры связи и социального обеспечения, особенно в удаленных регионах России.

Таблица 1

Основные технико-экономические показатели

№ п/п	Наименование характеристики (показателя)	2021 год	2022 год	2023 год	2024 год
1	Количество почтовых отделений, ед.	42 000	41 800	41 500	41 200
2	Численность сотрудников, чел.	335 000	340 000	345 000	350 000
3	Объем отправок (почта и посылки), млн ед.	2 850	3 100	3 400	3 650
4	Выручка, млрд руб.	214	245	278	305 (план)
5	Чистая прибыль/убыток, млрд руб.	-12	-8	-5	-2 (план)
6	Количество денежных переводов, млн операций	180	195	210	225
7	Доля интернет-заказов в посылочных отправлениях, %	65	72	78	83
8	Среднее время доставки по России, дней	6.5	6.0	5.5	5.0
9	Инвестиции в модернизацию, млрд руб.	15	18	22	25
10	Количество клиентов в месяц, млн чел.	45	48	52	55

1.1.2 Организационная структура управления предприятием

ФГУП "Почта России" обладает уникальной системой управления, которая сочетает строгую централизацию с гибкой территориальной адаптацией, что позволяет эффективно координировать работу крупнейшей почтовой сети страны. В основе организационной модели лежит четкое функционально-территориальное разделение, обеспечивающее слаженную работу всех звеньев от центрального аппарата до самых отдаленных почтовых отделений.

Сердцем этой масштабной системы выступает центральный офис в Москве, где сосредоточены ключевые управленческие функции. Генеральный директор задает стратегический курс развития, который детализируется и реализуется через коллегиальные решения правления. Специализированные департаменты - от финансового до HR - образуют мощный управленческий хаб, разрабатывающий стандарты и регламенты для всей сети.

Особенностью структуры являются восемь макрорегиональных управлений, которые выступают важнейшим связующим звеном между центром и регионами. Эти подразделения адаптируют общекорпоративные решения с учетом специфики федеральных округов, обеспечивая баланс между едиными стандартами и региональными особенностями. На уровне 85 субъектов Федерации работают региональные управления, где директора филиалов координируют операционную деятельность через специализированные отделы.

Завершают эту четко выстроенную систему районные узлы связи и более 40 000 почтовых отделений - те самые "точки соприкосновения" с клиентами. Здесь начальники отделений, операторы и почтальоны превращают корпоративные стратегии в конкретные услуги для миллионов россиян. Такая многоуровневая, но логически выверенная структура позволяет "Почте России" оставаться надежным связующим звеном между государством, бизнесом и гражданами по всей территории страны.

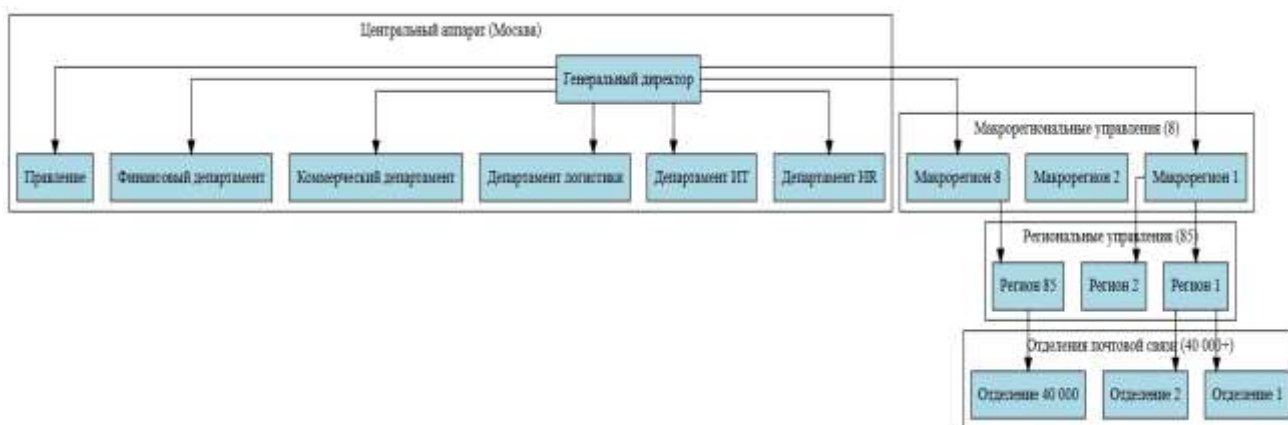


Рисунок 1. Организационная структура

ФГУП "Почта России" выстроила уникальную систему функциональных подразделений, где каждый блок играет свою особую роль в сложном механизме почтового гиганта. В операционном сердце компании бьются три ключевых департамента: почтовых услуг, отвечающий за миллионы отправок, логистики, создающий кровеносную систему доставки, и финансовых услуг, превращающий отделения в многофункциональные центры. Коммерческое крыло компании представлено департаментами продаж и маркетинга, задающими темп развития, и клиентской службой, создающей лицо компании. А в основе всего - мощные поддерживающие структуры: цифровые архитекторы из ИТ-департамента, финансовые стратеги, HR-специалисты, формирующие команду, и юридические эксперты, охраняющие интересы компании.

Управленческая структура "Почты России" - это искусный симбиоз дисциплины и гибкости. Здесь строгая вертикаль власти гармонично сочетается с территориальной специализацией, создавая идеальный баланс между контролем и автономией. Каждый уровень управления имеет четко очерченную зону ответственности: пока центр сосредоточен на глобальных стратегиях, регионы получают свободу в оперативном управлении. Особую динамику системе придают кросс-функциональные проектные группы, которые, словно спецназ, решают самые сложные задачи, преодолевая бюрократические границы.

Механизм взаимодействия в этой сложной системе отработан до совершенства. Центральный офис выступает в роли архитектора, создающего концепции и стандарты. Макрорегиональные управления - это талантливые дирижеры, адаптирующие партитуру для своего оркестра. Региональные филиалы - виртуозные исполнители, превращающие ноты в музыку повседневной работы. А функциональные департаменты - это мудрые наставники, всегда готовые подставить плечо. Такой продуманный симбиоз позволяет гигантскому организму с 350-тысячным коллективом двигаться как единое целое, сохраняя фирменный стиль обслуживания по всей стране, но при этом чутко реагируя на особенности каждого региона. Это не просто структура - это живой, дышащий механизм национального масштаба.

1.1.3 Программная и техническая архитектура ИС предприятия

Аппаратная архитектура компании «Почта России» спроектирована с учетом масштабируемости, надежности и централизованного управления, обеспечивая бесперебойную работу всех уровней организационной структуры и поддержку эффективного управления проектами. Для проектирования схем в дипломной работе был выбран ЯП Python с библиотекой graphviz

Головной офис Почты России в столице представляет собой технологически продвинутый командный центр, где сосредоточены все стратегические подразделения - от Правления до специализированных департаментов. Рабочие места сотрудников оснащены мощными компьютерами с расширенной конфигурацией, особенно в IT-отделе и проектных командах, где установлены системы с двумя дисплеями для повышения производительности. Техническая база включает в себя современный дата-центр с комплексом серверного оборудования: вычислительные мощности для бизнес-приложений, хранилища корпоративных данных, системы управления пользователями, почтовые серверы и защищенные шлюзы для дистанционной работы. Сетевая инфраструктура построена на профессиональном оборудовании, включающем многоуровневые коммутаторы, маршрутизаторы корпоративного класса, современные системы защиты периметра, решения для предотвращения утечек данных и комплексную антивирусную защиту. Интернет-подключение организовано по выделенным дублированным линиям, а связь с филиалами осуществляется через зашифрованные VPN-туннели.

Восемь межрегиональных центров управления оснащены компактными серверными помещениями с локальными вычислительными мощностями, включая кэш-серверы и системы хранения файлов. Эти подразделения поддерживают собственные доменные службы и синхронизированные копии основных баз данных. Персонал использует стандартные компьютерные рабочие места или терминальные решения с централизованным доступом к корпоративным ресурсам. Сетевое взаимодействие обеспечивается через защищенные каналы связи с использованием специализированного VPN-оборудования, объединяющего региональные офисы с центральным узлом.

На уровне 85 региональных представительств применяются облегченные серверные решения или облачные технологии с возможностью подключения к центральному дата-центру. Рабочие места укомплектованы стандартными компьютерами, многофункциональными печатными

устройствами и клиентскими терминалами для самостоятельного обслуживания.

Разветвленная сеть почтовых отделений по всей стране оборудована специализированной техникой: платежными терминалами, кассовыми аппаратами, компактными компьютерами с базовым программным обеспечением, принтерами для печати ярлыков и сканерами штрих-кодов. Крупные отделения дополнительно оснащены банкоматами и информационными киосками. В отдаленных точках, где отсутствует проводная связь, используются мобильные решения на базе LTE-технологий. Корпоративная сеть организации построена по принципу распределенной архитектуры с единым центром управления, применяя современные технологии маршрутизации, виртуальных частных сетей и сегментации трафика. Мониторинг инфраструктуры осуществляется через специализированные платформы, а вопросы информационной безопасности решаются с помощью комплексных систем анализа событий и управления угрозами.

Программная платформа предприятия разрабатывалась с применением современных технологий программирования, включая язык Python и специализированные библиотеки для визуализации структур данных, что позволило создать четкую и эффективную архитектуру информационной системы.

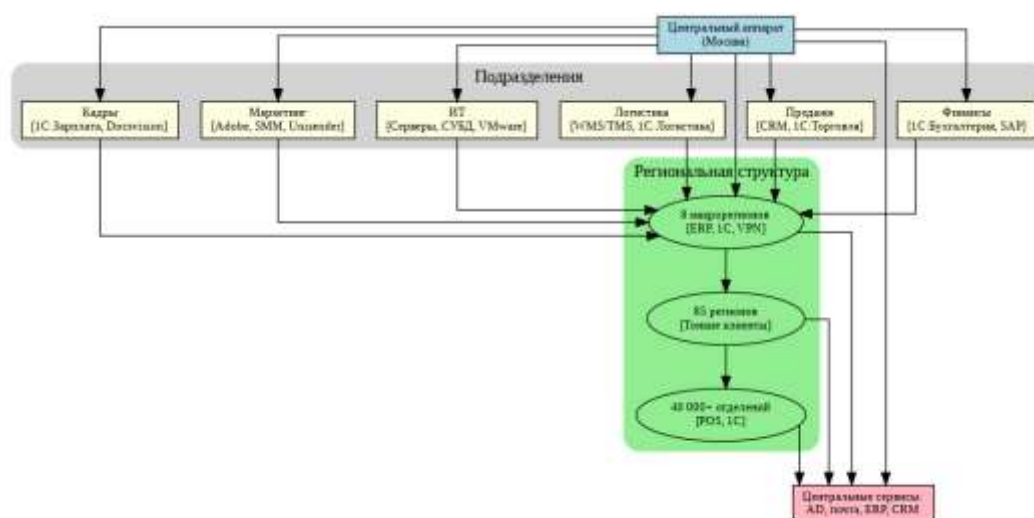


Рисунок 3. Программная архитектура

Программная экосистема национального почтового оператора представляет собой многоуровневую структуру с единым центром управления, охватывающую все географически распределенные подразделения компании. Основу этой системы составляют принципы безопасного взаимодействия, бесперебойной работы и полной совместимости между различными бизнес-процессами организации.

В столичном головном офисе развернуты специализированные программные комплексы для каждого направления деятельности. Финансовый блок использует комплекс решений для автоматизации бухгалтерского учета, включая популярные платформы для корпоративного планирования и анализа финансовых показателей. Коммерческие службы применяют современные инструменты управления клиентскими отношениями, дополненные специализированными торговыми модулями и средствами электронной коммуникации.

Оптимизацией транспортных потоков и управлением складскими запасами занимаются профессиональные логистические системы, интегрированные с картографическими сервисами и специализированными платформами учета. Техническую основу всей инфраструктуры обеспечивает комплекс серверных решений, включая системы хранения и обработки данных, платформы виртуализации и инструменты контроля ИТ-процессов.

Маркетинговый блок оснащен профессиональными графическими пакетами и комплексными решениями для digital-продвижения и управления контентом. Кадровая служба использует автоматизированные системы расчета заработной платы и электронные платформы документооборота, дополненные стандартными офисными приложениями.

Межрегиональные центры работают с облегченными версиями корпоративных систем, обеспечивающими безопасный доступ к централизованным ресурсам через защищенные соединения. Региональные представительства используют веб-интерфейсы ключевых бизнес-приложений и специализированные клиентские модули для взаимодействия с

вышестоящими структурами.

Сетевая инфраструктура почтовых отделений включает специализированные программы для торговых операций, веб-версии учетных систем и прикладное программное обеспечение для работы с клиентами. Все компоненты системы объединены в единое информационное пространство через централизованные сервисы, размещенные в защищенных дата-центрах, где обеспечивается управление пользователями, хранение данных, корпоративная коммуникация и комплексная защита информации.

1.2 Характеристика комплекса задач, задачи и обоснование необходимости автоматизации

1.2.1. Выбор комплекса задач автоматизации и характеристика существующих бизнес-процессов

В процессе изучения и ознакомления с предметной областью была спроектирована диаграмма бизнес-процессов предприятия, которая описывает все основные бизнес-процессы организации, необходимые для ее функционирования. Также во время анализа была определена область для будущей разработки, а именно информационной системы, которая будет автоматизировать основной процесс «Автоматизация управления проектами «Почта России»».



Рисунок 4. IDEF0

Данная Idef0 модель представляет собой структурированное описание

системы автоматизации управления проектами в "Почте России". Модель начинается с корпоративных политик и внутренних регламентов, которые задают общие рамки деятельности, а также включает законодательство РФ и отраслевые стандарты, обеспечивающие соответствие внешним требованиям. Далее учитываются требования бизнеса, стратегические цели компании и данные о текущих проектах, что подчеркивает ориентацию на практические нужды организации. Центральным элементом модели является блок автоматизации, включающий оптимизированные бизнес-процессы, автоматизированные рабочие процессы, систему отчетности и аналитики, а также интегрированную ИТ-систему, что обеспечивает комплексный подход к управлению проектами. Реализация этих задач поддерживается проектной командой и инструментами разработки, а также учитывает бюджет автоматизации и существующую инфраструктуру "Почты России".

1.2.2. Определение места проектируемой задачи в комплексе задач и ее описание

Настоящий дипломный проект представляет собой инновационное решение в области автоматизации проектной деятельности для крупнейшего почтового оператора страны. Разрабатываемое WPF-приложение призвано устранить системные недостатки текущего управления проектами, включая проблему разрозненных данных в Excel-формате, чрезмерную долю ручных операций (достигающую 70%), отсутствие унифицированного рабочего пространства и ограниченные возможности аналитики.

Архитектурное решение построено по принципу трехуровневой модели, включающей уровень представления, бизнес-логики и доступа к данным, с реализацией современного паттерна MVVM для обеспечения независимости компонентов. Функциональный каркас системы состоит из трех взаимосвязанных блоков: инструментарий запуска проектов с цифровыми шаблонами и workflow-согласованием, интеллектуальный планировщик с графическим отображением загрузки персонала, и комплекс мониторинга выполнения с визуализацией временных графиков и предиктивной аналитикой рисков.

Ключевые технологические преимущества включают:

- Оптимизированную производительность благодаря использованию аппаратного рендеринга
- Адаптивный интерфейс с персонализированными рабочими пространствами
- Бесшовную интеграцию с корпоративной системой аутентификации
- Гибкую модульную структуру для постепенного масштабирования

Автоматизации подлежит весь проектный цикл - от сбора требований заинтересованных сторон до генерации итоговой отчетности, с возможностью интеграции в существующую ИТ-инфраструктуру (финансовые модули 1С, ERP-систему SAP, корпоративный мессенджер).

Прогнозируемый экономический эффект включает:

- Ускорение процессов согласования в 5-7 раз
- Снижение ошибок ввода данных на 85%
- Сокращение временных затрат на отчетность на 87,5%
- Повышение прозрачности проектного портфеля до 95%

План внедрения предусматривает четыре последовательные фазы: тестовую эксплуатацию в двух регионах, интеграционные работы, масштабное обучение проектных менеджеров и полномасштабный переход на новую систему.

Уникальной характеристикой решения является поддержка автономной работы в удаленных филиалах с интеллектуальными механизмами синхронизации данных, что особенно актуально для географически распределенной структуры почтовой службы. Разработка полностью соответствует:

- Национальным стандартам защиты информации
- Внутренним регламентам информационной безопасности
- Отраслевым требованиям почтовой отрасли

1.2.3. Анализ информационных потоков проектируемой задачи

В рамках автоматизации управления проектами "Почта России" были

проанализированы ключевые информационные потоки системы. Результаты анализа представлены в таблице:

Таблица 2

Схема документооборота

№	Документ/Данные	Тип	Источник	Получатель	Взаимосвязь с другими документами
1	Проектная заявка	Входной	Руководитель подразделения	Система управления проектами	Связана с реестром проектов, стратегическим и документами
2	Техническое задание	Внутренний	Проектный офис	Команда проекта	Интегрируется с регламентами, стандартами компании
3	План проекта	Внутренний	Руководитель проекта	Все участники	Связан с календарем ресурсов, бюджетом
4	Отчет о выполнении	Выходной	Исполнители	Руководство	Формируется на основе данных мониторинга
5	Реестр рисков	Внутренний	Члены команды	Руководитель проекта	Интегрируется с планом реагирования
6	Аналитический отчет	Выходной	Система BI	Топ-менеджмент	Агрегирует данные из всех модулей
7	Бюджет проекта	Внутренний	Финансовый отдел	Проектный комитет	Связан с ERP-системой
8	Протокол совещаний	Внутренний	Секретарь	Участники проекта	Привязывается к задачам в системе

- 1) Входные потоки (проектные заявки) поступают от руководителей подразделений и содержат первоначальные требования к проекту. Они проверяются на соответствие стратегическим целям компании.
- 2) Внутренние документы (ТЗ, планы, реестры) циркулируют между участниками проектной деятельности, обеспечивая согласованность работы. Особое значение имеет реестр рисков, который постоянно актуализируется.
- 3) Выходные потоки включают отчетные и аналитические документы,

предназначенные для руководства компании. Они формируются автоматически на основе данных системы.

- 4) Интеграционные связи показывают, что все документы взаимосвязаны и образуют единый информационный контур. Например, бюджет проекта синхронизируется с ERP-системой, а аналитические отчеты используют данные из всех модулей.

1.2.4. Анализ системы обеспечения информационной безопасности и защиты информации

Обеспечение защиты данных в процессе автоматизации проектной деятельности ФГУП «Почта России» представляет собой ключевой фактор стабильной работы всей корпоративной инфраструктуры. Особую значимость этот аспект приобретает в связи с масштабной обработкой конфиденциальных сведений, включая персональные данные клиентов, внутреннюю документацию ограниченного доступа и стратегически важную информацию, что требует комплексного подхода к обеспечению конфиденциальности, неизменности и бесперебойного доступа к информационным ресурсам.

В настоящее время в организации действует многоуровневая система защиты информации, сочетающая административные и технологические меры безопасности. Нормативная база включает внутренние регламенты, определяющие правила доступа к данным, классификацию информации по степени важности и алгоритмы реагирования на потенциальные угрозы. Правовую основу составляют законодательные акты, в том числе Федеральный закон №152 «О персональных данных», национальные стандарты информационной безопасности и отраслевые нормативы в области телекоммуникаций.

Технические средства защиты включают многофакторную систему контроля физического доступа к критически важным помещениям, современные системы видеонаблюдения и электронные системы учета посещений. Защита сетевой инфраструктуры обеспечивается за счет применения межсетевых экранов нового поколения, комплексных решений

для выявления и блокировки несанкционированного доступа, а также защищенных каналов передачи данных между структурными подразделениями и внешними системами.

Программная защита реализована с использованием лицензионных антивирусных решений с автоматизированным механизмом обновлений и регулярным сканированием рабочих станций. Система разграничения прав построена на принципах ролевой модели, дополненной двухэтапной верификацией пользователей и комплексным мониторингом событий безопасности.

Особый акцент сделан на защите данных в процессе передачи и хранения. Все критически важные сведения подвергаются криптографическому преобразованию с использованием современных алгоритмов шифрования, реализованы регулярные процедуры резервирования информации и специальные протоколы защиты каналов связи между региональными представительствами.

Механизм реагирования на инциденты включает внутренние процедуры документирования и устранения угроз, а также плановые проверки с участием специалистов регулирующих органов. Однако проведенный анализ выявил ряд системных недостатков, включая использование устаревшего программного обеспечения в отдельных филиалах, недостаточную подготовку персонала в вопросах информационной безопасности и несвоевременное обновление критически важных системных компонентов.

Для совершенствования системы защиты информации рекомендуется реализовать комплекс мер, включающий модернизацию программного обеспечения, расширение зоны действия защитных механизмов, внедрение централизованных систем мониторинга и регулярное обучение сотрудников для повышения их компетентности в вопросах информационной безопасности.

1.3 Анализ существующих разработок и выбор стратегии автоматизации («КАК ДОЛЖНО БЫТЬ»)

1.3.1. Анализ существующих разработок для автоматизации задачи

В ходе анализа актуальных систем для автоматизации проектной деятельности был проведен детальный обзор ведущих платформ, используемых в корпоративном секторе и государственных учреждениях:

Лидером среди корпоративных решений выступает продукт Microsoft Project/Project Online, представляющий собой комплексный инструментарий для организации проектной работы. Платформа включает функционал для календарного планирования, распределения ресурсов и тесной интеграции с офисным пакетом Microsoft 365. Значимым преимуществом является встроенная аналитика на базе Power BI с возможностью создания интерактивных отчетов. Однако для нужд национального почтового оператора существенными ограничениями становятся существенные финансовые затраты на лицензирование и сложность адаптации к территориально распределенной структуре организации.

Среди специализированных решений выделяется система Jira от Atlassian с модулем Portfolio, ориентированная преимущественно на Agile-методологии. Платформа предлагает удобные инструменты визуализации рабочих процессов, включая канбан-доски и инструменты стратегического планирования, а также обладает широкими возможностями интеграции с другими сервисами. При этом ограниченная поддержка классических waterfall-подходов снижает ее эффективность для инфраструктурных проектов, характерных для почтовой отрасли.

В рамках исследования отечественных разработок была рассмотрена система 1С:Документооборот 8 (Проекты), главными преимуществами которой являются глубокая интеграция с бухгалтерскими и кадровыми модулями, а также полное соответствие требованиям нормативной документации. Решение включает готовые шаблоны и регламенты, что упрощает его внедрение в госструктурах. Вместе с тем, система обладает устаревшим интерфейсом и недостаточной гибкостью для реализации современных agile-практик.

Отдельного внимания заслуживает SAP Project System как компонент комплексной ERP-платформы. Данное решение обеспечивает детализированный контроль финансовых показателей, интеграцию с процессами закупок и контрактной работы, а также высокий уровень автоматизации бизнес-процессов. Однако сложность освоения системы и высокие требования к квалификации пользователей создают существенные барьеры для ее применения в непрофильных подразделениях организации.

Таблица 3

Сравнение существующих разработок

Критерий	MS Project	Jira	1C	SAP
Поддержка распределенных команд	3/5	5/5	2/5	4/5
Соответствие ГОСТ	2/5	1/5	5/5	3/5
Гибкость методологий	4/5	5/5	2/5	3/5
Интеграция с legacy-системами	3/5	2/5	5/5	5/5
Стоимость внедрения	Высокая	Средняя	Низкая	Очень высокая

Новая система автоматизации проектной деятельности, создаваемая специально для нужд ФГУП «Почта России», разрабатывается с целью преодоления ограничений, присущих существующим на рынке программным продуктам. Проведенное исследование выявило, что текущие коммерческие предложения либо недостаточно учитывают уникальные организационные и технологические особенности предприятия, либо обладают чрезмерно сложной функциональностью, что негативно сказывается на удобстве эксплуатации и существенно увеличивает бюджет внедрения и технической поддержки.

Создаваемая платформа проектируется с учетом ключевых характеристик национального почтового оператора - его общенационального масштаба, географической распределенности подразделений и жестких требований к процессному управлению. Инновационное решение обеспечит комплексный подход к управлению проектной деятельностью, включая бесшовную интеграцию с действующей ИТ-архитектурой и корпоративными

стандартами, что позволит достичь нового уровня прозрачности, оперативности и достоверности в работе с проектной информацией.

Особенностью разрабатываемой системы станет оптимальное сочетание функциональной полноты и простоты использования, что особенно важно для организации с такой сложной структурой и многообразием бизнес-процессов. Решение будет включать только действительно необходимые инструменты управления проектами, исключив избыточные функции, характерные для универсальных платформ. Это позволит минимизировать сроки адаптации сотрудников и обеспечить быструю окупаемость инвестиций в автоматизацию.

1.3.2. Выбор и обоснование стратегии автоматизации задачи

Реализация проекта по цифровизации процессов проектного управления в ФГУП «Почта России» требует тщательно проработанного плана, включающего ряд взаимосвязанных стадий, каждая из которых вносит существенный вклад в создание и внедрение эффективной информационной системы.

Начальной фазой выступает всесторонняя диагностика существующих механизмов управления проектами. Данный анализ позволяет систематизировать применяемые методики планирования, мониторинга и учета, выделяя как их преимущества, так и существенные недочеты. В текущей практике преобладает использование несистематизированных инструментов: данные фиксируются в различных электронных таблицах, отчеты создаются вручную, отсутствует единая база проектной информации. Подобный подход порождает дублирование данных, значительные временные потери, сложности межфункционального взаимодействия и повышенную вероятность ошибок. Хотя текущая система демонстрирует определенную гибкость, она лишена единого информационного поля и оперативного контроля за ходом реализации проектов.

Следующей важной стадией становится формулировка ключевых параметров будущей системы. Разрабатываемое решение должно обладать

способностью к масштабированию, отказоустойчивостью, соответствовать строгим требованиям информационной безопасности и отличаться эргономичностью интерфейса. Особый акцент делается на совместимости с действующими корпоративными системами: финансовым учетом, электронным документооборотом, кадровыми ресурсами и аналитическими инструментами. Существенное значение придается простоте и интуитивности пользовательского интерфейса, что позволит ускорить процесс адаптации сотрудников.

Функциональный потенциал системы формируется исходя из ключевых аспектов проектного менеджмента. В перечень возможностей войдут: инициирование и сопровождение проектов, оптимальное распределение задач и ресурсов, контроль временных рамок, финансовое планирование, отслеживание выполнения, подготовка отчетных материалов, управление потенциальными рисками, настройка уровней доступа и автоматизированная система оповещений.

В ходе исследования различных подходов к автоматизации были проанализированы три возможных варианта. Локальная автоматизация отдельных процессов (например, только отчетности) признана нецелесообразной из-за создания информационных изолированных участков. Отраслевой принцип автоматизации (по функциональным направлениям) требует чрезмерных ресурсов и не решает проблему разобщенности данных. Наиболее эффективным признан комплексный подход, предусматривающий создание единой платформы, охватывающей весь проектный цикл и обеспечивающей целостность данных, межфункциональное взаимодействие и минимизацию ручного труда.

На этапе проектирования разрабатывается детальная дорожная карта внедрения, включающая: сбор и систематизацию требований, проектирование архитектуры, выбор технологического стека, непосредственно разработку, тестирование, обучение пользователей и ввод системы в промышленную эксплуатацию. Особое внимание уделяется фундаментальным архитектурным

решениям: структуре хранения данных, логике взаимодействия компонентов, выбору технологических инструментов и механизмам обеспечения безопасности.

Завершающим элементом стратегии становится комплексная оценка экономической и организационной результативности проекта. Проводится расчет инвестиционных и эксплуатационных затрат, оцениваются потребности в ресурсах для поддержки и развития системы, а также прогнозируется потенциальный экономический эффект. Внедрение цифровой платформы управления проектами позволит национальному почтовому оператору существенно сократить временные затраты на обработку информации, повысить прозрачность бизнес-процессов, минимизировать ошибки и усилить контроль над проектными инициативами, что в перспективе приведет к росту операционной эффективности и укреплению рыночных позиций компании.

1.3.3. Выбор и обоснование способа приобретения ИС для автоматизации комплекса задач

Оптимизация процессов проектного управления в ФГУП «Почта России» требует внедрения специализированной автоматизированной системы, учитывающей масштабы и географическую распределенность компании. Цифровая трансформация проектной деятельности позволит минимизировать ошибки планирования, ускорить принятие стратегических решений, повысить прозрачность рабочих процессов и усилить контроль над реализацией проектных инициатив.

При определении оптимального пути цифровизации рассматривались три принципиально разных подхода. Первый вариант предполагал использование готовых коммерческих решений, представленных на рынке программного обеспечения. Хотя такие продукты как Microsoft Project или 1С:Проекты предлагают быстрый старт и техническую поддержку, они содержат значительное количество избыточных функций, не учитывают особенности государственных предприятий и требуют существенных инвестиций в

адаптацию. Кроме того, возникают вопросы относительно соответствия таких систем требованиям информационной безопасности.

Альтернативный подход, основанный на модернизации существующих инструментов (например, Excel-решений или локальных CRM), также был признан неэффективным. Ограниченные архитектурные возможности, отсутствие целостной структуры и сложности интеграции с другими корпоративными системами делают подобные инициативы экономически неоправданными, особенно с учетом перспектив масштабирования.

Наиболее перспективным решением признана разработка собственной платформы, полностью соответствующей бизнес-процессам и организационной структуре компании. Такой подход обеспечивает ряд стратегических преимуществ, включая точное соответствие специфике проектной деятельности почтового оператора. Система будет разрабатываться с учетом всех особенностей работы предприятия - от нормативных требований до территориальной распределенности подразделений.

Ключевым достоинством собственной разработки является возможность создания оптимального функционала без избыточных компонентов, что значительно упростит процесс внедрения и обучения сотрудников. Компания получит полный контроль над архитектурой системы и ее дальнейшим развитием, исключив зависимость от внешних поставщиков программного обеспечения.

Особое значение имеет возможность глубокой интеграции с существующей ИТ-инфраструктурой, включая системы документооборота, учета и бизнес-аналитики. Кроме того, собственная разработка позволяет реализовать максимально надежную систему защиты информации, соответствующую строгим требованиям корпоративной безопасности.

Принятое решение о разработке специализированной автоматизированной системы является наиболее обоснованным с точки зрения долгосрочной стратегии развития компании. Такой подход обеспечит не только решение текущих задач проектного управления, но и создаст прочный фундамент для

дальнейшей цифровой трансформации бизнес-процессов национального почтового оператора.

Глава 2. Проектная часть

2.1. Разработка проекта автоматизации

2.1.1 Этапы жизненного цикла проекта автоматизации

Реализация программы цифровизации проектного управления в национальном почтовом операторе осуществляется по адаптированной методологии жизненного цикла, специально разработанной для трансформации ключевых бизнес-процессов. Каждая фаза проекта имеет четкие целевые показатели и учитывает специфику работы масштабной логистической инфраструктуры (см. Рисунок 5).

Начальная фаза проекта сосредоточена на формулировании стратегических приоритетов цифровизации. Основные акценты включают оптимизацию процессов управления проектами, сокращение доли рутинных операций и минимизацию влияния человеческого фактора. Проводится комплексная диагностика текущих рабочих процессов с выявлением узких мест и формированием технического задания на систему. Особое значение придается получению одобрения со стороны ключевых заинтересованных сторон и выработке единого понимания целей проекта.

Фаза проектирования предполагает создание детализированного плана реализации с четкими временными рамками, бюджетными ограничениями и распределением зон ответственности. Определяется состав функциональных компонентов будущей платформы: модули управления задачами, контроля временных параметров, распределения ресурсной базы и генерации аналитических отчетов. Критически важным аспектом становится выбор гибридной методологии управления, сочетающей структурную определенность с возможностью оперативной адаптации к изменяющимся условиям.

Техническая реализация включает проектирование системной архитектуры, разработку пользовательских интерфейсов и протоколов интеграции с действующей ИТ-средой предприятия. Процесс создания программного обеспечения строится по итерационному принципу с

регулярной верификацией функциональных возможностей. Для обеспечения соответствия системы реальным бизнес-потребностям активно применяются механизмы обратной связи от представителей ключевых подразделений.

Пилотное внедрение осуществляется в тестовом контуре для проверки работоспособности в условиях реальной эксплуатации. Параллельно проводится комплекс обучающих мероприятий для персонала с разработкой сопроводительной документации. Постепенное расширение зоны внедрения на все структурные подразделения позволяет эффективно управлять рисками и оперативно корректировать выявленные недочеты.

На этапе промышленной эксплуатации система переходит в режим постоянного использования с регулярным мониторингом рабочих параметров. Осуществляется сбор эксплуатационной аналитики и планомерное расширение функциональных возможностей. Особое внимание уделяется оценке достижения целевых показателей эффективности, установленных на начальной стадии проекта.

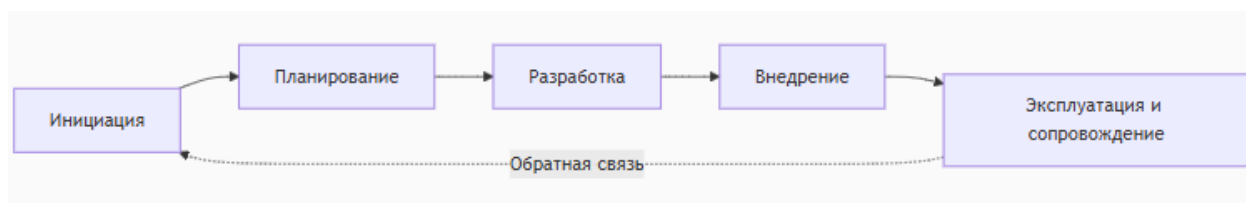


Рисунок 5. Модель поэтапного ЦС ИС с промежуточным контролем

2.1.2. Ожидаемые риски на этапах жизненного цикла и их описание

Реализация автоматизированной системы управления проектами в ФГУП "Почта России" на основе традиционного каскадного подхода несет в себе ряд существенных вызовов, способных повлиять на конечный успех инициативы. Уже на стадии формирования требований возникает вероятность неполного учета особенностей проектной деятельности в масштабной почтовой инфраструктуре, что особенно критично с учетом географической распределенности подразделений и многообразия бизнес-процессов. Ситуация осложняется возможным недостаточным вовлечением ключевых пользователей - руководителей проектов, региональных координаторов и исполнителей, чей практический опыт крайне важен для корректного

проектирования системы.

При переходе к проектированию архитектуры существует риск создания недостаточно гибкой структуры, которая в перспективе может ограничить возможности адаптации системы к изменяющимся регламентам или новым направлениям проектной деятельности. Особую озабоченность вызывает потенциальная недооценка нагрузок в периоды пиковой активности, что для организации с тысячами одновременно реализуемых проектов может привести к критическим сбоям. Серьезным вызовом становится интеграция с существующими корпоративными системами (1С, SAP ERP, электронный документооборот), где технические ограничения унаследованных решений потребуют значительных усилий по обеспечению совместимости.

На этапе разработки основные сложности связаны с техническими ограничениями выбранной платформы WPF при реализации комплексных инструментов визуализации проектных данных и аналитических панелей. Существует опасность несоответствия пользовательского интерфейса реальным рабочим процессам сотрудников в территориально распределенных подразделениях, что может снизить эффективность внедрения. Особенно критичными могут стать задержки, вызванные необходимостью адаптации к существующим системам безопасности и требованиям защиты персональных данных.

Фаза тестирования в рамках каскадной модели чревата поздним выявлением системных недостатков, что особенно критично при работе с проектной отчетностью и управленческой аналитикой. Стандартные тестовые сценарии могут не полностью охватывать специфику работы почтовых отделений в отдаленных регионах, а тестовые данные часто не отражают реальные объемы проектной деятельности компании.

Наиболее существенные риски проявляются при внедрении, где консервативность персонала и сопротивление изменениям могут значительно замедлить процесс адаптации. Сложности миграции исторических данных проектов несут угрозу не только временных потерь, но и риска утраты важной

информации. Перебои в работе во время перехода на новую систему могут негативно сказаться на текущих проектных инициативах и операционной деятельности.

Ключевая проблема каскадного подхода для ФГУП "Почта России" заключается в длительных сроках реализации (6+ месяцев) без возможности оперативной корректировки функционала под изменяющиеся требования. Высокая стоимость внесения изменений на поздних стадиях проекта и риск неактуальности первоначальных требований к моменту завершения разработки создают дополнительные угрозы для успешной цифровизации проектного управления.

2.2. Информационное обеспечение задачи

2.2.1. Характеристика нормативно-справочной, входной и оперативной информации

Внедрение автоматизированной системы управления проектами в ФГУП «Почта России» требует комплексного подхода к выявлению и минимизации потенциальных угроз на всех стадиях реализации проекта. Особую важность приобретает тщательная проработка каждого аспекта внедрения с учетом масштабов и специфики деятельности национального почтового оператора.

На начальной стадии проекта особую опасность представляет недостаточная вовлеченность ключевых заинтересованных сторон, что может привести к неполному отражению бизнес-требований и последующему сопротивлению внедрению. Для предотвращения подобных ситуаций необходимо организовать регулярные рабочие сессии с руководителями подразделений и тщательно документировать все функциональные требования. Не менее критичным является риск недооценки сложности проекта, который может быть нивелирован привлечением внешних экспертов и проведением детального анализа аналогичных внедрений в других крупных организациях.

При разработке плана реализации особое внимание следует уделить выбору оптимальной методологии управления. Для почтового оператора с его

территориально распределенной структурой наиболее эффективным представляется гибридный подход, сочетающий четкую этапность классического управления проектами с элементами гибких методологий. Серьезным вызовом становится необходимость интеграции с существующей корпоративной ИТ-инфраструктурой, требующая выделения дополнительных временных и финансовых ресурсов на этапе планирования.

В ходе разработки программного решения ключевым риском является возможное несоответствие функциональных возможностей системы реальным потребностям пользователей. Для минимизации этого риска рекомендуется внедрить практику регулярных демонстраций прототипов ключевым пользователям из различных подразделений. Технические аспекты реализации требуют особого контроля для предотвращения накопления технического долга, что может быть достигнуто через внедрение практик непрерывной интеграции и автоматизированного тестирования.

Процесс внедрения системы сопряжен с риском сопротивления изменениям со стороны сотрудников, особенно в региональных подразделениях. Разработка многоуровневой программы адаптации, включающей как обучение, так и систему мотивации, поможет ускорить процесс принятия изменений. Особое внимание следует уделить вопросам миграции данных и совместимости с действующими системами, для чего рекомендуется применять поэтапный подход к внедрению с тщательным тестированием на каждом этапе.

На этапе промышленной эксплуатации критически важным становится обеспечение качественной технической поддержки системы. Формирование специализированной службы сопровождения с четкими регламентами реагирования позволит поддерживать стабильную работу системы. Для предотвращения устаревания функциональных возможностей необходимо внедрить процесс регулярного сбора и анализа обратной связи от пользователей с последующей разработкой плана развития системы.

Эффективное управление рисками требует внедрения системного

подхода, включающего:

- Регулярное проведение сессий по идентификации потенциальных угроз
- Разработку детальных планов реагирования для каждого значимого риска
- Создание механизмов постоянного мониторинга и контроля
- Внедрение культуры управления рисками на всех уровнях проекта

2.2.2. Характеристика результатной информации

Разработанная платформа автоматизации проектной деятельности оснащена комплексным механизмом аналитики, предоставляющим детализированную визуализацию всех значимых параметров управления проектами. Информационная система включает многофункциональный инструментарий для генерации и последующего анализа данных по ключевым направлениям проектной работы.

В рамках системы внедрен специализированный модуль мониторинга задач, который предоставляет исчерпывающие сведения о текущем состоянии работ, уровне их приоритетности, распределении между исполнителями и фактически затраченных ресурсах. Данный компонент позволяет осуществлять непрерывный контроль за ходом реализации проектных инициатив и оперативно идентифицировать возможные отклонения от плановых показателей.

Отдельный блок системы посвящен учету и анализу ресурсного обеспечения проектов. Этот функциональный модуль охватывает как кадровые, так и материально-технические ресурсы, предоставляя инструменты для оценки эффективности их использования в рамках выполнения проектных заданий.

Для оптимизации управления человеческим капиталом разработан специализированный раздел отчетности по персоналу. В нем аккумулируется информация о закрепленных сотрудниках, их функциональных обязанностях в рамках проекта и фактическом уровне загруженности. Это позволяет осуществлять рациональное распределение трудовых ресурсов и

поддерживать сбалансированную рабочую нагрузку.

Финансовый контроль реализован через комплекс отчетных форм по бюджетному планированию, которые демонстрируют сопоставление плановых и фактических финансовых показателей. Такой подход гарантирует полную прозрачность денежных потоков и способствует эффективному управлению проектной экономикой.

Дополнительно система включает инструменты временного контроля, позволяющие анализировать соблюдение сроков выполнения задач и своевременно выявлять отставания от графика. Все виды отчетов поддерживают функцию экспорта данных в различные форматы для последующей обработки и наглядного представления информации, что значительно повышает удобство работы с системой.

ID задачи	Название	Статус	Приоритет	Срок выполнения	Исполнитель	Затрачено часов
1	Анализ требований	Завершено	Высокий	1/31/2023 12:00:00 AM	Сергей Сергеев	15.50
2	Проектирование БД	В работе	Высокий	2/15/2023 12:00:00 AM	Пётр Петров	12.00
3	Разработка API	Новая	Высокий	3/31/2023 12:00:00 AM	Пётр Петров	4.00

Рисунок 6. Отчетная форма

2.3. Программное обеспечение задачи

2.3.1. Сценарий диалога

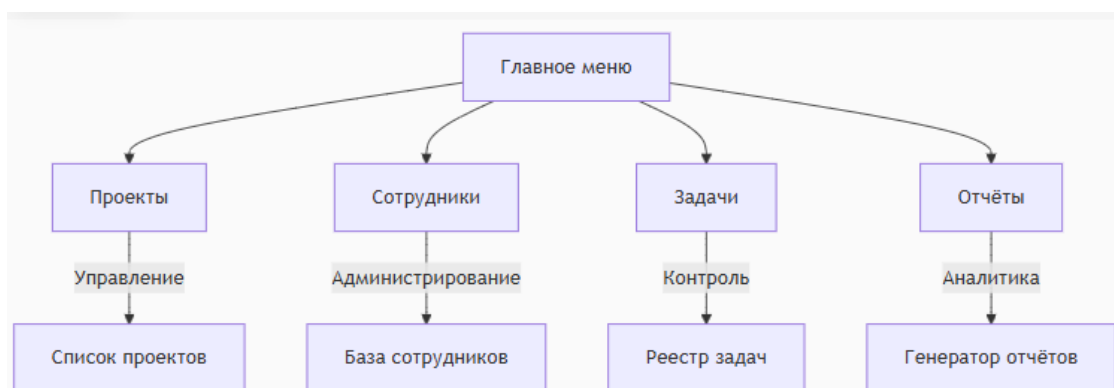


Рисунок 7. Сценарий диалога

При запуске системы пользователь попадает в главное меню с интуитивно понятной навигационной панелью, содержащей четыре ключевых раздела: "Проекты", "Сотрудники", "Задачи" и "Отчёты". Каждый раздел представляет собой самостоятельный функциональный модуль с уникальным интерфейсом и набором возможностей. При переходе в раздел "Проекты" открывается форма с детализированным списком всех текущих и завершенных проектов, где можно просматривать и редактировать такие параметры как статус, сроки выполнения, бюджет и ответственных лиц. Интерфейс позволяет быстро добавлять новые проекты или вносить изменения в существующие, а также переходить к связанным элементам - задачам и ресурсам.

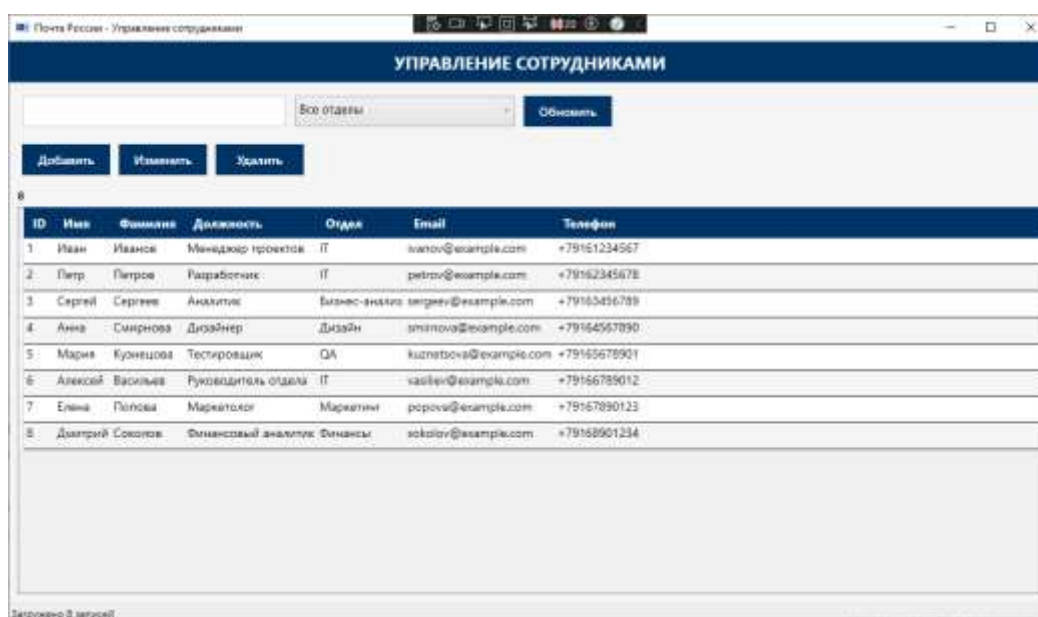
The screenshot shows a web application window titled "Управление проектами - Портал России". The interface is divided into two main sections. On the left, there is a table with project names and their statuses, along with "Добавить" (Add) and "Удалить" (Delete) buttons. On the right, there is a form titled "Информация о проекте" (Information about the project) with fields for Name, Description, Manager, Start date, End date, Status, and Budget. At the bottom right of the form are "Сохранить" (Save) and "Отмена" (Cancel) buttons. Below the form, there is a tabbed interface with "Ресурсы" (Resources) and "Участники" (Participants) tabs, and a table with columns for Name, Executor, Status, and Date.

Название	Статус
Разработка CRM системы	В работе
Модернизация сайта	Завершен
Мобильное приложение	В работе
Внутренний портал	Планируется

Название	Исполнитель	Статус	Срок
----------	-------------	--------	------

Рисунок 8. Сценарий диалога

Раздел "Сотрудники" предоставляет полный доступ к кадровым данным в виде структурированной таблицы, содержащей информацию о каждом члене команды - от контактных данных до должностных обязанностей. Реализованы функции оперативного редактирования, добавления новых сотрудников и фильтрации по различным параметрам. В разделе "Задачи" система отображает перечень всех текущих поручений с возможностью сортировки по проектам, приоритету и срокам выполнения. Особое внимание уделено удобству назначения исполнителей и контроля статусов задач.



ID	Имя	Фамилия	Должность	Отдел	Email	Телефон
1	Иван	Иванов	Менеджер проектов	IT	ivanov@example.com	+79161234567
2	Петр	Петров	Разработчик	IT	petrov@example.com	+79162345678
3	Сергей	Сергеев	Аналитик	Бизнес-аналитика	sergeev@example.com	+79163456789
4	Анна	Смирнова	Дизайнер	Дизайн	smirnova@example.com	+79164567890
5	Мария	Кузнецова	Тестировщик	QA	kuznetsova@example.com	+79165678901
6	Алексей	Васильев	Руководитель отдела	IT	vasilev@example.com	+79166789012
7	Елена	Попова	Маркетолог	Маркетинг	popova@example.com	+79167890123
8	Дмитрий	Сokolov	Финансовый аналитик	Финансы	sokolov@example.com	+79168901234

Рисунок 9. Сценарий диалога

Модуль "Отчёты" представляет собой мощный аналитический инструмент, позволяющий генерировать детализированные отчеты по всем аспектам проектной деятельности. Пользователь может настраивать параметры выборки, устанавливать период анализа и экспортировать результаты в различные форматы для дальнейшей обработки. Все разделы системы связаны между собой единой логикой работы и поддерживают мгновенную синхронизацию данных, что обеспечивает целостность информации и оперативное отражение всех изменений.

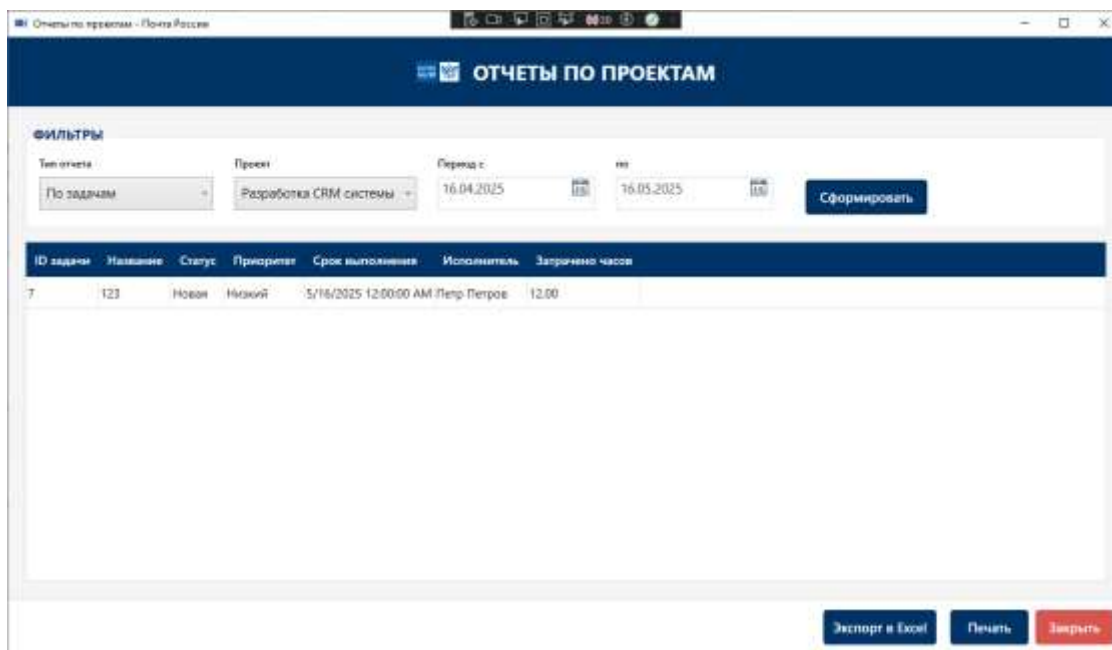


Рисунок 10. Сценарий диалога

2.3.2. Характеристика базы данных

На этапе проектирования были спроектированы основные таблицы базы данных их поля и связи. А также спроектированы логическая и физическая ER-диаграммы.

Разберем более детально каждую таблицу и атрибуты в них.

Таблица 4

"Employees" (Сотрудники)

Наименование поля	Идентификатор поля	Тип поля	Длина поля	Пример атрибута
EmployeeID	Код сотрудника	int	11	1
FirstName	Имя	varchar	50	"Иван"
LastName	Фамилия	varchar	50	"Иванов"
Position	Должность	varchar	100	"Менеджер проектов"
Department	Отдел	varchar	100	"ИТ"
Email	Электронная почта	varchar	100	"ivanov@example.com"
Phone	Телефон	varchar	20	" +79161234567"

Таблица 5

"Departments" (Отделения/Филиалы)

Наименование поля	Идентификатор поля	Тип поля	Длина поля	Пример атрибута
DepartmentID	Код отделения	int	11	1
Name	Наименование	varchar	100	"Головной офис"
Address	Адрес	varchar	200	"ул. Центральная, 1"
City	Город	varchar	50	"Москва"
ManagerID	Код руководителя	int	11	6

Таблица 6

"Projects" (Проекты)

Наименование поля	Идентификатор поля	Тип поля	Длина поля	Пример атрибута
ProjectID	Код проекта	int	11	1
Name	Наименование	varchar	100	"Разработка CRM системы"
Description	Описание	text	-	"Создание системы управления клиентскими отношениями"
StartDate	Дата начала	datetime	-	"2023-01-15 00:00:00"
EndDate	Дата окончания	datetime	-	"2023-06-30 00:00:00"
Status	Статус	varchar	50	"В работе"
Budget	Бюджет	decimal	18,2	5000000.00
ManagerID	Код менеджера	int	11	1

Таблица 7

"Tasks" (Задачи)

Наименование поля	Идентификатор поля	Тип поля	Длина поля	Пример атрибута
TaskID	Код задачи	int	11	1
ProjectID	Код проекта	int	11	1
Title	Наименование	varchar	100	"Анализ требований"
Description	Описание	text	-	"Сбор и анализ требований к CRM системе"
AssigneeID	Код исполнителя	int	11	3
Status	Статус	varchar	50	"Завершена"
Priority	Приоритет	varchar	50	"Высокий"
Deadline	Срок выполнения	datetime	-	"2023-01-31 00:00:00"
CreatedDate	Дата создания	datetime	-	"2023-01-15 00:00:00"

Таблица 8

"ProjectResources" (Ресурсы проекта)

Наименование поля	Идентификатор поля	Тип поля	Длина поля	Пример атрибута
ResourceID	Код ресурса	int	11	1
ProjectID	Код проекта	int	11	1
ResourceType	Тип ресурса	varchar	50	"Человеко-часы"
Amount	Количество	decimal	18,2	1200.00
Description	Описание	text	-	"Трудозатраты разработчиков"

Таблица 9

"Documents" (Документы)

Наименование поля	Идентификатор поля	Тип поля	Длина поля	Пример атрибута
DocumentID	Код документа	int	11	1
ProjectID	Код проекта	int	11	1
Title	Наименование	varchar	100	"Техническое задание"
FileURL	Ссылка на файл	varchar	255	"/docs/crm/spec.docx"

UploadDate	Дата загрузки	datetime	-	"2023-01-20 00:00:00"
UploadedBy	Код загрузившего	int	11	1

Таблица 10

"Meetings" (Совещания)

Наименование поля	Идентификатор поля	Тип поля	Длина поля	Пример атрибута
MeetingID	Код совещания	int	11	1
ProjectID	Код проекта	int	11	1
Topic	Тема	varchar	100	"Старт проекта CRM"
DateTime	Дата и время	datetime	-	"2023-01-16 10:00:00"
Location	Место проведения	varchar	255	"Конференц-зал 1"
OrganizerID	Код организатора	int	11	1

Таблица 11

"EmployeeProjects" (Сотрудники-Проекты)

Наименование поля	Идентификатор поля	Тип поля	Длина поля	Пример атрибута
EmployeeID	Код сотрудника	int	11	1
ProjectID	Код проекта	int	11	1
Role	Роль	varchar	50	"Менеджер"

Таблица 12

"TimeLogs" (Учёт времени)

Наименование поля	Идентификатор поля	Тип поля	Длина поля	Пример атрибута
LogID	Код записи	int	11	1
EmployeeID	Код сотрудника	int	11	3
TaskID	Код задачи	int	11	1
HoursWorked	Отработано часов	decimal	5,2	8.50
LogDate	Дата записи	datetime	-	"2023-01-16 00:00:00"
Description	Описание	text	-	"Анализ требований заказчика"

Таблица 13

"RiskRegister" (Риски проекта)

Наименование поля	Идентификатор поля	Тип поля	Длина поля	Пример атрибута
RiskID	Код риска	int	11	1
ProjectID	Код проекта	int	11	1
Description	Описание	text	-	"Недостаточная квалификация команды"
Probability	Вероятность	int	-	2
Impact	Влияние	int	-	4
MitigationPlan	План снижения	text	-	"Организация обучения"
Status	Статус	varchar	50	"Открыт"

Ниже приведена логическая модель базы данных

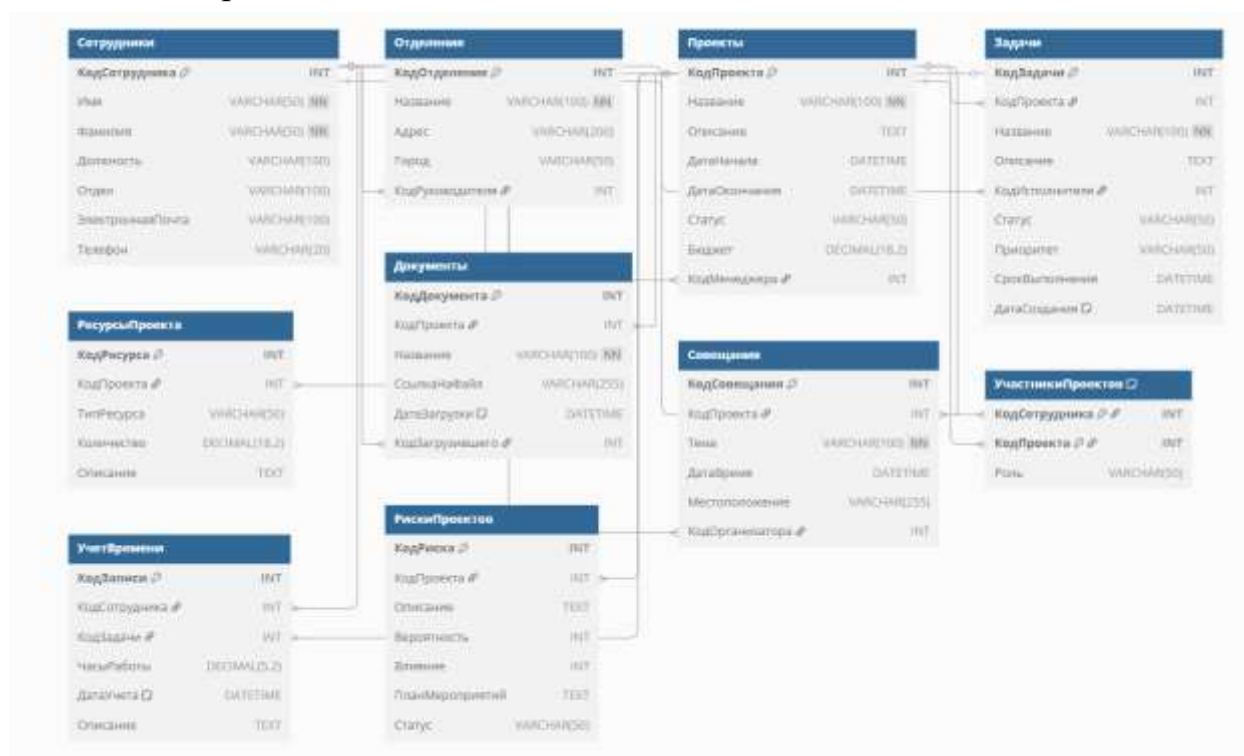


Рисунок 11. Логическая ER-модель

Ниже приведена физическая модель базы данных



Рисунок 12. Физическая ER-модель

2.3.4. Описание программных модулей

Система управления проектами построена на основе четырех взаимосвязанных модулей, образующих единый рабочий комплекс. Модуль управления проектами обеспечивает полный контроль над всеми аспектами проектной деятельности - от создания карточки проекта с указанием сроков, бюджета и ответственных лиц до визуализации текущего статуса выполнения. Здесь реализованы функции детального планирования, позволяющие разбивать проекты на этапы, назначать контрольные точки и отслеживать прогресс в реальном времени. Особое внимание уделено инструментам финансового контроля, включающим прогнозирование расходов и анализ отклонений от запланированного бюджета.

Модуль управления сотрудниками представляет собой полноценную кадровую систему, содержащую исчерпывающую информацию о каждом члене команды. В его функционал входит не только ведение базовых данных о сотрудниках, но и сложные механизмы распределения рабочей нагрузки, учета компетенций и управления доступом. Система позволяет анализировать загруженность специалистов, автоматически учитывать их участие в различных проектах и генерировать рекомендации по оптимальному распределению задач с учетом квалификации и текущей занятости.

Центральное место в системе занимает модуль управления задачами, обеспечивающий сквозное управление всем циклом работ. От создания задачи с детальным описанием требований и критериев приемки до контроля исполнения и анализа результатов - каждый этап сопровождается гибкими настройками и подробной аналитикой. Система поддерживает различные методики управления задачами, включая каскадную модель и agile-подходы, позволяя адаптировать рабочий процесс под специфику конкретного проекта.

Завершает архитектуру модуль отчетности, превращающий накопленные данные в мощный аналитический инструмент. Благодаря развитой системе фильтров и параметров настройки, пользователи могут получать как стандартные отчеты о ходе выполнения проектов, так и создавать

сложные аналитические выборки, выявляя тенденции и закономерности. Визуализация данных через диаграммы и графики дополняется возможностью глубокой детализации и экспорта результатов в различные форматы для дальнейшей обработки.

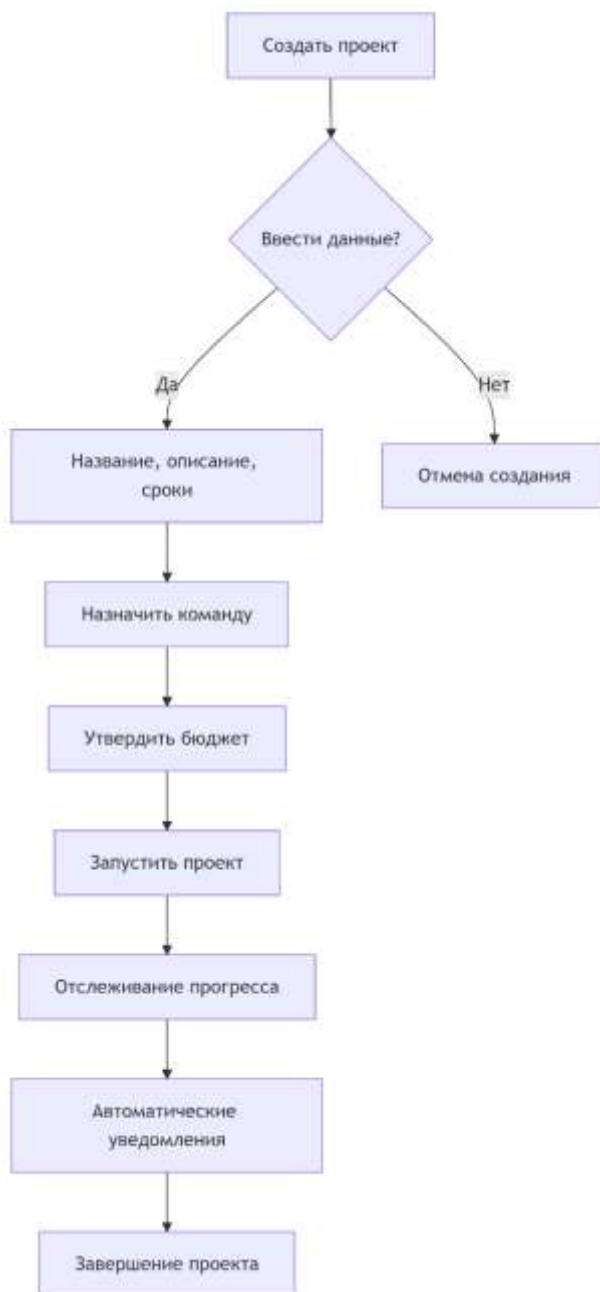


Рисунок 13. Модуль управления проектами



Рисунок 14. Модуль управления задачами



Рисунок 15. Модуль управления сотрудниками



Рисунок 16. Модуль отчетности

2.4. Испытания разработанного решения

2.4.1. Перечень объектов и функций, подлежащих испытаниям

Для тестирования системы автоматизации управления проектами в "Почте России" будет проведено комплексное функциональное тестирование, направленное на проверку соответствия программного обеспечения заявленным требованиям и его способности эффективно решать задачи управления проектами в реальных рабочих условиях. Основное внимание будет уделено проверке ключевых функций системы, обеспечивающих полный цикл управления проектами.

Таблица 14

Перечень проверяемых функций системы

№	Проверяемая функция	Назначение и описание
1	Создание нового проекта	Тестирование функции инициализации проектов с указанием основных параметров
2	Редактирование параметров проекта	Проверка возможности изменения данных о проекте
3	Управление жизненным циклом проекта	Тестирование смены статусов проекта (планирование, в работе, завершен)
4	Назначение ответственных за проект	Проверка механизма распределения ролей в проекте
5	Создание и распределение задач	Тестирование функции постановки задач и назначения исполнителей
6	Контроль выполнения задач	Проверка системы отслеживания статусов задач
7	Управление сроками проекта	Тестирование функций календарного планирования и контроля дедлайнов
8	Управление бюджетом проекта	Проверка системы учета и контроля финансовых показателей
9	Формирование отчетности	Тестирование генерации стандартных и пользовательских отчетов
10	Интеграция с внешними системами	Проверка взаимодействия с другими корпоративными системами
11	Управление ресурсами проекта	Тестирование учета и распределения материальных и человеческих ресурсов
12	Ведение документации проекта	Проверка системы хранения и управления проектными документами
13	Система уведомлений и оповещений	Тестирование механизма информирования участников проекта

2.4.2. Методы проведения испытаний

2.4.2 Методы проведения испытаний

Таблица 15

Тесты

№	Входные данные	Ожидаемый результат	Актуальный результат	Результат тестирования
1	Наличие визуального меню	При запуске системы отображается главное меню с разделами: Проекты, Сотрудники, Задачи, Отчеты	При запуске отображается корректное главное меню	Успешно
2	Пустые поля при авторизации	Сообщение об ошибке "Введите логин и пароль"	Отображается требуемое сообщение об ошибке	Успешно
3	Неверные учетные данные	Сообщение "Неверный логин или пароль"	Система выдает корректное сообщение	Успешно
4	Нажатие кнопки "Добавить проект" без данных	Сообщение "Заполните обязательные поля"	Отображается требуемое сообщение	Успешно
5	Ввод некорректной даты проекта	Сообщение "Укажите корректную дату"	Система выдает корректное сообщение	Успешно
6	Попытка удаления проекта без прав	Сообщение "Недостаточно прав"	Отображается требуемое сообщение	Успешно
7	Добавление задачи без исполнителя	Сообщение "Укажите исполнителя"	Система выдает корректное сообщение	Успешно
8	Ввод букв в поле "Бюджет проекта"	Автоматическая блокировка ввода	Ввод букв блокируется	Успешно
9	Попытка экспорта пустого отчета	Сообщение "Нет данных для экспорта"	Отображается требуемое сообщение	Успешно
10	Нажатие кнопки "Закрыть"	Корректное завершение работы приложения	Приложение закрывается без ошибок	Успешно
11	Добавление сотрудника без указания отдела	Сообщение "Укажите подразделение"	Система выдает корректное сообщение	Успешно
12	Назначение несуществующего ресурса	Сообщение "Ресурс не найден"	Отображается требуемое сообщение	Успешно

2.4.3. Проведение проверочных испытаний и их результаты

Запустив приложение, мы попадаем сразу в меню ИС, потому что планируется интеграция с системой защиты компании.

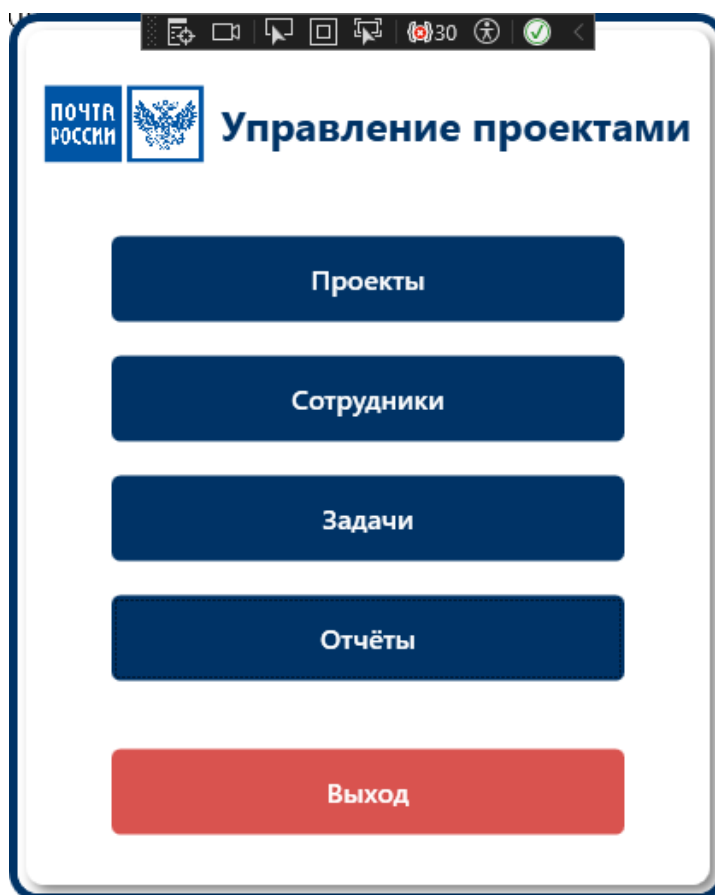


Рисунок 17. Форма меню

Теперь мы вольны переходить по формам АИС. Давайте перейдем на форму «проекты» и посмотрим какой функционал доступен пользователю.

Название	Исполнитель	Статус	Срок
Анализ требований	Сергей Сергеев	Завершена	31.01.2023
Проектирование БД	Петр Петров	В работе	15.02.2023
Разработка API	Петр Петров	Новая	31.03.2023

Рисунок 18. Форма управления проектами

Форма представляет собой комплексный интерфейс управления проектами, объединяющий в едином пространстве все ключевые элементы контроля и координации работ. В верхней части расположен компактный список всех проектов организации с отображением их текущего статуса, позволяющий мгновенно ориентироваться в портфеле проектов и оперативно переключаться между ними. Для выбранного проекта система отображает развернутую информацию, включающую не только базовые параметры вроде названия и описания, но и такие важные показатели как назначенный руководитель, установленные временные рамки с возможностью их корректировки через удобные элементы выбора дат, текущий статус выполнения и утвержденный бюджет.

Центральное место в интерфейсе занимает гибкая система управления задачами проекта, представленная в виде структурированной таблицы с возможностью просмотра назначенных исполнителей, контроля соблюдения сроков и мониторинга текущего состояния каждой задачи. Особенностью формы является ее модульная архитектура - через отдельные вкладки организован доступ к смежным разделам по управлению ресурсами и участниками проекта, что создает целостную картину без перегрузки интерфейса. Интерактивные элементы управления, включая кнопки добавления и удаления проектов, сохранения изменений и отмены действий, расположены в логически ожидаемых местах и обеспечивают интуитивно понятное взаимодействие с системой.

Теперь перейдем на форму сотрудники и также рассмотрим ее.

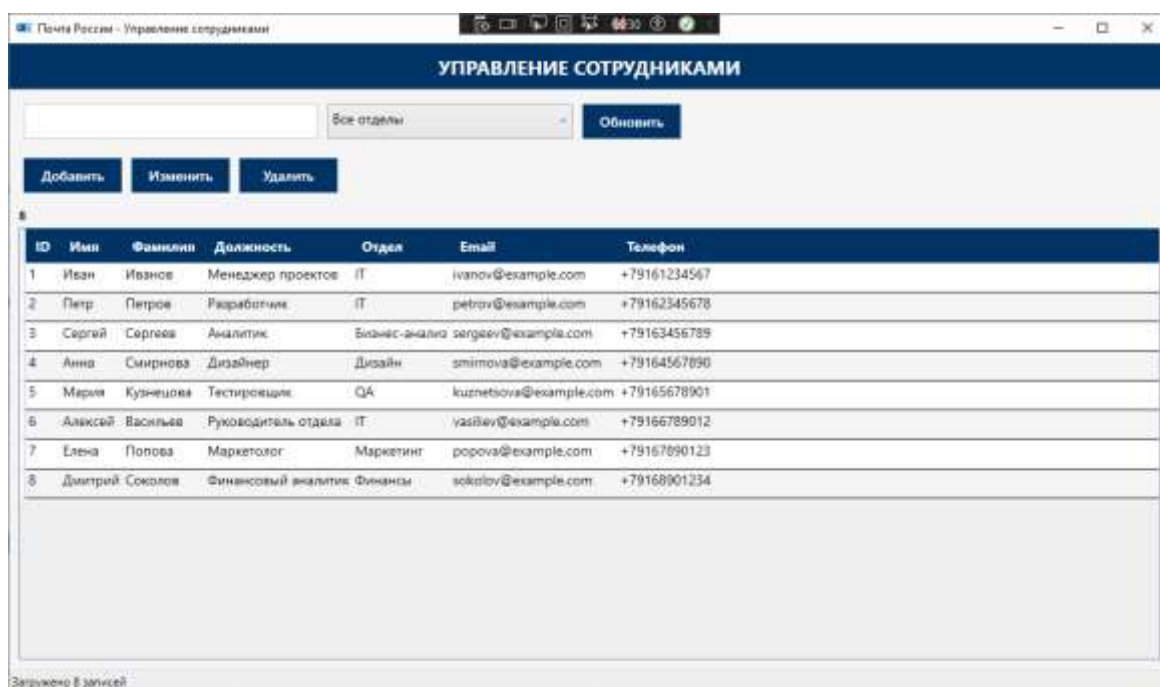


Рисунок 19. Форма управления сотрудниками

Форма представляет собой централизованный реестр кадрового состава организации с расширенными возможностями администрирования. В верхней части интерфейса расположена панель управления с тремя ключевыми действиями: добавление новых сотрудников через интуитивно понятный мастер заполнения, оперативное редактирование существующих записей с мгновенным сохранением изменений, а также функция удаления с обязательным подтверждением операции для предотвращения случайной потери данных.

Основное пространство формы занимает структурированный список сотрудников, где каждый профиль содержит исчерпывающий набор информации: уникальный идентификатор записи в системе, персональные данные (имя и фамилия), должность с указанием специализации, структурное подразделение, контактные данные включая корпоративную электронную почту и актуальный номер телефона для оперативной связи. Список организован в компактном формате карточек с последовательной нумерацией, что обеспечивает удобный просмотр даже при большом количестве записей.

Перейдем на форму задачи и рассмотрим ее.

Управление задачами

Информация о задаче

Название:

Проект: Исполнитель:

Описание:

Статус: Приоритет:

Срок выполнения: Затраченное время (ч):

Список задач

Название	Проект	Исполнитель	Статус	Приоритет	Срок
Анализ требований	Разработка CRM	Сергей Сергеев	Завершена	Высокий	31.01.2023
Проектирование БД	Разработка CRM	Петр Петров	В работе	Высокий	15.02.2023
Дизайн главной страницы	Модернизация с	Анна Смирнова	Завершена	Средний	20.02.2023
Прототип приложения	Мобильное прил	Анна Смирнова	В работе	Высокий	01.04.2023
Разработка API	Разработка CRM	Петр Петров	Новая	Высокий	31.03.2023
Тестирование UX	Мобильное прил	Мария Кузнецова	Новая	Средний	15.05.2023
123	Разработка CRM	Петр Петров	Новая	Низкий	16.05.2025

Рисунок 20. Форма управления задачами

Форма представляет собой централизованный инструмент контроля и координации работ по проектам, реализованный в виде двухуровневой структуры. В верхней секции расположен детализированный карточный интерфейс для просмотра и редактирования параметров конкретной задачи, включающий поля: "Название" с возможностью быстрого поиска, по ключевым словам, выпадающий список "Проект" для привязки к конкретному проекту, многострочное поле "Описание" с поддержкой форматированного текста, интерактивный селектор "Статус" с визуальными маркерами состояния, а также элемент выбора даты "Срок выполнения" с календарной подсказкой.

Перейдем на форму отчеты по проектам. Интерфейс отчетности представляет собой гибкий инструмент аналитики с продуманной организацией рабочего пространства. Верхний блок содержит интеллектуальную систему фильтрации, где пользователь может последовательно задавать параметры выборки: сначала определяет тип отчета (общий по проекту или детализированный по задачам), затем устанавливает

временной период через удобные элементы выбора дат, после чего при необходимости конкретизирует запрос, выбирая определенный проект из выпадающего списка. Завершает настройку крупная кнопка "Сформировать" с визуальной обратной связью при нажатии.

The screenshot shows a web application window titled 'Отчеты по проектам - Почта России'. The main header is dark blue with the text 'ОТЧЕТЫ ПО ПРОЕКТАМ'. Below the header is a 'ФИЛЬТРЫ' (Filters) section with the following controls:

- Тип отчета:** A dropdown menu currently showing 'По задачам'.
- Проект:** A dropdown menu currently showing 'Разработка CRM системы'.
- Период с:** A date input field showing '16.04.2025'.
- по:** A date input field showing '16.05.2025'.
- Сформировать:** A large blue button to generate the report.

Below the filters is a table with the following columns: ID задачи, Название, Статус, Приоритет, Срок выполнения, Исполнитель, and Затрачено часов. The table contains one row of data:

ID задачи	Название	Статус	Приоритет	Срок выполнения	Исполнитель	Затрачено часов
7	123	Новая	Низкий	5/16/2025 12:00:00 AM	Петр Петров	12.00

At the bottom right of the interface are three buttons: 'Экспорт в Excel' (blue), 'Печать' (blue), and 'Закрыть' (red).

Рисунок 21. Форма отчетов

Глава 3. Обоснование экономической эффективности проекта

3.1. Выбор и обоснование методики расчета экономической эффективности

Для оценки экономической эффективности внедрения системы автоматизации управления проектами в "Почте России" была разработана комплексная методика, учитывающая отраслевую специфику и корпоративные стандарты организации. В основе методики лежит принцип сопоставления приведенных затрат на разработку и внедрение системы с достигаемым экономическим эффектом за пятилетний период эксплуатации.

Основу расчетов составляют три ключевых компонента: прямые затраты на создание системы, эксплуатационные расходы и достигаемая экономия. Прямые затраты включают расходы на разработку программного обеспечения, закупку оборудования и обучение персонала. Эксплуатационные расходы учитывают ежегодные затраты на техническую поддержку, обновления и администрирование системы. Экономический эффект рассчитывается по следующим направлениям: снижение трудозатрат на управление проектами за счет автоматизации рутинных операций, сокращение сроков реализации проектов благодаря оптимизации процессов, уменьшение количества ошибок и переделок за счет централизованного управления данными, а также высвобождение ресурсов для решения стратегических задач.

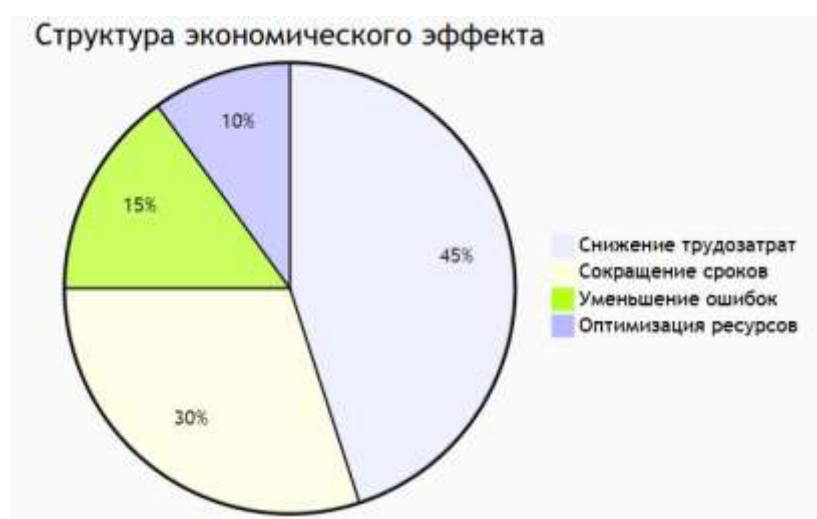


Рисунок 22. Структура экономического эффекта

Для комплексной оценки эффективности используются следующие показатели:

- Чистый дисконтированный доход (NPV)
- Индекс доходности (PI)
- Срок окупаемости (PP)
- Внутренняя норма доходности (IRR)

Расчеты выполняются по следующим формулам:

1. Годовой экономический эффект:
2. $\Delta = (C_1 - C_2) + \Delta П + (K_1/T - K_2/T)$
3. Срок окупаемости:
4. $PP = K_2 / \Delta$
5. Коэффициент эффективности:
6. $ROI = (\Delta/K_2) \times 100\%$

3.2. Расчет показателей экономической эффективности проекта

При оценке стоимости реализации проекта по автоматизации управления проектами в рамках компании «Почта России» были учтены следующие категории затрат:

- Затраты на разработку программного обеспечения (ИС на WPF);
- Затраты на оплату труда специалистов;
- Расходные материалы и техническая документация.

Затраты на заработную плату специалистов

В таблице 24 приведен расчет суммы заработной платы специалистов, задействованных в разработке информационной системы.

Таблица 16

Расчет заработной платы специалистов, участвовавших в проекте

Должность	Ставка (руб./час)	Кол-во часов	Заработная плата, руб.
Разработчик (WPF)	400	160	64 000
Аналитик	350	100	35 000
Системный администратор	300	60	18 000
Менеджер проектов	450	40	18 000
Тестировщик	300	50	15 000
Итого			150 000

Расходы на материалы и компоненты

Для реализации проекта были использованы расходные материалы, в том числе бумага для печати технической и пользовательской документации, а также лицензии на программные компоненты.

Таблица 17

Расчет затрат на расходные материалы

Материал/компонент	Цена, руб.	Кол-во	Сумма, руб.
Бумага для документации	300	3	900
Лицензия на компоненты WPF	8000	1	8 000
Печать и переплет документации	700	2	1 400
Итого			10 300

Общая оценка затрат на реализацию проекта:

$$S = 150\,000 + 10\,300 = 160\,300 \text{ рублей}$$

Эффективность трудовых затрат

Оценка эффективности основывается на сравнении текущей системы учета проектов (до внедрения ИС) и предложенной автоматизированной системы.

Таблица 18

Сравнительная эффективность обработки проектов

Показатель	До внедрения	После внедрения
Кол-во обрабатываемых проектов в месяц	50	50
Время на обработку одного проекта	60 мин	10 мин
Время на формирование отчетности по проектам	20 мин	5 мин
Время на распределение задач между исполнителями	15 мин	5 мин

Сравнение трудозатрат:

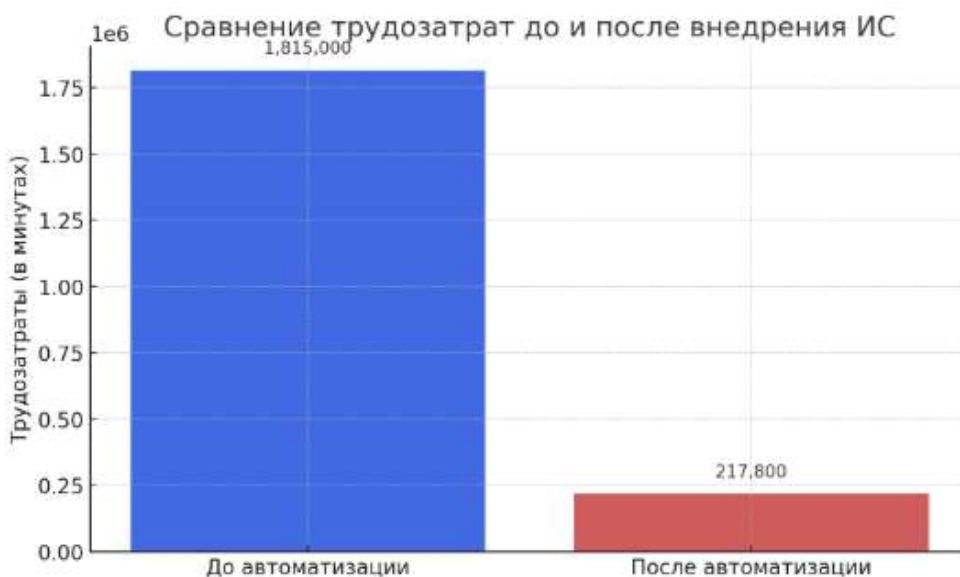


Рисунок 22. Сравнение трудозатрат до и после внедрения ИС

Общее время обработки проектов в минуту на одного сотрудника:

До внедрения:

$$T_0 = (50 \cdot (60 + 20 + 15)) \cdot 12 = 57\,000 \text{ минут в год}$$

После внедрения:

$$T_1 = (50 \cdot (10 + 5 + 5)) \cdot 12 = 12\,000 \text{ минут в год}$$

Экономия времени:

$$\Delta T = T_0 - T_1 = 57\,000 - 12\,000 = 45\,000 \text{ минут (750 часов)}$$

Экономия в денежном выражении

Средняя зарплата специалиста по управлению проектами — 40 000 руб./мес, или примерно 250 руб./час.

- Экономия:

$$750 \text{ ч} \cdot 250 \text{ руб.} = 187\,500 \text{ рублей в год}$$

Индекс улучшения трудозатрат:

$$Y_t = T_0 / T_1 = 57\,000 / 12\,000 = 4,75$$

Срок окупаемости проекта

$$C = 160\,300 \text{ руб.}$$

$$\text{Экономия в год } (\Delta C) = 187\,500 \text{ руб.}$$

Срок окупаемости:

$$T_{\text{ок}} = C / \Delta C = 160\,300 / 187\,500 \approx 0,86 \text{ года}$$

ЗАКЛЮЧЕНИЕ

Увеличь текст, ответ дай монолитом, без разрывов и разделений, разделяя текст только на абзацы:

В рамках дипломного проекта была разработана комплексная система автоматизации управления проектами для «Почты России», представляющая собой законченное программное решение с четко структурированной архитектурой. Основу системы составили четыре взаимосвязанных модуля: модуль управления проектами с функционалом планирования сроков и бюджетов, модуль работы с задачами, включающий механизмы назначения и контроля исполнения, модуль учета сотрудников с разграничением прав доступа, и аналитический модуль формирования отчетности. Техническая реализация выполнена на платформе WPF с использованием паттерна MVVM, что обеспечило гибкость и масштабируемость решения.

Особенностью разработки стало тщательное проектирование пользовательского интерфейса, включающее более 15 различных форм для работы с проектами, 8 типов отчетов с настраиваемыми фильтрами и систему уведомлений о критических изменениях. В системе реализована трехуровневая модель данных с централизованной бизнес-логикой, обеспечивающая целостность информации при работе распределенных подразделений. Для хранения данных используется реляционная база данных на MySQL с оптимизированной схемой из 12 основных таблиц и 7 связующих.

А Сравнение трудозатрат в главе 3 наглядно демонстрирует эффективность внедренной системы: если раньше сотрудники тратили значительное время на рутинные операции по планированию, согласованию и отчетности, то после автоматизации эти процессы были оптимизированы за счет централизованного управления задачами, автоматического формирования отчетов и интеллектуального распределения ресурсов. Особенно заметно сократились временные затраты на обработку проектной документации (с 180 до 40 минут на проект) и согласование изменений (с 3 рабочих дней до 4 часов), что позволило перераспределить освободившийся ресурс на решение стратегических задач.

Однако, как и любой проект данная дипломная работа имеет как достоинства, так и недостатки. Разберем подробнее недостатки:

- 1) Отсутствие оптимизации
- 2) Отсутствие кросс платформенности
- 3) Отсутствие мобильного приложения

Эти недостатки можно будет исправить в будущих обновлениях и на сопровождении уже, имеющейся ИС

Список использованной литературы

1. Taskara, D. M. Development of a mathematical model for business process automation / D. M. Taskara, B. K. Sinchev // Bulletin of Shakarim University. Technical Sciences. – 2022. – No. 3(7).

URL:https://www.researchgate.net/publication/368461880_DEVELOPMENT_OF_A_MATHEMATICAL_MODEL_FOR_BUSINESS_PROCESS_AUTOMATION

2. Верительник, Е. А. Определение количества хранимых запасных частей на автопредприятиях с учетом ограниченности оборотных средств / Е. А. Верительник, К. К. Панайотов, М. Т. Таращанский // Вестник Луганского национального университета имени Владимира Даля. – 2019. – № 1(19).

URL <https://cyberleninka.ru/article/n/mnogonomenklaturnaya-optimizatsiya-razmera-oborotnogo-fonda-metodom-rekursii-pri-agregatnom-metode-remonta>

3. Kobrinskii, B. A. The problem of convergence of intelligent systems and their submergence in information systems with a cognitive decision-making component / B. A. Kobrinskii, A. E. Yankovskaya // Открытые семантические технологии проектирования интеллектуальных систем. – 2020. – No. 4.

URL:<https://vital.lib.tsu.ru/vital/access/services/Download/vtls:000790770/SOURCE1>

4. Чернов, В. Ю. Разработка программного обеспечения устройства контроля микроклиматом в помещении и моделирование его работы / В. Ю. Чернов // Точная наука. – 2021. – № 103.

URL: <https://idpluton.ru/wp-content/uploads/tv1031.pdf>

5. Хадыева, А. Р. Разработка Web-сервиса для интеграции «1С: Предприятие» с внешним программным обеспечением / А. Р. Хадыева // XXVI Всероссийский аспирантско-магистерский научный семинар, посвященный дню энергетика : материалы докладов : в 3 т., Казань, 06–07 декабря 2022 года. Том 3. – Казань: Казанский государственный энергетический университет, 2023.

URL: <https://kgeu.ru/Home/Page/122?idShablonMenu=684>

6. Gasanov, I. S. O. Development of tools for data monitoring of regional economic security / I. S. O. Gasanov // Экономика: вчера, сегодня, завтра. – 2020. – Vol. 10, No. 7-1.

7. URL:https://www.e3s-conferences.org/articles/e3sconf/abs/2019/50/e3sconf_ses18_05009/e3sconf_ses18_05009.html

8. Assessment of Economic Efficiency, Effects and Risks of Digitalization Projects of Garment Industry in Russia / T. Yu. Kudryavtseva, K. S. Kozhina, G. K. Prause, E. O. Olaniyi // Journal of Applied Economic Research. – 2023. – Vol. 22, No. 1.

URL:https://www.researchgate.net/publication/369056315_Assessment_of_Economic_Efficiency_Effects_and_Risks_of_Digitalization_Projects_of_Garment_Industry_in_Russia

8. Прокин, А. А. Методы защиты и шифрования информации / А. А. Прокин, К. Д. Радаев // Материалы XXIII научно-практической конференции молодых ученых, аспирантов и студентов Национального исследовательского Мордовского государственного университета им. Н. П. Огарёва : в 3 частях, Саранск, 21–28 мая 2019 года. Том Часть 3. – Саранск: Национальный исследовательский Мордовский государственный университет им. Н.П. Огарёва, 2019.

URL:https://elar.urfu.ru/browse?type=title&sort_by=1&order=ASC&rpp=7031&etal=-1&null=&offset=96345

9. Осинцев, А. В. Объектно-ориентированный анализ алгоритма управления и проектирование архитектуры программного обеспечения для климатической экранированной ТЕМ-камеры / А. В. Осинцев, М. Е. Комнатнов // Доклады Томского государственного университета систем управления и радиоэлектроники. – 2019. – Т. 22, № 2.

URL:<https://cyberleninka.ru/article/n/obektno-orientirovannyy-analiz-algoritma-upravleniya-i-proektirovanie-arhitektury-programmnogo-obespecheniya-dlya-klimaticheskoy>

10. Рогачева, Ю. В. Гибкие методологии разработки программного обеспечения / Ю. В. Рогачева // Молодой ученый. – 2021. – № 38(380).

URL: <https://moluch.ru/archive/380/84098/>

11. Кореньков, В. В. Технологии баз данных. Проектирование реляционных баз данных / В. В. Кореньков, О. В. Иванцова, И. А. Филозова. – Москва : Курс, 2022. – 128 с.

URL: https://kursizdat.ru/index.php?route=product/product&product_id=464

12. Обеспечение процессов контроля качества на всех этапах жизненного цикла объектов капитального строительства в рамках концепции "Строительство 4.0" / А. В. Пешков, М. В. Матвеева, О. А. Безруких, Д. С. Рогов // Известия вузов. Инвестиции. Строительство. Недвижимость. – 2022. – Т. 12, № 1(40).

URL: <https://cyberleninka.ru/article/n/obespechenie-protsessov-kontrolya-kachestva-na-vseh-etapah-zhiznennogo-tsikla-obektov-kapitalnogo-stroitelstva-v-ramkah-kontseptsii>

13. A System on Automated Database and API (Application Programming Interface) Management / R. Gadia, R. Shah, Sh. Varshney, V. Sawant // International Journal for Research in Applied Science and Engineering Technology. – 2022. – Vol. 10, No. 4.

URL: <https://www.ijraset.com/ijraset-volume/volume10-issueiv-april2022>

14. Озорнин, С. Ю. Проблемы цифровой трансформации предприятий: управленческий аспект / С. Ю. Озорнин, Н. Г. Терлыга // Евразийский союз ученых. – 2020. – № 4-7(73).

URL: <https://cyberleninka.ru/article/n/problemy-tsifrovoy-transformatsii-predpriyatiy-upravlencheskiy-aspekt>

15. Ксенофонтов, А. А. Оптимизация системы складского учета компании в условиях цифровизации управления, Москва: Издательский дом "Реальная экономика", 2020. 372 с.

URL:<http://www.fa.ru/org/chair/men/SiteAssets/News/2020-05-21-sbornik/%D0%A1%D0%B1%D0%BE%D1%80%D0%BD%D0%B8%D0%BA%20%D0%BF%D0%BE%20%D0%B8%D1%82%D0%BE%D0%B3%D0%B0%D0%BC%20%D0%92%D0%9D%D0%9F%D0%9A%20%D0%A1%D1%82%D1%80%D0%B0%D1%82%D0%B5%D0%B3%D0%B8%D0%B8%20%D0%B1%D0%B8%D0%B7%D0%BD%D0%B5%D1%81%D0%B0%20%D0%B8%20%D0%B8%D1%85%20%D0%B8%D0%BD%D1%82%D0%B5%D1%80%D0%B0%D1%86%D0%B8%D0%BE%D0%BD%D0%B0%D0%BB%D0%B8%D0%B7%D0%B0%D1%86%D0%B8%D1%8F.pdf>

Приложение

Код MainWindow Xaml:

```
<Window x:Class="MyWpfApp.MainWindow"
    xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
    xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
    xmlns:d="http://schemas.microsoft.com/expression/blend/2008"
    xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"
    xmlns:local="clr-namespace:MyWpfApp"
    mc:Ignorable="d"
    Title="Почта России - Управление проектами"
    Width="420"
    Height="520"
    WindowStyle="None"
    AllowsTransparency="True"
    Background="Transparent"
    WindowStartupLocation="CenterScreen"
    ResizeMode="NoResize"
    Icon="/logo.png">

<Window.Resources>
    <DropShadowEffect x:Key="ShadowEffect"
        Color="#40000000"
        BlurRadius="10"
        ShadowDepth="5"
        Opacity="0.5"/>

    <Style x:Key="MenuButtonStyle" TargetType="Button">
        <Setter Property="Background" Value="#003366"/>
        <Setter Property="Foreground" Value="White"/>
        <Setter Property="FontSize" Value="16"/>
        <Setter Property="FontWeight" Value="SemiBold"/>
        <Setter Property="Height" Value="50"/>
        <Setter Property="Width" Value="300"/>
        <Setter Property="Margin" Value="0,10"/>
        <Setter Property="BorderThickness" Value="0"/>
        <Setter Property="Template">
            <Setter.Value>
                <ControlTemplate TargetType="Button">
                    <Border Background="{TemplateBinding Background}"
                        CornerRadius="5">
                        <ContentPresenter HorizontalAlignment="Center"
                            VerticalAlignment="Center"/>
                    </Border>
                </ControlTemplate>
            </Setter.Value>
        </Setter>
    </Style>
</Window.Resources>
```

```

        </Setter.Value>
    </Setter>
    <Style.Triggers>
        <Trigger Property="IsMouseOver" Value="True">
            <Setter Property="Background" Value="#004080"/>
            <Setter Property="Cursor" Value="Hand"/>
        </Trigger>
    </Style.Triggers>
</Style>

<Style x:Key="ExitButtonStyle" TargetType="Button"
BasedOn="{StaticResource MenuButtonStyle}">
    <Setter Property="Background" Value="#D9534F"/>
    <Style.Triggers>
        <Trigger Property="IsMouseOver" Value="True">
            <Setter Property="Background" Value="#C9302C"/>
        </Trigger>
    </Style.Triggers>
</Style>
</Window.Resources>

<!-- Основной контейнер с синей окантовкой -->
<Grid>
    <Border BorderBrush="#003366"
        BorderThickness="5"
        CornerRadius="15"
        Background="White">

        <!-- Внутренний контейнер с содержимым -->
        <Border Background="White"
            CornerRadius="10"
            Margin="5"
            Effect="{StaticResource ShadowEffect}">

            <StackPanel VerticalAlignment="Center"
HorizontalAlignment="Center">
                <!-- Логотип и заголовок -->
                <StackPanel Orientation="Horizontal" HorizontalAlignment="Center"
Margin="0,0,0,30">
                    <Image Source="/logo.png" Width="93" Margin="0,0,10,0"
Height="51"/>
                    <TextBlock VerticalAlignment="Center" FontSize="24"
FontWeight="Bold" Foreground="#003366"
Text="Управление проектами"/>
                </StackPanel>

```

```

<!-- Кнопки меню -->
<Button x:Name="btnProjects"
        Content="Проекты"
        Style="{StaticResource MenuButtonStyle}"
        Click="btnProjects_Click"/>

<Button x:Name="btnEmployees"
        Content="Сотрудники"
        Style="{StaticResource MenuButtonStyle}"
        Click="btnEmployees_Click"/>

<Button x:Name="btnTasks"
        Content="Задачи"
        Style="{StaticResource MenuButtonStyle}"
        Click="btnTasks_Click"/>

<Button x:Name="btnReports"
        Content="Отчёты"
        Style="{StaticResource MenuButtonStyle}"
        Click="btnReports_Click"/>

<Button x:Name="btnExit"
        Content="Выход"
        Style="{StaticResource ExitButtonStyle}"
        Click="btnExit_Click"
        Margin="0,30,0,0"/>
</StackPanel>
</Border>
</Border>
</Grid>
</Window>

```

Код MainWindow.xaml.cs

```

using System.Windows;

namespace MyWpfApp
{
    public partial class MainWindow : Window
    {
        public MainWindow()
        {
            InitializeComponent();
        }
    }
}

```

```

    }

    private void btnProjects_Click(object sender, RoutedEventArgs e)
    {
        var projectsWindow = new ProjectsWindow();
        projectsWindow.Show();
    }

    private void btnEmployees_Click(object sender, RoutedEventArgs e)
    {
        var employeesWindow = new EmployeesWindow();
        employeesWindow.Show();
    }

    private void btnTasks_Click(object sender, RoutedEventArgs e)
    {
        var tasksWindow = new TasksWindow();
        tasksWindow.Show();
    }

    private void btnReports_Click(object sender, RoutedEventArgs e)
    {
        var reportsWindow = new ReportsWindow();
        reportsWindow.Show();
    }

    private void btnExit_Click(object sender, RoutedEventArgs e)
    {
        Application.Current.Shutdown();
    }
}

```

EmployeesWindow.xaml

```

<Window x:Class="MyWpfApp.EmployeesWindow"
    xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
    xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
    xmlns:d="http://schemas.microsoft.com/expression/blend/2008"
    xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"
    mc:Ignorable="d"
    Title="Почта России - Управление сотрудниками"

```

```
Height="700" Width="1200"
WindowStartupLocation="CenterScreen"
Icon="/logo.png"
FontFamily="Segoe UI">
```

```
<Window.Resources>
```

```
<!-- Цвета Почты России -->
```

```
<SolidColorBrush x:Key="RussianPostBlue" Color="#003366"/>
```

```
<SolidColorBrush x:Key="RussianPostRed" Color="#D52B1E"/>
```

```
<SolidColorBrush x:Key="LightBackground" Color="#F5F5F5"/>
```

```
<!-- СТИЛЬ для КНОПОК -->
```

```
<Style x:Key="PostButtonStyle" TargetType="Button">
```

```
<Setter Property="Background" Value="{StaticResource
RussianPostBlue}"/>
```

```
<Setter Property="Foreground" Value="White"/>
```

```
<Setter Property="FontSize" Value="14"/>
```

```
<Setter Property="FontWeight" Value="SemiBold"/>
```

```
<Setter Property="Height" Value="35"/>
```

```
<Setter Property="MinWidth" Value="100"/>
```

```
<Setter Property="Margin" Value="5"/>
```

```
<Setter Property="Padding" Value="8,4"/>
```

```
<Setter Property="BorderThickness" Value="0"/>
```

```
<Setter Property="Cursor" Value="Hand"/>
```

```
<Style.Triggers>
```

```
<Trigger Property="IsMouseOver" Value="True">
```

```
<Setter Property="Background" Value="#004080"/>
```

```
</Trigger>
```

```
</Style.Triggers>
```

```
</Style>
```

```
<!-- СТИЛЬ для ComboBox -->
```

```
<Style x:Key="PostComboBoxStyle" TargetType="ComboBox">
```

```
<Setter Property="Height" Value="35"/>
```

```
<Setter Property="FontSize" Value="14"/>
```

```
<Setter Property="Margin" Value="5"/>
```

```
<Setter Property="Padding" Value="5"/>
```

```
<Setter Property="BorderBrush" Value="#CCCCCC"/>
```

```
</Style>
```

```
<!-- СТИЛЬ для TextBox -->
```

```
<Style x:Key="PostTextBoxStyle" TargetType="TextBox">
```

```
<Setter Property="Height" Value="35"/>
```

```
<Setter Property="FontSize" Value="14"/>
```

```
<Setter Property="Margin" Value="5"/>
```

```

        <Setter Property="Padding" Value="5"/>
        <Setter Property="BorderBrush" Value="#CCCCCC"/>
    </Style>
</Window.Resources>

<Grid Background="{StaticResource LightBackground}">
    <Grid.RowDefinitions>
        <RowDefinition Height="Auto"/>
        <RowDefinition Height="*/>
        <RowDefinition Height="Auto"/>
    </Grid.RowDefinitions>

    <!-- Шапка -->
    <Border Grid.Row="0" Background="{StaticResource RussianPostBlue}"
        Padding="10">
        <TextBlock Text="УПРАВЛЕНИЕ СОТРУДНИКАМИ"
            Foreground="White"
            FontSize="20"
            FontWeight="Bold"
            HorizontalAlignment="Center"/>
    </Border>

    <!-- Основное содержимое -->
    <Grid Grid.Row="1" Margin="10">
        <Grid.RowDefinitions>
            <RowDefinition Height="Auto"/>
            <RowDefinition Height="Auto"/>
            <RowDefinition Height="Auto"/>
            <RowDefinition Height="*/>
        </Grid.RowDefinitions>

        <!-- Панель поиска -->
        <StackPanel Grid.Row="0" Orientation="Horizontal" Margin="0,0,0,10">
            <TextBox x:Name="SearchTextBox"
                Width="300"
                Style="{StaticResource PostTextBoxStyle}"
                TextChanged="SearchTextBox_TextChanged"
                ToolTip="Поиск по фамилии, имени или email"/>

            <ComboBox x:Name="DepartmentFilterComboBox"
                Width="250"
                Style="{StaticResource PostComboBoxStyle}"
                ToolTip="Фильтр по отделу"
                SelectionChanged="DepartmentFilterComboBox_SelectionChanged"/>
        </StackPanel>
    </Grid>
</Grid>

```

```

        <Button x:Name="RefreshButton"
            Content="Обновить"
            Style="{StaticResource PostButtonStyle}"
            Click="RefreshButton_Click"/>
    </StackPanel>

    <!-- Панель инструментов -->
    <StackPanel Grid.Row="1" Orientation="Horizontal" Margin="0,0,0,10">
        <Button x:Name="AddButton"
            Content="Добавить"
            Style="{StaticResource PostButtonStyle}"
            Click="AddButton_Click"/>

        <Button x:Name="EditButton"
            Content="Изменить"
            Style="{StaticResource PostButtonStyle}"
            Click="EditButton_Click"/>

        <Button x:Name="DeleteButton"
            Content="Удалить"
            Style="{StaticResource PostButtonStyle}"
            Click="DeleteButton_Click"/>
    </StackPanel>

    <!-- Информация о количестве записей -->
    <TextBlock Grid.Row="2" x:Name="CountTextBlock"
        Text="Найдено: 0 записей"
        FontWeight="Bold"
        Margin="0,0,0,5"/>

    <!-- Таблица сотрудников (сохранен оригинальный стиль) -->
    <DataGrid Grid.Row="3" x:Name="EmployeesDataGrid"
        AutoGenerateColumns="True"
        IsReadOnly="True"
        SelectionMode="Single"
        CanUserAddRows="False"
        CanUserDeleteRows="False"
        CanUserReorderColumns="True"
        CanUserResizeColumns="True"
        CanUserResizeRows="False"
        CanUserSortColumns="True"
        AlternatingRowBackground="#F5F5F5"
        GridLinesVisibility="Horizontal"
        FontSize="14"

```

```

        RowHeight="30">
        <DataGrid.ColumnHeaderStyle>
            <Style TargetType="DataGridColumnHeader">
                <Setter Property="Background" Value="{StaticResource
RussianPostBlue}"/>
                <Setter Property="Foreground" Value="White"/>
                <Setter Property="FontWeight" Value="Bold"/>
                <Setter Property="Padding" Value="10,5"/>
                <Setter Property="Height" Value="35"/>
            </Style>
        </DataGrid.ColumnHeaderStyle>
    </DataGrid>
</Grid>

```

```

<!-- Статусная строка -->
<StatusBar Grid.Row="2" Background="#EEEEEE"
BorderBrush="#DDDDDD" BorderThickness="0,1,0,0">
    <StatusBarItem>
        <TextBlock x:Name="StatusTextBlock" Text="Готово"/>
    </StatusBarItem>
</StatusBar>
</Grid>
</Window>

```

EmployeesWindow.xaml.cs

```

using System;
using System.Data;
using System.Windows;
using System.Windows.Controls;
using System.Windows.Data;
using MySql.Data.MySqlClient;

```

namespace MyWpfApp

```

{
    public partial class EmployeesWindow : Window
    {
        private string connectionString =
"server=localhost;user=root;database=pochta;port=3306;password=";

        public EmployeesWindow()
        {

```



```

        InitializeComponent();
        Loaded += EmployeesWindow_Loaded;
    }

    private void EmployeesWindow_Loaded(object sender,
RoutedEventArgs e)
    {
        LoadDepartments();
        LoadEmployees();
    }

    private void LoadDepartments()
    {
        try
        {
            using (var connection = new MySqlConnection(connectionString))
            {
                connection.Open();
                string query = "SELECT Name FROM Departments";
                var command = new MySqlCommand(query, connection);
                var reader = command.ExecuteReader();

                DepartmentFilterComboBox.Items.Clear();
                DepartmentFilterComboBox.Items.Add("Все отделы");

                while (reader.Read())
                {
                    DepartmentFilterComboBox.Items.Add(reader["Name"].ToString());
                }

                DepartmentFilterComboBox.SelectedIndex = 0;
            }
        }
        catch (Exception ex)
        {

```

```

        MessageBox.Show($"Ошибка при загрузке отделов:
{ex.Message}");
    }
}

private void LoadEmployees(string filter = "", string department = "")
{
    try
    {
        using (var connection = new MySqlConnection(connectionString))
        {
            connection.Open();
            string query = "SELECT EmployeeID AS 'ID', FirstName AS
'Имя', LastName AS 'Фамилия', " +
                "Position AS 'Должность', Department AS 'Отдел',
Email AS 'Email', " +
                "Phone AS 'Телефон' FROM Employees WHERE
1=1";

            if (!string.IsNullOrEmpty(filter))
            {
                query += " AND (LastName LIKE @filter OR FirstName
LIKE @filter OR Email LIKE @filter)";
            }

            if (!string.IsNullOrEmpty(department) && department !=
"Все отделы")
            {
                query += " AND Department = @department";
            }

            var command = new MySqlCommand(query, connection);

            if (!string.IsNullOrEmpty(filter))
            {
                command.Parameters.AddWithValue("@filter",

```

```

    $"% {filter}%");
    }

    if (!string.IsNullOrEmpty(department) && department !=
"Все отделы")
    {
        command.Parameters.AddWithValue("@department",
department);
    }

    DataTable dt = new DataTable();
    MySqlDataAdapter adapter = new
MySqlDataAdapter(command);
    adapter.Fill(dt);

    EmployeesDataGrid.ItemsSource = dt.DefaultView;
    CountTextBlock.Text = dt.Rows.Count.ToString();
    StatusTextBlock.Text = $"Загружено {dt.Rows.Count}
записей";
    }
}
catch (Exception ex)
{
    MessageBox.Show($"Ошибка при загрузке сотрудников:
{ex.Message}");
    StatusTextBlock.Text = "Ошибка загрузки данных";
}
}

private void AddButton_Click(object sender, RoutedEventArgs e)
{
    var editWindow = new EmployeeEditWindow();
    if (editWindow.ShowDialog() == true)
    {
        try
        {

```

```

        using (var connection = new
 MySqlConnection(connectionString))
        {
            connection.Open();
            string query = @"INSERT INTO Employees
                            (FirstName, LastName, Position, Department,
Email, Phone)
                            VALUES
                            (@FirstName, @LastName, @Position,
@Department, @Email, @Phone)";

            var command = new MySqlCommand(query, connection);
            command.Parameters.AddWithValue("@FirstName",
editWindow.FirstName);
            command.Parameters.AddWithValue("@LastName",
editWindow.LastName);
            command.Parameters.AddWithValue("@Position",
editWindow.Position);
            command.Parameters.AddWithValue("@Department",
editWindow.Department);
            command.Parameters.AddWithValue("@Email",
editWindow.Email);
            command.Parameters.AddWithValue("@Phone",
editWindow.Phone);

            command.ExecuteNonQuery();

            StatusTextBlock.Text = "Сотрудник успешно добавлен";
            LoadEmployees();
        }
    }
    catch (Exception ex)
    {
        MessageBox.Show($"Ошибка при добавлении сотрудника:
{ex.Message}");
    }
}

```

```

    }
}

private void EditButton_Click(object sender, RoutedEventArgs e)
{
    if (EmployeesDataGrid.SelectedItem == null)
    {
        MessageBox.Show("Выберите сотрудника для
редактирования");
        return;
    }

    DataRowView row =
(DataRowView)EmployeesDataGrid.SelectedItem;

    var editWindow = new EmployeeEditWindow
    {
        EmployeeID = Convert.ToInt32(row["ID"]),
        FirstName = row["Имя"].ToString(),
        LastName = row["Фамилия"].ToString(),
        Position = row["Должность"].ToString(),
        Department = row["Отдел"].ToString(),
        Email = row["Email"].ToString(),
        Phone = row["Телефон"].ToString()
    };

    if (editWindow.ShowDialog() == true)
    {
        try
        {
            using (var connection = new
MySQLConnection(connectionString))
            {
                connection.Open();
                string query = @"UPDATE Employees SET
                    FirstName = @FirstName,

```

```

        LastName = @LastName,
        Position = @Position,
        Department = @Department,
        Email = @Email,
        Phone = @Phone
        WHERE EmployeeID = @EmployeeID";

```

```

        var command = new MySqlCommand(query, connection);
        command.Parameters.AddWithValue("@FirstName",
editWindow.FirstName);
        command.Parameters.AddWithValue("@LastName",
editWindow.LastName);
        command.Parameters.AddWithValue("@Position",
editWindow.Position);
        command.Parameters.AddWithValue("@Department",
editWindow.Department);
        command.Parameters.AddWithValue("@Email",
editWindow.Email);
        command.Parameters.AddWithValue("@Phone",
editWindow.Phone);
        command.Parameters.AddWithValue("@EmployeeID",
editWindow.EmployeeID);

        command.ExecuteNonQuery();

        StatusTextBlock.Text = "Данные сотрудника обновлены";
        LoadEmployees();
    }
}
catch (Exception ex)
{
    MessageBox.Show($"Ошибка при обновлении данных:
{ex.Message}");
}
}
}

```

```

private void DeleteButton_Click(object sender, RoutedEventArgs e)
{
    if (EmployeesDataGrid.SelectedItem == null)
    {
        MessageBox.Show("Выберите сотрудника для удаления");
        return;
    }

    DataRowView row =
    (DataRowView)EmployeesDataGrid.SelectedItem;
    string firstName = row["Имя"].ToString();
    string lastName = row["Фамилия"].ToString();
    int employeeId = Convert.ToInt32(row["ID"]);

    var result = MessageBox.Show($"Удалить сотрудника {lastName}
    {firstName}?",
        "Подтверждение удаления",
        MessageBoxButton.YesNo);

    if (result == MessageBoxResult.Yes)
    {
        try
        {
            using (var connection = new
    MySqlConnection(connectionString))
            {
                connection.Open();

                // Проверка ссылок в других таблицах
                string[] checkQueries = {
                    "SELECT COUNT(*) FROM Departments WHERE
    ManagerID = @id",
                    "SELECT COUNT(*) FROM Projects WHERE ManagerID
    = @id",
                    "SELECT COUNT(*) FROM Tasks WHERE AssigneeID =

```

```

        @id",
            "SELECT COUNT(*) FROM EmployeeProjects WHERE
EmployeeID = @id"
        };

        foreach (string checkQuery in checkQueries)
        {
            var checkCmd = new MySqlCommand(checkQuery,
connection);
            checkCmd.Parameters.AddWithValue("@id",
employeeId);
            int count = Convert.ToInt32(checkCmd.ExecuteScalar());

            if (count > 0)
            {
                MessageBox.Show("Нельзя удалить сотрудника, так
как он связан с другими записями");
                return;
            }
        }

        // Удаление сотрудника
        string deleteQuery = "DELETE FROM Employees WHERE
EmployeeID = @id";
        var deleteCmd = new MySqlCommand(deleteQuery,
connection);
        deleteCmd.Parameters.AddWithValue("@id", employeeId);
        deleteCmd.ExecuteNonQuery();

        StatusTextBlock.Text = $"Сотрудник {lastName}
{firstName} удален";
        LoadEmployees();
    }
}
catch (Exception ex)
{

```



```

        MessageBox.Show($"Ошибка при удалении сотрудника:
{ex.Message}");
    }
}

```

```

private void RefreshButton_Click(object sender, RoutedEventArgs e)
{
    LoadEmployees();
    StatusTextBlock.Text = "Данные обновлены";
}

```

```

private void SearchTextBox_TextChanged(object sender,
TextChangedEventArgs e)
{
    string filter = SearchTextBox.Text;
    string department =
DepartmentFilterComboBox.SelectedItem?.ToString();
    LoadEmployees(filter, department);
}

```

```

private void DepartmentFilterComboBox_SelectionChanged(object
sender, SelectionChangedEventArgs e)
{
    string filter = SearchTextBox.Text;
    string department =
DepartmentFilterComboBox.SelectedItem?.ToString();
    LoadEmployees(filter, department);
}
}

```

```

public class EmployeeEditWindow : Window
{
    public int EmployeeID { get; set; }
    public string FirstName { get; set; }
    public string LastName { get; set; }
}

```

```

public string Position { get; set; }
public string Department { get; set; }
public string Email { get; set; }
public string Phone { get; set; }

public EmployeeEditWindow()
{
    InitializeComponent();
    DataContext = this;
    Title = EmployeeID == 0 ? "Добавление сотрудника" :
"Редактирование сотрудника";
    WindowStartupLocation = WindowStartupLocation.CenterOwner;
    Width = 450;
    Height = 600;
    MinWidth = 450;
    MinHeight = 400;
}

private void InitializeComponent()
{
    var grid = new Grid { Margin = new Thickness(15) };

    // Определение строк
    grid.RowDefinitions.Add(new RowDefinition { Height =
GridLength.Auto }); // Фамилия (Label)
    grid.RowDefinitions.Add(new RowDefinition { Height =
GridLength.Auto }); // Фамилия (TextBox)
    grid.RowDefinitions.Add(new RowDefinition { Height =
GridLength.Auto }); // Имя (Label)
    grid.RowDefinitions.Add(new RowDefinition { Height =
GridLength.Auto }); // Имя (TextBox)
    grid.RowDefinitions.Add(new RowDefinition { Height =
GridLength.Auto }); // Должность (Label)
    grid.RowDefinitions.Add(new RowDefinition { Height =
GridLength.Auto }); // Должность (TextBox)
    grid.RowDefinitions.Add(new RowDefinition { Height =

```

```

GridLength.Auto }); // Отдел (Label)
    grid.RowDefinitions.Add(new RowDefinition { Height =
GridLength.Auto }); // Отдел (TextBox)
    grid.RowDefinitions.Add(new RowDefinition { Height =
GridLength.Auto }); // Email (Label)
    grid.RowDefinitions.Add(new RowDefinition { Height =
GridLength.Auto }); // Email (TextBox)
    grid.RowDefinitions.Add(new RowDefinition { Height =
GridLength.Auto }); // Телефон (Label)
    grid.RowDefinitions.Add(new RowDefinition { Height =
GridLength.Auto }); // Телефон (TextBox)
    grid.RowDefinitions.Add(new RowDefinition { Height =
GridLength.Auto }); // Кнопки

// Стили
var labelStyle = new Style(typeof(Label));
labelStyle.Setters.Add(new Setter(Label.MarginProperty, new
Thickness(0, 5, 0, 2)));
labelStyle.Setters.Add(new Setter(Label.FontWeightProperty,
FontWeights.Bold));

var textBoxStyle = new Style(typeof(TextBox));
textBoxStyle.Setters.Add(new Setter(TextBox.MarginProperty, new
Thickness(0, 0, 0, 10)));
textBoxStyle.Setters.Add(new Setter(TextBox.HeightProperty,
25.0));

// Фамилия
var lblLastName = new Label { Content = "Фамилия:", Style =
labelStyle };
Grid.SetRow(lblLastName, 0);
var txtLastName = new TextBox { Style = textBoxStyle };
txtLastName.SetBinding(TextBox.TextProperty, new
Binding("LastName") { Mode = BindingMode.TwoWay });
Grid.SetRow(txtLastName, 1);

```

```

// Имя
var lblFirstName = new Label { Content = "Имя:", Style = labelStyle
};
Grid.SetRow(lblFirstName, 2);
var txtFirstName = new TextBox { Style = textBoxStyle };
txtFirstName.SetBinding(TextBox.TextProperty, new
Binding("FirstName") { Mode = BindingMode.TwoWay });
Grid.SetRow(txtFirstName, 3);

// Должность
var lblPosition = new Label { Content = "Должность:", Style =
labelStyle };
Grid.SetRow(lblPosition, 4);
var txtPosition = new TextBox { Style = textBoxStyle };
txtPosition.SetBinding(TextBox.TextProperty, new
Binding("Position") { Mode = BindingMode.TwoWay });
Grid.SetRow(txtPosition, 5);

// Отдел
var lblDepartment = new Label { Content = "Отдел:", Style =
labelStyle };
Grid.SetRow(lblDepartment, 6);
var txtDepartment = new TextBox { Style = textBoxStyle };
txtDepartment.SetBinding(TextBox.TextProperty, new
Binding("Department") { Mode = BindingMode.TwoWay });
Grid.SetRow(txtDepartment, 7);

// Email
var lblEmail = new Label { Content = "Email:", Style = labelStyle };
Grid.SetRow(lblEmail, 8);
var txtEmail = new TextBox { Style = textBoxStyle };
txtEmail.SetBinding(TextBox.TextProperty, new Binding("Email") {
Mode = BindingMode.TwoWay });
Grid.SetRow(txtEmail, 9);

// Телефон

```

```

var lblPhone = new Label { Content = "Телефон:", Style = labelStyle
};

Grid.SetRow(lblPhone, 10);
var txtPhone = new TextBox { Style = textBoxStyle };
txtPhone.SetBinding(TextBox.TextProperty, new Binding("Phone") {
Mode = BindingMode.TwoWay });
Grid.SetRow(txtPhone, 11);

// КНОПКИ
var buttonPanel = new StackPanel
{
    Orientation = Orientation.Horizontal,
    HorizontalAlignment = HorizontalAlignment.Right,
    Margin = new Thickness(0, 15, 0, 0)
};
Grid.SetRow(buttonPanel, 12);

var btnSave = new Button
{
    Content = "СОХРАНИТЬ",
    Width = 100,
    Margin = new Thickness(0, 0, 10, 0),
    Padding = new Thickness(5),
    IsDefault = true
};
btnSave.Click += (s, e) => {
    if (string.IsNullOrEmpty(FirstName) ||
string.IsNullOrEmpty(LastName))
    {
        MessageBox.Show("Фамилия и имя обязательны для
заполнения", "Ошибка", MessageBoxButton.OK,
MessageBoxImage.Warning);
        return;
    }
    DialogResult = true;
    Close();
}

```

```

};

var btnCancel = new Button
{
    Content = "Отмена",
    Width = 100,
    Padding = new Thickness(5),
    IsCancel = true
};
btnCancel.Click += (s, e) => { DialogResult = false; Close(); };

buttonPanel.Children.Add(btnSave);
buttonPanel.Children.Add(btnCancel);

// Добавление элементов на grid
grid.Children.Add(lblLastName);
grid.Children.Add(txtLastName);
grid.Children.Add(lblFirstName);
grid.Children.Add(txtFirstName);
grid.Children.Add(lblPosition);
grid.Children.Add(txtPosition);
grid.Children.Add(lblDepartment);
grid.Children.Add(txtDepartment);
grid.Children.Add(lblEmail);
grid.Children.Add(txtEmail);
grid.Children.Add(lblPhone);
grid.Children.Add(txtPhone);
grid.Children.Add(buttonPanel);

    Content = grid;
}
}
}
ProjectsWindow
<Window x:Class="MyWpfApp.ProjectsWindow"
    xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"

```

```

xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
xmlns:d="http://schemas.microsoft.com/expression/blend/2008"
xmlns:mc="http://schemas.openxmlformats.org/markup-
compatibility/2006"
mc:Ignorable="d"
Title="Управление проектами - Почта России"
Height="900" Width="1400"
WindowStartupLocation="CenterScreen"
ResizeMode="NoResize"
Icon="/logo.png"
FontFamily="Segoe UI">

```

```
<Window.Resources>
```

```
<!-- Цвета в стиле Почты России -->
```

```
<SolidColorBrush x:Key="RussianPostBlue" Color="#003366"/>
```

```
<SolidColorBrush x:Key="RussianPostRed" Color="#D52B1E"/>
```

```
<SolidColorBrush x:Key="LightBackground" Color="#F5F5F5"/>
```

```
<SolidColorBrush x:Key="HoverBlue" Color="#004080"/>
```

```
<SolidColorBrush x:Key="SelectedBlue" Color="#0050A0"/>
```

```
<SolidColorBrush x:Key="BorderColor" Color="#CCCCCC"/>
```

```
<SolidColorBrush x:Key="LightGray" Color="#EEEEEE"/>
```

```
<!-- Увеличенные базовые стили -->
```

```
<Style x:Key="BaseControlStyle" TargetType="Control">
```

```
<Setter Property="Margin" Value="0,0,0,15"/>
```

```
<Setter Property="Padding" Value="10"/>
```

```
<Setter Property="FontSize" Value="16"/>
```

```
<Setter Property="Background" Value="White"/>
```

```
<Setter Property="BorderBrush" Value="{StaticResource
BorderColor}"/>
```

```
<Setter Property="BorderThickness" Value="1"/>
```

```
<Setter Property="VerticalContentAlignment" Value="Center"/>
```

```
</Style>
```

```
<!-- Увеличенный TextBox -->
```

```
<Style x:Key="ModernTextBox" TargetType="TextBox"
```

```

BasedOn="{StaticResource BaseControlStyle}">
    <Setter Property="Height" Value="42"/>
    <Style.Triggers>
        <Trigger Property="IsMouseOver" Value="True">
            <Setter Property="BorderBrush" Value="{StaticResource
RussianPostBlue}"/>
        </Trigger>
        <Trigger Property="IsFocused" Value="True">
            <Setter Property="BorderBrush" Value="{StaticResource
RussianPostBlue}"/>
            <Setter Property="BorderThickness" Value="2"/>
        </Trigger>
    </Style.Triggers>
</Style>

```

```

<!-- Увеличенный ComboBox -->
<Style x:Key="ModernComboBox" TargetType="ComboBox"
BasedOn="{StaticResource BaseControlStyle}">
    <Setter Property="Height" Value="42"/>
    <Setter Property="MinWidth" Value="220"/>
    <Style.Triggers>
        <Trigger Property="IsMouseOver" Value="True">
            <Setter Property="BorderBrush" Value="{StaticResource
RussianPostBlue}"/>
        </Trigger>
        <Trigger Property="IsKeyboardFocusWithin" Value="True">
            <Setter Property="BorderBrush" Value="{StaticResource
RussianPostBlue}"/>
            <Setter Property="BorderThickness" Value="2"/>
        </Trigger>
    </Style.Triggers>
</Style>

```

```

<!-- Увеличенный DatePicker -->
<Style x:Key="ModernDatePicker" TargetType="DatePicker"
BasedOn="{StaticResource BaseControlStyle}">

```



```

        <Setter Property="Height" Value="45"/>
        <Setter Property="Width" Value="220"/>
    </Style>

    <!-- Маленькая кнопка поиска -->
    <Style x:Key="SmallSearchButton" TargetType="Button">
        <Setter      Property="Background"      Value="{StaticResource
RussianPostBlue}"/>
        <Setter Property="Foreground" Value="White"/>
        <Setter Property="BorderThickness" Value="0"/>
        <Setter Property="Width" Value="40"/>
        <Setter Property="Height" Value="40"/>
        <Setter Property="Padding" Value="0"/>
        <Setter Property="Margin" Value="5,0,0,0"/>
        <Setter Property="FontSize" Value="18"/>
        <Setter Property="Cursor" Value="Hand"/>
        <Setter Property="Template">
            <Setter.Value>
                <ControlTemplate TargetType="Button">
                    <Border Background="{TemplateBinding Background}"
                        CornerRadius="4">
                        <ContentPresenter HorizontalAlignment="Center"
                            VerticalAlignment="Center"/>
                    </Border>
                </ControlTemplate>
            </Setter.Value>
        </Setter>
        <Style.Triggers>
            <Trigger Property="IsMouseOver" Value="True">
                <Setter      Property="Background"      Value="{StaticResource
HoverBlue}"/>
            </Trigger>
            <Trigger Property="IsPressed" Value="True">
                <Setter      Property="Background"      Value="{StaticResource
SelectedBlue}"/>
            </Trigger>
        </Style.Triggers>
    </Style>

```

```

        </Style.Triggers>
    </Style>

    <!-- Стиль для обычных кнопок -->
    <Style x:Key="ModernButton" TargetType="Button">
        <Setter      Property="Background"      Value="{StaticResource
RussianPostBlue}"/>
        <Setter Property="Foreground" Value="White"/>
        <Setter Property="BorderThickness" Value="0"/>
        <Setter Property="Padding" Value="15,8"/>
        <Setter Property="Margin" Value="8"/>
        <Setter Property="FontSize" Value="16"/>
        <Setter Property="FontWeight" Value="SemiBold"/>
        <Setter Property="Cursor" Value="Hand"/>
        <Setter Property="MinWidth" Value="120"/>
        <Setter Property="Height" Value="45"/>
        <Setter Property="Template">
            <Setter.Value>
                <ControlTemplate TargetType="Button">
                    <Border Background="{TemplateBinding Background}"
                        CornerRadius="5">
                        <ContentPresenter HorizontalAlignment="Center"
                            VerticalAlignment="Center"/>
                    </Border>
                </ControlTemplate>
            </Setter.Value>
        </Setter>
    <Style.Triggers>
        <Trigger Property="IsMouseOver" Value="True">
            <Setter      Property="Background"      Value="{StaticResource
HoverBlue}"/>
        </Trigger>
        <Trigger Property="IsPressed" Value="True">
            <Setter      Property="Background"      Value="{StaticResource
SelectedBlue}"/>
        </Trigger>
    </Style.Triggers>

```

```
</Style.Triggers>
</Style>
```

```
<!-- СТИЛЬ ДЛЯ КНОПКИ ВЫХОДА -->
<Style x:Key="ExitButtonStyle" TargetType="Button"
BasedOn="{StaticResource ModernButton}">
    <Setter Property="Background" Value="{StaticResource
RussianPostRed}" />
    <Style.Triggers>
        <Trigger Property="IsMouseOver" Value="True">
            <Setter Property="Background" Value="#C03020" />
        </Trigger>
    </Style.Triggers>
</Style>
```

```
<!-- СТИЛЬ ДЛЯ ListView -->
<Style x:Key="ModernListView" TargetType="ListView">
    <Setter Property="Background" Value="White" />
    <Setter Property="BorderBrush" Value="#DDDDDD" />
    <Setter Property="BorderThickness" Value="1" />
    <Setter Property="Margin" Value="10" />
    <Setter Property="FontSize" Value="15" />
    <Setter Property="ScrollViewer.HorizontalScrollBarVisibility"
Value="Disabled" />
</Style>
```

```
<!-- СТИЛЬ ДЛЯ DataGrid -->
<Style x:Key="ModernDataGrid" TargetType="DataGrid">
    <Setter Property="Background" Value="White" />
    <Setter Property="BorderBrush" Value="#DDDDDD" />
    <Setter Property="BorderThickness" Value="1" />
    <Setter Property="RowBackground" Value="White" />
    <Setter Property="AlternatingRowBackground" Value="#F9F9F9" />
    <Setter Property="GridLinesVisibility" Value="None" />
    <Setter Property="HeadersVisibility" Value="Column" />
    <Setter Property="AutoGenerateColumns" Value="False" />
```

```

<Setter Property="CanUserAddRows" Value="False"/>
<Setter Property="CanUserDeleteRows" Value="False"/>
<Setter Property="IsReadOnly" Value="True"/>
<Setter Property="SelectionMode" Value="Single"/>
<Setter Property="SelectionUnit" Value="FullRow"/>
<Setter Property="Margin" Value="5"/>
<Setter Property="FontSize" Value="15"/>
<Setter Property="RowHeight" Value="40"/>
<Setter Property="ColumnHeaderStyle">
    <Setter.Value>
        <Style TargetType="DataGridColumnHeader">
            <Setter Property="Background" Value="{StaticResource
RussianPostBlue}"/>
            <Setter Property="Foreground" Value="White"/>
            <Setter Property="Padding" Value="10,5"/>
            <Setter Property="FontWeight" Value="SemiBold"/>
            <Setter Property="HorizontalContentAlignment"
Value="Left"/>
        </Style>
    </Setter.Value>
</Setter>
<Setter Property="CellStyle">
    <Setter.Value>
        <Style TargetType="DataGridCell">
            <Setter Property="Padding" Value="10,5"/>
            <Setter Property="BorderThickness" Value="0"/>
            <Setter Property="FocusVisualStyle" Value="{x:Null}"/>
        </Style>
    </Setter.Value>
</Setter>
</Style>

<!-- СТИЛЬ ДЛЯ TabControl -->
<Style x:Key="ModernTabControl" TargetType="TabControl">
    <Setter Property="Margin" Value="0,20,0,0"/>
    <Setter Property="Background" Value="Transparent"/>

```

```

    <Setter Property="BorderBrush" Value="#DDDDDD"/>
    <Setter Property="BorderThickness" Value="0"/>
    <Setter Property="FontSize" Value="16"/>
</Style>

```

<!-- Полностью переработанный стиль для TabItem с гарантированной видимостью текста -->

```

<Style x:Key="ModernTabItem" TargetType="TabItem">
    <Setter Property="Template">
        <Setter.Value>
            <ControlTemplate TargetType="TabItem">
                <Border Name="Border" Background="{TemplateBinding
Background}"
                    BorderBrush="{TemplateBinding BorderBrush}"
                    BorderThickness="{TemplateBinding
BorderThickness}"
                    CornerRadius="2,2,0,0">
                    <ContentPresenter x:Name="ContentSite"
                        VerticalAlignment="Center"
                        HorizontalAlignment="Center"
                        ContentSource="Header"
                        Margin="15,8"
                        RecognizesAccessKey="True"/>
                </Border>
                <ControlTemplate.Triggers>
                    <Trigger Property="IsSelected" Value="True">
                        <Setter TargetName="Border" Property="Background"
Value="{StaticResource RussianPostBlue}"/>
                        <Setter TargetName="ContentSite"
Property="TextElement.Foreground" Value="White"/>
                    </Trigger>
                    <Trigger Property="IsSelected" Value="False">
                        <Setter TargetName="Border" Property="Background"
Value="White"/>
                        <Setter TargetName="ContentSite"
Property="TextElement.Foreground" Value="Black"/>
                </ControlTemplate.Triggers>
            </ControlTemplate>
        </Setter.Value>
    </Setter>
</Style>

```

```

        </Trigger>
        <Trigger Property="IsMouseOver" Value="True">
            <Setter TargetName="Border" Property="Background"
Value="{StaticResource HoverBlue}"/>
            <Setter TargetName="ContentSite"
Property="TextElement.Foreground" Value="White"/>
        </Trigger>
    </ControlTemplate.Triggers>
</ControlTemplate>
</Setter.Value>
</Setter>
<Setter Property="Foreground" Value="Black"/>
<Setter Property="FontWeight" Value="SemiBold"/>
<Setter Property="Margin" Value="0,0,5,0"/>
<Setter Property="Padding" Value="15,8"/>
</Style>

<!-- СТИЛЬ ДЛЯ ТЕКСТА ЗАГОЛОВКОВ ВКЛАДОК -->
<Style x:Key="TabHeaderTextStyle" TargetType="TextBlock">
    <Setter Property="Foreground" Value="Black"/>
    <Setter Property="FontSize" Value="16"/>
    <Setter Property="FontWeight" Value="SemiBold"/>
</Style>

<!-- УВЕЛИЧЕННЫЕ ЗАГОЛОВКИ -->
<Style x:Key="HeaderTextStyle" TargetType="TextBlock">
    <Setter Property="FontSize" Value="20"/>
    <Setter Property="FontWeight" Value="Bold"/>
    <Setter Property="Margin" Value="0,0,0,20"/>
</Style>
</Window.Resources>

<Grid Background="{StaticResource LightBackground}">
    <!-- УВЕЛИЧЕННЫЙ ЗАГОЛОВОК ОКНА -->
    <Border Background="{StaticResource RussianPostBlue}"
Height="90" VerticalAlignment="Top">

```

```

        <StackPanel
            Orientation="Horizontal"
            HorizontalAlignment="Center" VerticalAlignment="Center">
            <Image Source="/logo.png" Width="60" Height="60"
                Margin="0,0,20,0"/>
            <TextBlock Text="УПРАВЛЕНИЕ ПРОЕКТАМИ"
                Foreground="White"
                FontSize="26"
                FontWeight="Bold"
                VerticalAlignment="Center"/>
        </StackPanel>
    </Border>

    <Grid Margin="20" VerticalAlignment="Stretch">
        <Grid.ColumnDefinitions>
            <ColumnDefinition Width="400"/>
            <ColumnDefinition Width="20"/>
            <ColumnDefinition Width="*/>
        </Grid.ColumnDefinitions>

        <!-- Левая панель - список проектов -->
        <Border Grid.Column="0" Background="White" CornerRadius="5"
            Padding="20">
            <DockPanel>
                <StackPanel DockPanel.Dock="Top" Orientation="Horizontal"
                    Margin="0,0,0,20">
                    <TextBox x:Name="txtSearch" Width="300"
                        Style="{StaticResource ModernTextBox}"
                        Tag="Поиск проектов..."/>
                    <Button x:Name="btnSearch" Content="🔍"
                        Style="{StaticResource SmallSearchButton}"
                        Click="btnSearch_Click"/>
                </StackPanel>

                <ListView x:Name="projectsListView" DockPanel.Dock="Top"
                    Style="{StaticResource ModernListView}"

```

```

SelectionChanged="projectsListView_SelectionChanged">
    <ListView.View>
        <GridView>
            <GridViewColumn                                Header="Название"
DisplayMemberBinding="{Binding Name}" Width="250"/>
            <GridViewColumn                                Header="Статус"
DisplayMemberBinding="{Binding Status}" Width="120"/>
        </GridView>
    </ListView.View>
</ListView>

    <StackPanel                                            DockPanel.Dock="Bottom"
Orientation="Horizontal"                                HorizontalAlignment="Center"
Margin="0,15,0,0">
        <Button    x:Name="btnAddProject"    Content="Добавить"
Style="{StaticResource ModernButton}" Click="btnAddProject_Click"/>
        <Button    x:Name="btnDeleteProject" Content="Удалить"
Style="{StaticResource ModernButton}"    Margin="10,0,0,0"
Click="btnDeleteProject_Click"/>
    </StackPanel>
</DockPanel>
</Border>

<!-- Правая панель - детали проекта -->
<ScrollViewer                                            Grid.Column="2"
VerticalScrollBarVisibility="Auto">
    <Border Background="White" CornerRadius="5" Padding="25">
        <StackPanel>
            <TextBlock    Text="Информация    о    проекте"
Style="{StaticResource HeaderTextStyle}"
                                Foreground="{StaticResource RussianPostBlue}"/>

            <Grid>
                <Grid.ColumnDefinitions>
                    <ColumnDefinition Width="Auto"/>
                    <ColumnDefinition Width="*/>

```



```

</Grid.ColumnDefinitions>
<Grid.RowDefinitions>
    <RowDefinition Height="Auto"/>
    <RowDefinition Height="Auto"/>
    <RowDefinition Height="Auto"/>
    <RowDefinition Height="Auto"/>
    <RowDefinition Height="Auto"/>
    <RowDefinition Height="Auto"/>
    <RowDefinition Height="Auto"/>
</Grid.RowDefinitions>

<!-- Основная информация -->
<TextBlock      Grid.Row="0"      Grid.Column="0"
Text="Название:"      Margin="5"      VerticalAlignment="Center"
FontSize="16"/>
<TextBox      Grid.Row="0"      Grid.Column="1"
x:Name="txtProjectName" Style="{StaticResource ModernTextBox}"/>

<TextBlock      Grid.Row="1"      Grid.Column="0"
Text="Описание:"      Margin="5"      VerticalAlignment="Center"
FontSize="16"/>
<TextBox      Grid.Row="1"      Grid.Column="1"
x:Name="txtDescription" Style="{StaticResource ModernTextBox}"
AcceptsReturn="True"      Height="100"
VerticalScrollBarVisibility="Auto"/>

<TextBlock      Grid.Row="2"      Grid.Column="0"
Text="Руководитель:"      Margin="5"      VerticalAlignment="Center"
FontSize="16"/>
<ComboBox      Grid.Row="2"      Grid.Column="1"
x:Name="cmbManager" Style="{StaticResource ModernComboBox}"
DisplayMemberPath="FullName"/>

<TextBlock Grid.Row="3" Grid.Column="0" Text="Дата
начала:" Margin="5" VerticalAlignment="Center" FontSize="16"/>

```

```
<DatePicker      Grid.Row="3"      Grid.Column="1"
x:Name="dpStartDate" Style="{StaticResource ModernDatePicker}"/>
```

```
<TextBlock Grid.Row="4" Grid.Column="0" Text="Дата
окончания:" Margin="5" VerticalAlignment="Center" FontSize="16"/>
```

```
<DatePicker      Grid.Row="4"      Grid.Column="1"
x:Name="dpEndDate" Style="{StaticResource ModernDatePicker}"/>
```

```
<TextBlock      Grid.Row="5"      Grid.Column="0"
Text="Статус:" Margin="5" VerticalAlignment="Center" FontSize="16"/>
```

```
<ComboBox      Grid.Row="5"      Grid.Column="1"
x:Name="cmbStatus" Style="{StaticResource ModernComboBox}"/>
```

```
<ComboBoxItem Content="Планируется"/>
```

```
<ComboBoxItem Content="В работе"/>
```

```
<ComboBoxItem Content="Приостановлен"/>
```

```
<ComboBoxItem Content="Завершен"/>
```

```
<ComboBoxItem Content="Отменен"/>
```

```
</ComboBox>
```

```
<TextBlock      Grid.Row="6"      Grid.Column="0"
Text="Бюджет:" Margin="5" VerticalAlignment="Center" FontSize="16"/>
```

```
<TextBox      Grid.Row="6"      Grid.Column="1"
x:Name="txtBudget" Style="{StaticResource ModernTextBox}"/>
```

```
<!-- Кнопки сохранения -->
```

```
<StackPanel      Grid.Row="7"      Grid.Column="1"
Orientation="Horizontal" HorizontalAlignment="Right" Margin="0,20,0,0">
```

```
<Button  x:Name="btnSave"  Content="Сохранить"
Style="{StaticResource ModernButton}" Click="btnSave_Click"/>
```

```
<Button  x:Name="btnCancel" Content="Отмена"
Style="{StaticResource ModernButton}" Margin="15,0,0,0"
Click="btnCancel_Click"/>
```

```
</StackPanel>
```

```
</Grid>
```

```
<!-- Вкладки для дополнительной информации с
```

исправленными надписями -->

```
<TabControl Style="{StaticResource ModernTabControl}">
  <TabItem Style="{StaticResource ModernTabItem}">
    <TabItem.Header>
      <TextBlock Text="Задачи" Style="{StaticResource
TabHeaderTextStyle}"/>
    </TabItem.Header>
    <DataGrid x:Name="tasksDataGrid"
Style="{StaticResource ModernDataGrid}">
      <DataGrid.Columns>
        <DataGridTextColumn Header="Название"
Binding="{Binding Title}" Width="*/>
        <DataGridTextColumn Header="Исполнитель"
Binding="{Binding Assignee.FullName}" Width="250"/>
        <DataGridTextColumn Header="Статус"
Binding="{Binding Status}" Width="150"/>
        <DataGridTextColumn Header="Срок"
Binding="{Binding Deadline, StringFormat={ } {0:dd.MM.yyyy} }"
Width="150"/>
      </DataGrid.Columns>
    </DataGrid>
  </TabItem>

  <TabItem Style="{StaticResource ModernTabItem}">
    <TabItem.Header>
      <TextBlock Text="Ресурсы" Style="{StaticResource
TabHeaderTextStyle}"/>
    </TabItem.Header>
    <DataGrid x:Name="resourcesDataGrid"
Style="{StaticResource ModernDataGrid}">
      <DataGrid.Columns>
        <DataGridTextColumn Header="Тип"
Binding="{Binding ResourceType}" Width="200"/>
        <DataGridTextColumn Header="Количество"
Binding="{Binding Amount}" Width="150"/>
        <DataGridTextColumn Header="Описание"
```

```

Binding="{Binding Description}" Width="*/>
    </DataGrid.Columns>
</DataGrid>
</TabItem>

<TabItem Style="{StaticResource ModernTabItem}">
    <TabItem.Header>
        <TextBlock                                Text="Участники"
Style="{StaticResource TabHeaderTextStyle}" />
    </TabItem.Header>
    <DataGrid                                x:Name="membersDataGrid"
Style="{StaticResource ModernDataGrid}">
        <DataGrid.Columns>
            <DataGridTextColumn            Header="Сотрудник"
Binding="{Binding Employee.FullName}" Width="*/>
            <DataGridTextColumn            Header="Роль"
Binding="{Binding Role}" Width="250"/>
        </DataGrid.Columns>
    </DataGrid>
</TabItem>
</TabControl>
</StackPanel>
</Border>
</ScrollViewer>
</Grid>
</Grid>
</Window>

```

ProjectsWindow.xaml.cs

```

using System;
using System.Collections.Generic;
using System.Data;
using System.Windows;
using System.Windows.Controls;
using MySql.Data.MySqlClient;

```

namespace MyWpfApp

```

{
    public partial class ProjectsWindow : Window
    {
        private string connectionString =
"server=localhost;user=root;database=pochta;port=3306;password=";
        private int currentProjectId = -1;

        public ProjectsWindow()
        {
            InitializeComponent();
            LoadProjects();
            LoadManagers();
        }

        private void LoadProjects(string searchTerm = "")
        {
            projectsListView.Items.Clear();

            using (var connection = new MySqlConnection(connectionString))
            {
                connection.Open();
                string query = "SELECT ProjectID, Name, Status FROM Projects";

                if (!string.IsNullOrEmpty(searchTerm))
                {
                    query += " WHERE Name LIKE @searchTerm";
                }

                var command = new MySqlCommand(query, connection);

                if (!string.IsNullOrEmpty(searchTerm))
                {
                    command.Parameters.AddWithValue("@searchTerm",
$"%" + searchTerm + "%");
                }
            }
        }
    }
}

```

```

        using (var reader = command.ExecuteReader())
        {
            while (reader.Read())
            {
                projectsListView.Items.Add(new
                {
                    ProjectID = reader.GetInt32("ProjectID"),
                    Name = reader.GetString("Name"),
                    Status = reader.GetString("Status")
                });
            }
        }
    }

    private void LoadManagers()
    {
        cmbManager.Items.Clear();

        using (var connection = new MySqlConnection(connectionString))
        {
            connection.Open();
            string query = "SELECT EmployeeID, CONCAT(FirstName, ' ',
LastName) AS FullName FROM Employees";
            var command = new MySqlCommand(query, connection);

            using (var reader = command.ExecuteReader())
            {
                while (reader.Read())
                {
                    cmbManager.Items.Add(new
                    {
                        EmployeeID = reader.GetInt32("EmployeeID"),
                        FullName = reader.GetString("FullName")
                    });
                }
            }
        }
    }

```

```

    }
}

private void LoadProjectDetails(int projectId)
{
    currentProjectId = projectId;

    using (var connection = new MySqlConnection(connectionString))
    {
        connection.Open();
        string query = "SELECT * FROM Projects WHERE ProjectID =
@projectId";
        var command = new MySqlCommand(query, connection);
        command.Parameters.AddWithValue("@projectId", projectId);

        using (var reader = command.ExecuteReader())
        {
            if (reader.Read())
            {
                txtProjectName.Text = reader["Name"].ToString();
                txtDescription.Text = reader["Description"].ToString();
                dpStartDate.SelectedDate = reader["StartDate"] !=
DBNull.Value ? (DateTime?)reader["StartDate"] : null;
                dpEndDate.SelectedDate = reader["EndDate"] !=
DBNull.Value ? (DateTime?)reader["EndDate"] : null;
                cmbStatus.Text = reader["Status"].ToString();
                txtBudget.Text = reader["Budget"].ToString();

                // Установка менеджера
                if (reader["ManagerID"] != DBNull.Value)
                {
                    int managerId = Convert.ToInt32(reader["ManagerID"]);
                    foreach (dynamic item in cmbManager.Items)
                    {
                        if (item.EmployeeID == managerId)

```

```

        {
            cmbManager.SelectedItem = item;
            break;
        }
    }
}
}
}
}
}
}

```

```

LoadProjectTasks(projectId);
LoadProjectResources(projectId);
LoadProjectMembers(projectId);
}

```

```

private void LoadProjectTasks(int projectId)
{
    tasksDataGrid.Items.Clear();

    using (var connection = new MySqlConnection(connectionString))
    {
        connection.Open();
        string query = @"SELECT t.*, CONCAT(e.FirstName, ' ',
e.LastName) AS AssigneeName
                        FROM Tasks t
                        LEFT JOIN Employees e ON t.AssigneeID =
e.EmployeeID
                        WHERE t.ProjectID = @projectId";
        var command = new MySqlCommand(query, connection);
        command.Parameters.AddWithValue("@projectId", projectId);

        using (var reader = command.ExecuteReader())
        {
            while (reader.Read())
            {
                tasksDataGrid.Items.Add(new

```



```

        {
            TaskID = reader.GetInt32("TaskID"),
            Title = reader.GetString("Title"),
            Status = reader.GetString("Status"),
            Deadline = reader["Deadline"] != DBNull.Value ?
reader.GetDateTime("Deadline") : (DateTime?)null,
            Assignee = new { FullName =
reader["AssigneeName"].ToString() }
        });
    }
}
}
}

```

```

private void LoadProjectResources(int projectId)
{
    resourcesDataGrid.Items.Clear();

    using (var connection = new MySqlConnection(connectionString))
    {
        connection.Open();
        string query = "SELECT * FROM ProjectResources WHERE
ProjectID = @projectId";
        var command = new MySqlCommand(query, connection);
        command.Parameters.AddWithValue("@projectId", projectId);

        using (var reader = command.ExecuteReader())
        {
            while (reader.Read())
            {
                resourcesDataGrid.Items.Add(new
                {
                    ResourceID = reader.GetInt32("ResourceID"),
                    ResourceType = reader.GetString("ResourceType"),
                    Amount = reader.GetDecimal("Amount"),
                    Description = reader["Description"].ToString()
                }
            }
        }
    }
}

```

```

        });
    }
}

private void LoadProjectMembers(int projectId)
{
    membersDataGrid.Items.Clear();

    using (var connection = new MySqlConnection(connectionString))
    {
        connection.Open();
        string query = @"SELECT ep.*, CONCAT(e.FirstName, ' ',
e.LastName) AS EmployeeName
                        FROM EmployeeProjects ep
                        JOIN Employees e ON ep.EmployeeID = e.EmployeeID
                        WHERE ep.ProjectID = @projectId";
        var command = new MySqlCommand(query, connection);
        command.Parameters.AddWithValue("@projectId", projectId);

        using (var reader = command.ExecuteReader())
        {
            while (reader.Read())
            {
                membersDataGrid.Items.Add(new
                {
                    EmployeeID = reader.GetInt32("EmployeeID"),
                    Employee = new { FullName =
reader.GetString("EmployeeName") },
                    Role = reader.GetString("Role")
                });
            }
        }
    }
}

```

```

        private void projectsListView_SelectionChanged(object sender,
System.Windows.Controls.SelectionChangedEventArgs e)
        {
            if (projectsListView.SelectedItem != null)
            {
                dynamic selectedItem = projectsListView.SelectedItem;
                LoadProjectDetails(selectedItem.ProjectID);
            }
        }

        private void btnSearch_Click(object sender, RoutedEventArgs e)
        {
            LoadProjects(txtSearch.Text);
        }

        private void btnAddProject_Click(object sender, RoutedEventArgs e)
        {
            currentProjectId = -1;
            ClearForm();
        }

        private void btnDeleteProject_Click(object sender, RoutedEventArgs e)
        {
            if (projectsListView.SelectedItem == null)
            {
                MessageBox.Show("Выберите проект для удаления", "Ошибка",
MessageBoxButton.OK, MessageBoxImage.Warning);
                return;
            }

            if (MessageBox.Show("Вы уверены, что хотите удалить этот
проект?", "Подтверждение удаления",
                MessageBoxButton.YesNo, MessageBoxImage.Question) ==
MessageBoxResult.Yes)
            {

```

```

dynamic selectedItem = projectsListView.SelectedItem;

using (var connection = new MySqlConnection(connectionString))
{
    connection.Open();
    var command = new MySqlCommand("DELETE FROM
Projects WHERE ProjectID = @projectId", connection);
    command.Parameters.AddWithValue("@projectId",
selectedItem.ProjectID);
    command.ExecuteNonQuery();
}

LoadProjects();
ClearForm();
}

private void btnSave_Click(object sender, RoutedEventArgs e)
{
    if (string.IsNullOrEmpty(txtProjectName.Text))
    {
        MessageBox.Show("Введите название проекта", "Ошибка",
MessageBoxButton.OK, MessageBoxImage.Warning);
        return;
    }

    try
    {
        using (var connection = new MySqlConnection(connectionString))
        {
            connection.Open();
            MySqlCommand command;

            if (currentProjectId == -1)
            {
                // Добавление нового проекта

```

```

        command = new MySqlCommand(
            @"INSERT INTO Projects
            (Name, Description, StartDate, EndDate, Status, Budget,
ManagerID)
            VALUES
            (@name, @desc, @startDate, @endDate, @status,
@budget, @managerId)", connection);
    }
    else
    {
        // Обновление существующего проекта
        command = new MySqlCommand(
            @"UPDATE Projects SET
            Name = @name,
            Description = @desc,
            StartDate = @startDate,
            EndDate = @endDate,
            Status = @status,
            Budget = @budget,
            ManagerID = @managerId
            WHERE ProjectID = @projectId", connection);
        command.Parameters.AddWithValue("@projectId",
currentProjectId);
    }

    command.Parameters.AddWithValue("@name",
txtProjectName.Text);
    command.Parameters.AddWithValue("@desc",
txtDescription.Text);
    command.Parameters.AddWithValue("@startDate",
dpStartDate.SelectedDate.HasValue ? dpStartDate.SelectedDate.Value :
(object)DBNull.Value);
    command.Parameters.AddWithValue("@endDate",
dpEndDate.SelectedDate.HasValue ? dpEndDate.SelectedDate.Value :
(object)DBNull.Value);
    command.Parameters.AddWithValue("@status",

```

```

cmbStatus.Text);
        command.Parameters.AddWithValue("@budget",
string.IsNullOrEmpty(txtBudget.Text) ? (object)DBNull.Value :
decimal.Parse(txtBudget.Text));
        command.Parameters.AddWithValue("@managerId",
cmbManager.SelectedItem != null ?
((dynamic)cmbManager.SelectedItem).EmployeeID :
(object)DBNull.Value);

        command.ExecuteNonQuery();

        if (currentProjectId == -1)
        {
            currentProjectId = (int)command.LastInsertedId;
        }

        MessageBox.Show("Проект сохранен", "Успех",
MessageBoxButton.OK, MessageBoxImage.Information);
        LoadProjects();
    }
}
catch (Exception ex)
{
    MessageBox.Show($"Ошибка при сохранении проекта:
{ex.Message}", "Ошибка", MessageBoxButton.OK,
MessageBoxImage.Error);
}
}

private void btnCancel_Click(object sender, RoutedEventArgs e)
{
    if (projectsListView.SelectedItem != null)
    {
        dynamic selectedItem = projectsListView.SelectedItem;
        LoadProjectDetails(selectedItem.ProjectID);
    }
}

```

```

else
{
    ClearForm();
}
}

```

```

private void ClearForm()
{
    txtProjectName.Text = "";
    txtDescription.Text = "";
    dpStartDate.SelectedDate = null;
    dpEndDate.SelectedDate = null;
    cmbStatus.SelectedIndex = -1;
    txtBudget.Text = "";
    cmbManager.SelectedIndex = -1;
    tasksDataGrid.Items.Clear();
    resourcesDataGrid.Items.Clear();
    membersDataGrid.Items.Clear();
}
}
}

```

ReportsWindow.xaml

```

<Window x:Class="MyWpfApp.ReportsWindow"
    xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
    xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
    xmlns:d="http://schemas.microsoft.com/expression/blend/2008"
    xmlns:mc="http://schemas.openxmlformats.org/markup-
compatibility/2006"
    mc:Ignorable="d"
    Title="Отчеты по проектам - Почта России"
    Height="750" Width="1300"
    WindowStartupLocation="CenterScreen"
    ResizeMode="CanResizeWithGrip"
    Icon="/logo.png"
    FontFamily="Segoe UI">

```

```

<Window.Resources>
  <!-- Цвета Почты России -->
  <SolidColorBrush x:Key="RussianPostBlue" Color="#003366"/>
  <SolidColorBrush x:Key="RussianPostRed" Color="#D52B1E"/>
  <SolidColorBrush x:Key="LightBackground" Color="#F5F5F5"/>
  <SolidColorBrush x:Key="HoverBlue" Color="#004080"/>
  <SolidColorBrush x:Key="SelectedBlue" Color="#0050A0"/>
  <SolidColorBrush x:Key="BorderColor" Color="#CCCCCC"/>
  <SolidColorBrush x:Key="LightGray" Color="#EEEEEE"/>
  <SolidColorBrush x:Key="ExitButtonRed" Color="#D9534F"/>
  <SolidColorBrush x:Key="ExitButtonHover" Color="#C9302C"/>

  <!-- СТИЛЬ ДЛЯ КНОПОК -->
  <Style x:Key="ModernButton" TargetType="Button">
    <Setter      Property="Background"      Value="{StaticResource
RussianPostBlue}"/>
    <Setter Property="Foreground" Value="White"/>
    <Setter Property="BorderThickness" Value="0"/>
    <Setter Property="Padding" Value="12,6"/>
    <Setter Property="Margin" Value="8,0"/>
    <Setter Property="FontSize" Value="15"/>
    <Setter Property="FontWeight" Value="SemiBold"/>
    <Setter Property="Cursor" Value="Hand"/>
    <Setter Property="MinWidth" Value="100"/>
    <Setter Property="Height" Value="40"/>
    <Setter Property="Template">
      <Setter.Value>
        <ControlTemplate TargetType="Button">
          <Border Background="{TemplateBinding Background}"
CornerRadius="4">
            <ContentPresenter      HorizontalAlignment="Center"
VerticalAlignment="Center"/>
          </Border>
        </ControlTemplate>
      </Setter.Value>
    </Setter>
  </Style>

```



```

<Style.Triggers>
    <Trigger Property="IsMouseOver" Value="True">
        <Setter Property="Background" Value="{StaticResource
HoverBlue}"/>
    </Trigger>
    <Trigger Property="IsPressed" Value="True">
        <Setter Property="Background" Value="{StaticResource
SelectedBlue}"/>
    </Trigger>
</Style.Triggers>
</Style>

```

```

<!-- СТИЛЬ ДЛЯ КНОПКИ ЗАКРЫТИЯ -->
<Style x:Key="ExitButtonStyle" TargetType="Button"
BasedOn="{StaticResource ModernButton}">
    <Setter Property="Background" Value="{StaticResource
ExitButtonRed}"/>
    <Setter Property="MinWidth" Value="90"/>
    <Style.Triggers>
        <Trigger Property="IsMouseOver" Value="True">
            <Setter Property="Background" Value="{StaticResource
ExitButtonHover}"/>
        </Trigger>
    </Style.Triggers>
</Style>

```

```

<!-- СТИЛЬ ДЛЯ DataGrid -->
<Style x:Key="ModernDataGrid" TargetType="DataGrid">
    <Setter Property="Background" Value="White"/>
    <Setter Property="BorderBrush" Value="#DDDDDD"/>
    <Setter Property="BorderThickness" Value="1"/>
    <Setter Property="RowBackground" Value="White"/>
    <Setter Property="AlternatingRowBackground" Value="#F9F9F9"/>
    <Setter Property="GridLinesVisibility" Value="All"/>
    <Setter Property="HeadersVisibility" Value="Column"/>
    <Setter Property="AutoGenerateColumns" Value="False"/>

```

```

<Setter Property="CanUserAddRows" Value="False"/>
<Setter Property="IsReadOnly" Value="True"/>
<Setter Property="SelectionMode" Value="Single"/>
<Setter Property="SelectionUnit" Value="FullRow"/>
<Setter Property="Margin" Value="0"/>
<Setter Property="FontSize" Value="14"/>
<Setter Property="RowHeight" Value="36"/>
<Setter Property="HorizontalGridLinesBrush" Value="#E0E0E0"/>
<Setter Property="VerticalGridLinesBrush" Value="#E0E0E0"/>
<Setter Property="ColumnHeaderStyle">
  <Setter.Value>
    <Style TargetType="DataGridColumnHeader">
      <Setter Property="Background" Value="{StaticResource
RussianPostBlue}"/>
      <Setter Property="Foreground" Value="White"/>
      <Setter Property="Padding" Value="10,8"/>
      <Setter Property="FontWeight" Value="SemiBold"/>
      <Setter Property="HorizontalContentAlignment"
Value="Left"/>
      <Setter Property="Height" Value="40"/>
    </Style>
  </Setter.Value>
</Setter>
<Setter Property="CellStyle">
  <Setter.Value>
    <Style TargetType="DataGridCell">
      <Setter Property="Padding" Value="10,8"/>
      <Setter Property="BorderThickness" Value="0"/>
      <Setter Property="FocusVisualStyle" Value="{x:Null}"/>
      <Setter Property="Template">
        <Setter.Value>
          <ControlTemplate TargetType="DataGridCell">
            <Grid Background="{TemplateBinding
Background}">
              <ContentPresenter VerticalAlignment="Center"/>
            </Grid>
          </Setter.Value>
        </ControlTemplate>
      </Setter>
    </Style>
  </Setter.Value>
</Setter>

```

```

        </ControlTemplate>
        </Setter.Value>
    </Setter>
</Style>
</Setter.Value>
</Setter>
</Style>

```

```

<!-- СТИЛЬ ДЛЯ ComboBox -->
<Style x:Key="ModernComboBox" TargetType="ComboBox">
    <Setter Property="Height" Value="38"/>
    <Setter Property="MinWidth" Value="200"/>
    <Setter Property="FontSize" Value="15"/>
    <Setter Property="Margin" Value="0,5,10,5"/>
    <Setter Property="Padding" Value="8"/>
    <Setter      Property="BorderBrush"      Value="{StaticResource
BorderColor}"/>
    <Setter Property="Background" Value="White"/>
    <Style.Triggers>
        <Trigger Property="IsMouseOver" Value="True">
            <Setter      Property="BorderBrush"      Value="{StaticResource
RussianPostBlue}"/>
        </Trigger>
        <Trigger Property="IsKeyboardFocusWithin" Value="True">
            <Setter      Property="BorderBrush"      Value="{StaticResource
RussianPostBlue}"/>
            <Setter Property="BorderThickness" Value="1.5"/>
        </Trigger>
    </Style.Triggers>
</Style>

```

```

<!-- СТИЛЬ ДЛЯ DatePicker -->
<Style x:Key="ModernDatePicker" TargetType="DatePicker">
    <Setter Property="Height" Value="38"/>
    <Setter Property="Width" Value="180"/>
    <Setter Property="FontSize" Value="15"/>

```

```

        <Setter Property="Margin" Value="0,5,10,5"/>
        <Setter      Property="BorderBrush"      Value="{StaticResource
BorderColor}"/>
        <Setter Property="Background" Value="White"/>
    </Style>

```

```

<!-- СТИЛЬ для GroupBox -->
<Style x:Key="FilterGroupBoxStyle" TargetType="GroupBox">
    <Setter Property="BorderBrush" Value="#DDDDDD"/>
    <Setter Property="BorderThickness" Value="1"/>
    <Setter Property="Margin" Value="0,0,0,15"/>
    <Setter Property="Padding" Value="10"/>
    <Setter Property="Background" Value="White"/>
    <Setter Property="HeaderTemplate">
        <Setter.Value>
            <DataTemplate>
                <TextBlock Text="{Binding}"
                    Foreground="{StaticResource RussianPostBlue}"
                    FontWeight="Bold"
                    FontSize="16"
                    Padding="0,0,0,5"/>
            </DataTemplate>
        </Setter.Value>
    </Setter>
</Style>
</Window.Resources>

```

```

<Grid Background="{StaticResource LightBackground}">
    <Grid.RowDefinitions>
        <RowDefinition Height="Auto"/>
        <RowDefinition Height="*/>
        <RowDefinition Height="Auto"/>
    </Grid.RowDefinitions>

```

```

<!-- Шапка с логотипом -->
<Border      Grid.Row="0"      Background="{StaticResource

```

```

RussianPostBlue}" Padding="15">
    <StackPanel
                                                Orientation="Horizontal"
HorizontalAlignment="Center">
        <Image      Source="/logo.png"      Width="50"      Height="50"
Margin="0,0,15,0"/>
        <TextBlock Text="ОТЧЕТЫ ПО ПРОЕКТАМ"
                    Foreground="White"
                    FontSize="24"
                    FontWeight="Bold"
                    VerticalAlignment="Center"/>
    </StackPanel>
</Border>

<!-- Основное содержимое -->
<Grid Grid.Row="1" Margin="20">
    <Grid.RowDefinitions>
        <RowDefinition Height="Auto"/>
        <RowDefinition Height="*/>
    </Grid.RowDefinitions>

    <!-- Панель фильтров -->
    <GroupBox      Grid.Row="0"      Header="ФИЛЬТРЫ"
Style="{StaticResource FilterGroupBoxStyle}">
        <StackPanel Orientation="Horizontal">
            <StackPanel Margin="0,0,15,0">
                <TextBlock Text="Тип отчета" FontWeight="SemiBold"
Margin="0,0,0,3"/>
                <ComboBox      x:Name="cmbReportType"
Style="{StaticResource ModernComboBox}" SelectedIndex="0">
                    <ComboBoxItem Content="По задачам"/>
                    <ComboBoxItem Content="По ресурсам"/>
                    <ComboBoxItem Content="По сотрудникам"/>
                    <ComboBoxItem Content="По бюджету"/>
                    <ComboBoxItem Content="По срокам"/>
                </ComboBox>
            </StackPanel>

```

```

        <StackPanel Margin="0,0,15,0">
            <TextBlock      Text="Проект"      FontWeight="SemiBold"
Margin="0,0,0,3"/>
            <ComboBox x:Name="cmbProjects" Style="{StaticResource
ModernComboBox}" DisplayMemberPath="Name"/>
        </StackPanel>

```

```

        <StackPanel Margin="0,0,15,0">
            <TextBlock      Text="Период  с"      FontWeight="SemiBold"
Margin="0,0,0,3"/>
            <DatePicker x:Name="dpStartDate" Style="{StaticResource
ModernDatePicker}"/>
        </StackPanel>

```

```

        <StackPanel Margin="0,0,20,0">
            <TextBlock      Text="по"      FontWeight="SemiBold"
Margin="0,0,0,3"/>
            <DatePicker x:Name="dpEndDate" Style="{StaticResource
ModernDatePicker}"/>
        </StackPanel>

```

```

        <Button Content="Сформировать"
            Style="{StaticResource ModernButton}"
            VerticalAlignment="Bottom"
            MinWidth="140"
            Click="GenerateReport_Click"/>
        </StackPanel>
    </GroupBox>

```

```

    <!-- Таблица с отчетом -->
    <DataGrid      Grid.Row="1"      x:Name="dgReport"
Style="{StaticResource ModernDataGrid}">
        <DataGrid.Columns>
            <DataGridTextColumn Header="ID" Binding="{Binding ID}"
Width="60"/>

```

```

        <DataGridTextColumn Header="Название" Binding="{Binding
Name}" Width="*/>
        <DataGridTextColumn Header="Статус" Binding="{Binding
Status}" Width="120"/>
        <DataGridTextColumn Header="Дата" Binding="{Binding
Date, StringFormat={ } {0:dd.MM.yyyy} }" Width="110"/>
        <DataGridTextColumn Header="ОТВЕТСТВЕННЫЙ"
Binding="{Binding Responsible}" Width="180"/>
        <DataGridTextColumn Header="Кол-во" Binding="{Binding
Amount}" Width="90"/>
    </DataGrid.Columns>
</DataGrid>
</Grid>

<!-- Панель действий -->
<Border Grid.Row="2" Background="White" Padding="12"
BorderBrush="#DDDDDD" BorderThickness="0,1,0,0">
    <StackPanel Orientation="Horizontal"
HorizontalAlignment="Right">
        <Button Content="Экспорт в Excel"
Style="{StaticResource ModernButton}"
MinWidth="130"
Click="ExportToExcel_Click"/>

        <Button Content="Печать"
Style="{StaticResource ModernButton}"
MinWidth="90"
Margin="8,0,0,0"
Click="PrintReport_Click"/>

        <Button Content="Закреть"
Style="{StaticResource ExitButtonStyle}"
Margin="8,0,0,0"
Click="CloseWindow_Click"/>
    </StackPanel>
</Border>

```

```

        </Grid>
    </Window>
}

ReportsWindow.xaml.cs
using System;
using System.Data;
using System.Windows;
using System.Windows.Controls;
using MySql.Data.MySqlClient;
using System.Windows.Documents;
using System.Printing;
using System.Runtime.InteropServices;
using Excel = Microsoft.Office.Interop.Excel;

namespace MyWpfApp
{
    public partial class ReportsWindow : Window
    {
        private string connectionString =
"server=localhost;user=root;database=pochta;port=3306;password=";

        public ReportsWindow()
        {
            InitializeComponent();
            LoadProjects();
            SetDefaultDates();
            InitializeReportTypes();
        }

        private void InitializeReportTypes()
        {
            cmbReportType.Items.Clear();
            cmbReportType.Items.Add(new ComboBoxItem { Content = "По
задачам", Tag = "Tasks" });
            cmbReportType.Items.Add(new ComboBoxItem { Content = "По
ресурсам", Tag = "Resources" });
            cmbReportType.Items.Add(new ComboBoxItem { Content = "По

```



```

сотрудникам", Tag = "Employees" });
        cmbReportType.Items.Add(new ComboBoxItem { Content = "По
бюджету", Tag = "Budget" });
        cmbReportType.Items.Add(new ComboBoxItem { Content = "По
срокам", Tag = "Deadlines" });
        cmbReportType.SelectedIndex = 0;
    }

    private void LoadProjects()
    {
        try
        {
            using (var connection = new MySqlConnection(connectionString))
            {
                connection.Open();
                var command = new MySqlCommand("SELECT ProjectID,
Name FROM Projects ORDER BY Name", connection);
                var reader = command.ExecuteReader();

                var projects = new System.Collections.Generic.List<Project>();
                while (reader.Read())
                {
                    projects.Add(new Project
                    {
                        Id = reader.GetInt32("ProjectID"),
                        Name = reader.GetString("Name")
                    });
                }

                cmbProjects.ItemsSource = projects;
                if (projects.Count > 0)
                    cmbProjects.SelectedIndex = 0;
            }
        }
        catch (Exception ex)
        {

```

```

        MessageBox.Show($"Ошибка при загрузке проектов:
{ex.Message}", "Ошибка", MessageBoxButton.OK,
MessageBoxImage.Error);
    }
}

```

```

private void SetDefaultDates()
{
    dpStartDate.SelectedDate = DateTime.Now.AddMonths(-1);
    dpEndDate.SelectedDate = DateTime.Now;
}

```

```

private void GenerateReport_Click(object sender, RoutedEventArgs e)
{
    if (cmbReportType.SelectedItem == null)
    {
        MessageBox.Show("Выберите тип отчета", "Внимание",
MessageBoxButton.OK, MessageBoxImage.Warning);
        return;
    }
}

```

```

var selectedItem = (ComboBoxItem)cmbReportType.SelectedItem;
var reportType = selectedItem.Tag.ToString();
var project = cmbProjects.SelectedItem as Project;
var startDate = dpStartDate.SelectedDate ?? DateTime.MinValue;
var endDate = dpEndDate.SelectedDate ?? DateTime.MaxValue;

try
{
    using (var connection = new MySqlConnection(connectionString))
    {
        connection.Open();
        MySqlCommand command;
        System.Data.DataTable dataTable = new
System.Data.DataTable();

```

```

switch (reportType)
{
    case "Tasks":
        command = new MySqlCommand(@"
            SELECT
                t.TaskID AS 'ID задачи',
                t.Title AS 'Название',
                t.Status AS 'Статус',
                t.Priority AS 'Приоритет',
                t.Deadline AS 'Срок выполнения',
                CONCAT(e.FirstName, ' ', e.LastName) AS
'Исполнитель',
                (SELECT COALESCE(SUM(HoursWorked), 0)
FROM TimeLogs WHERE TaskID = t.TaskID) AS 'Затрачено часов'
            FROM Tasks t
            LEFT JOIN Employees e ON t.AssigneeID =
e.EmployeeID
            WHERE (@projectId = 0 OR t.ProjectID = @projectId)
            AND t.Deadline BETWEEN @startDate AND
@endDate
            ORDER BY t.Deadline", connection);
        break;

    case "Resources":
        command = new MySqlCommand(@"
            SELECT
                pr.ResourceID AS 'ID ресурса',
                pr.ResourceType AS 'Тип ресурса',
                pr.Amount AS 'Количество',
                pr.Description AS 'Описание',
                (SELECT Name FROM Projects WHERE ProjectID =
pr.ProjectID) AS 'Проект'
            FROM ProjectResources pr
            WHERE (@projectId = 0 OR pr.ProjectID =
@projectId)", connection);
        break;

```

```

case "Employees":
    command = new MySqlCommand(@"
        SELECT
            e.EmployeeID AS 'ID сотрудника',
            CONCAT(e.FirstName, ' ', e.LastName) AS 'ФИО',
            e.Position AS 'Должность',
            e.Department AS 'Отдел',
            ep.Role AS 'Роль в проекте',
            (SELECT COALESCE(SUM(HoursWorked), 0)
             FROM TimeLogs
             WHERE EmployeeID = e.EmployeeID
             AND LogDate BETWEEN @startDate AND
@endDate) AS 'Отработано часов'
        FROM Employees e
        JOIN EmployeeProjects ep ON e.EmployeeID =
ep.EmployeeID
        WHERE (@projectId = 0 OR ep.ProjectID = @projectId)
        ORDER BY e.LastName, e.FirstName", connection);
    break;

case "Budget":
    command = new MySqlCommand(@"
        SELECT
            'Бюджет проекта' AS 'Статья',
            p.Budget AS 'Планный бюджет',
            (SELECT COALESCE(SUM(Amount), 0)
             FROM ProjectResources
             WHERE ProjectID = p.ProjectID) AS 'Фактические
расходы',
            p.Budget - (SELECT COALESCE(SUM(Amount), 0)
                        FROM ProjectResources
                        WHERE ProjectID = p.ProjectID) AS
'Остаток',
            CONCAT(e.FirstName, ' ', e.LastName) AS
'Ответственный'
    
```

```

        FROM Projects p
        JOIN Employees e ON p.ManagerID = e.EmployeeID
        WHERE (@projectId = 0 OR p.ProjectID = @projectId)",
connection);

        break;

    case "Deadlines":
        command = new MySqlCommand(@"
        SELECT
            p.ProjectID AS 'ID проекта',
            p.Name AS 'Название проекта',
            COUNT(t.TaskID) AS 'Количество задач',
            SUM(CASE WHEN t.Status = 'Завершена' THEN 1
ELSE 0 END) AS 'Завершено',
            SUM(CASE WHEN t.Deadline < CURDATE() AND
t.Status != 'Завершена' THEN 1 ELSE 0 END) AS 'Просрочено'
        FROM Projects p
        LEFT JOIN Tasks t ON p.ProjectID = t.ProjectID
        WHERE (@projectId = 0 OR p.ProjectID = @projectId)
        AND t.Deadline BETWEEN @startDate AND
@endDate

        GROUP BY p.ProjectID, p.Name", connection);
        break;

    default:
        MessageBox.Show("Неизвестный тип отчета",
"Ошибка", MessageBoxButtons.OK, MessageBoxIcon.Error);
        return;
    }

    command.Parameters.AddWithValue("@projectId", project?.Id
?? 0);

    command.Parameters.AddWithValue("@startDate", startDate);
    command.Parameters.AddWithValue("@endDate", endDate);

    MySqlDataAdapter adapter = new

```

```

MySqlDataAdapter(command);
    adapter.Fill(dataTable);

    dgReport.Columns.Clear();
    dgReport.AutoGenerateColumns = true;
    dgReport.ItemsSource = dataTable.DefaultView;

    if (dataTable.Rows.Count == 0)
    {
        MessageBox.Show("Нет данных для отображения по
выбранным критериям", "Информация", MessageBoxButton.OK,
MessageBoxImage.Information);
    }
}
catch (Exception ex)
{
    MessageBox.Show($"Ошибка при формировании отчета:
{ex.Message}", "Ошибка", MessageBoxButton.OK,
MessageBoxImage.Error);
}
}

private void ExportToExcel_Click(object sender, RoutedEventArgs e)
{
    if (dgReport.Items.Count == 0)
    {
        MessageBox.Show("Нет данных для экспорта", "Внимание",
MessageBoxButton.OK, MessageBoxImage.Warning);
        return;
    }

    Excel.Application excelApp = null;
    Excel.Workbook workbook = null;
    Excel.Worksheet worksheet = null;

```

```

try
{
    excelApp = new Excel.Application();
    excelApp.Visible = true;
    workbook = excelApp.Workbooks.Add();
    worksheet = (Excel.Worksheet)workbook.Sheets[1];

    DataView dataView = (DataView)dgReport.ItemsSource;
    System.Data.DataTable dataTable = dataView.ToTable();

    // Заголовки
    for (int i = 0; i < dataTable.Columns.Count; i++)
    {
        worksheet.Cells[1, i + 1] = dataTable.Columns[i].ColumnName;
    }

    // Данные
    for (int i = 0; i < dataTable.Rows.Count; i++)
    {
        for (int j = 0; j < dataTable.Columns.Count; j++)
        {
            worksheet.Cells[i + 2, j + 1] =
dataTable.Rows[i][j].ToString();
        }
    }

    // Форматирование
    Excel.Range headerRange = worksheet.Range[worksheet.Cells[1,
1], worksheet.Cells[1, dataTable.Columns.Count]];
    headerRange.Font.Bold = true;
    headerRange.Interior.Color = Excel.XlRgbColor.rgbLightGray;

    worksheet.Columns.AutoFit();

    MessageBox.Show("Экспорт в Excel выполнен успешно",
"Успех", MessageBoxButtons.OK, MessageBoxIcon.Information);

```

```

    }
    catch (Exception ex)
    {
        MessageBox.Show($"Ошибка при экспорте в Excel:
{ex.Message}", "Ошибка", MessageBoxButton.OK,
MessageBoxImage.Error);
    }
    finally
    {
        // Освобождение ресурсов
        if (worksheet != null) Marshal.ReleaseComObject(worksheet);
        if (workbook != null) Marshal.ReleaseComObject(workbook);
        if (excelApp != null) Marshal.ReleaseComObject(excelApp);

        GC.Collect();
        GC.WaitForPendingFinalizers();
    }
}

```

```

private void PrintReport_Click(object sender, RoutedEventArgs e)
{
    if (dgReport.Items.Count == 0)
    {
        MessageBox.Show("Нет данных для печати", "Внимание",
MessageBoxButton.OK, MessageBoxImage.Warning);
        return;
    }
}

```

```

try
{
    PrintDialog printDialog = new PrintDialog();
    if (printDialog.ShowDialog() == true)
    {
        FlowDocument document = new FlowDocument();
        document.PageWidth = printDialog.PrintableAreaWidth;
        document.PageHeight = printDialog.PrintableAreaHeight;
    }
}

```



```

document.PagePadding = new Thickness(50);
document.ColumnGap = 0;
document.ColumnWidth = printDialog.PrintableAreaWidth;

// Заголовок
Paragraph header = new Paragraph(new Run($"Отчет:
{{{ComboBoxItem)cmbReportType.SelectedItem}.Content}"));
header.FontSize = 14;
header.FontWeight = FontWeights.Bold;
header.TextAlignment = TextAlignment.Center;
document.Blocks.Add(header);

// Подзаголовок
if (cmbProjects.SelectedItem != null)
{
    Paragraph subheader = new Paragraph(new Run($"Проект:
{{{Project)cmbProjects.SelectedItem}.Name}"));
    subheader.FontSize = 12;
    subheader.TextAlignment = TextAlignment.Center;
    document.Blocks.Add(subheader);
}

// Период
Paragraph period = new Paragraph(new Run($"Период:
{dpStartDate.SelectedDate?.ToShortDateString()} -
{dpEndDate.SelectedDate?.ToShortDateString()}"));
period.FontSize = 12;
period.TextAlignment = TextAlignment.Center;
document.Blocks.Add(period);

// Дата формирования
Paragraph dateParagraph = new Paragraph(new Run($"Дата
формирования: {DateTime.Now.ToShortDateString()}"));
dateParagraph.FontSize = 12;
dateParagraph.TextAlignment = TextAlignment.Center;
document.Blocks.Add(dateParagraph);

```

```

// Таблица с данными
Table table = new Table();
document.Blocks.Add(table);

DataView dataView = (DataView)dgReport.ItemsSource;
System.Data.DataTable dataTable = dataView.ToTable();

// Стили таблицы
table.CellSpacing = 0;
table.BorderThickness = new Thickness(1);
table.BorderBrush = System.Windows.Media.Brushes.Black;

// Колонки
for (int i = 0; i < dataTable.Columns.Count; i++)
{
    table.Columns.Add(new TableColumn());
}

// Заголовки
TableRowGroup headerGroup = new TableRowGroup();
table.RowGroups.Add(headerGroup);
TableRow headerRow = new TableRow();
headerGroup.Rows.Add(headerRow);

foreach (DataColumn column in dataTable.Columns)
{
    headerRow.Cells.Add(new TableCell(new Paragraph(new
Run(column.ColumnName)))
    {
        Background = System.Windows.Media.Brushes.LightGray,
        FontWeight = FontWeights.Bold,
        BorderThickness = new Thickness(1),
        BorderBrush = System.Windows.Media.Brushes.Black,
        Padding = new Thickness(4)
    });
}

```

```

    }

    // Данные
    TableRowGroup dataGroup = new TableRowGroup();
    table.RowGroups.Add(dataGroup);

    foreach (DataRow row in dataTable.Rows)
    {
        TableRow dataRow = new TableRow();
        dataGroup.Rows.Add(dataRow);

        foreach (object item in row.ItemArray)
        {
            dataRow.Cells.Add(new TableCell(new Paragraph(new
Run(item.ToString()))
            {
                BorderThickness = new Thickness(1),
                BorderBrush = System.Windows.Media.Brushes.Black,
                Padding = new Thickness(4)
            }));
        }
    }

printDialog.PrintDocument(((IDocumentPaginatorSource)document).Docum
entPaginator,
    $"Отчет:
{((ComboBoxItem)cmbReportType.SelectedItem).Content}");
    }
}
catch (Exception ex)
{
    MessageBox.Show($"Ошибка при печати: {ex.Message}",
"Ошибка", MessageBoxButton.OK, MessageBoxImage.Error);
}
}

```

```

private void CloseWindow_Click(object sender, RoutedEventArgs e)
{
    this.Close();
}
}

```

```

public class Project
{
    public int Id { get; set; }
    public string Name { get; set; }
}

```

TasksWindow.xaml

```

<Window x:Class="MyWpfApp.TasksWindow"
    xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
    xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
    xmlns:d="http://schemas.microsoft.com/expression/blend/2008"
    xmlns:mc="http://schemas.openxmlformats.org/markup-
compatibility/2006"
    xmlns:local="clr-namespace:MyWpfApp"
    mc:Ignorable="d"
    Title="Почта России - Управление задачами"
    Width="900"
    Height="700"
    WindowStyle="None"
    AllowsTransparency="True"
    Background="Transparent"
    WindowStartupLocation="CenterScreen"
    ResizeMode="NoResize"
    Icon="/logo.png">

```

```

<Window.Resources>
    <DropShadowEffect x:Key="ShadowEffect"
        Color="#40000000"
        BlurRadius="10"

```

ShadowDepth="5"
Opacity="0.5"/>

```
<Style x:Key="HeaderTextStyle" TargetType="TextBlock">  
  <Setter Property="FontSize" Value="20"/>  
  <Setter Property="FontWeight" Value="Bold"/>  
  <Setter Property="Foreground" Value="#003366"/>  
  <Setter Property="Margin" Value="0,0,0,10"/>  
</Style>
```

```
<Style x:Key="LabelStyle" TargetType="Label">  
  <Setter Property="Foreground" Value="#003366"/>  
  <Setter Property="FontWeight" Value="SemiBold"/>  
  <Setter Property="Margin" Value="5"/>  
  <Setter Property="VerticalAlignment" Value="Center"/>  
</Style>
```

```
<Style x:Key="TextBoxStyle" TargetType="TextBox">  
  <Setter Property="Margin" Value="5"/>  
  <Setter Property="Padding" Value="5"/>  
  <Setter Property="BorderBrush" Value="#003366"/>  
  <Setter Property="BorderThickness" Value="1"/>  
  <Setter Property="FontSize" Value="14"/>  
</Style>
```

```
<Style x:Key="ComboBoxStyle" TargetType="ComboBox">  
  <Setter Property="Margin" Value="5"/>  
  <Setter Property="Padding" Value="5"/>  
  <Setter Property="BorderBrush" Value="#003366"/>  
  <Setter Property="BorderThickness" Value="1"/>  
  <Setter Property="FontSize" Value="14"/>  
  <Setter Property="Background" Value="White"/>  
</Style>
```

```
<Style x:Key="DatePickerStyle" TargetType="DatePicker">  
  <Setter Property="Margin" Value="5"/>
```

```

<Setter Property="Padding" Value="5"/>
<Setter Property="BorderBrush" Value="#003366"/>
<Setter Property="BorderThickness" Value="1"/>
<Setter Property="FontSize" Value="14"/>
<Setter Property="Background" Value="White"/>
</Style>

```

```

<Style x:Key="DataGridStyle" TargetType="DataGrid">
  <Setter Property="Margin" Value="5"/>
  <Setter Property="BorderBrush" Value="#003366"/>
  <Setter Property="BorderThickness" Value="1"/>
  <Setter Property="Background" Value="White"/>
  <Setter Property="RowBackground" Value="White"/>
  <Setter Property="AlternatingRowBackground" Value="#F0F8FF"/>
  <Setter Property="GridLinesVisibility" Value="None"/>
  <Setter Property="HeadersVisibility" Value="Column"/>
  <Setter Property="ScrollViewer.CanContentScroll" Value="True"/>
  <Setter      Property="ScrollViewer.VerticalScrollBarVisibility"
Value="Auto"/>
  <Setter      Property="ScrollViewer.HorizontalScrollBarVisibility"
Value="Auto"/>
  <Setter Property="FontSize" Value="14"/>
</Style>

```

```

<Style x:Key="ActionButtonStyle" TargetType="Button">
  <Setter Property="Background" Value="#003366"/>
  <Setter Property="Foreground" Value="White"/>
  <Setter Property="FontSize" Value="14"/>
  <Setter Property="FontWeight" Value="SemiBold"/>
  <Setter Property="Height" Value="35"/>
  <Setter Property="Width" Value="100"/>
  <Setter Property="Margin" Value="5"/>
  <Setter Property="BorderThickness" Value="0"/>
  <Setter Property="Template">
    <Setter.Value>
      <ControlTemplate TargetType="Button">

```

```

        <Border Background="{TemplateBinding Background}"
            CornerRadius="5">
            <ContentPresenter HorizontalAlignment="Center"
                VerticalAlignment="Center"/>
        </Border>
    </ControlTemplate>
</Setter.Value>
</Setter>
<Style.Triggers>
    <Trigger Property="IsMouseOver" Value="True">
        <Setter Property="Background" Value="#004080"/>
        <Setter Property="Cursor" Value="Hand"/>
    </Trigger>
</Style.Triggers>
</Style>

<Style      x:Key="CancelButtonStyle"      TargetType="Button"
BasedOn="{StaticResource ActionButtonStyle}">
    <Setter Property="Background" Value="#D9534F"/>
    <Style.Triggers>
        <Trigger Property="IsMouseOver" Value="True">
            <Setter Property="Background" Value="#C9302C"/>
        </Trigger>
    </Style.Triggers>
</Style>
</Window.Resources>

<!-- Основной контейнер с синей окантовкой -->
<Grid>
    <Border BorderBrush="#003366"
        BorderThickness="5"
        CornerRadius="15"
        Background="White">

        <!-- Внутренний контейнер с содержимым -->
        <Border Background="White"

```

CornerRadius="10"

Margin="5"

Effect="{StaticResource ShadowEffect}">

<Grid>

<Grid.RowDefinitions>

<RowDefinition Height="Auto"/>

<RowDefinition Height="*/>

<RowDefinition Height="Auto"/>

</Grid.RowDefinitions>

<!-- Заголовок с логотипом -->

<StackPanel Grid.Row="0" Orientation="Horizontal"
HorizontalAlignment="Center" Margin="0,10">

<Image Source="/logo.png" Width="60" Height="35"
Margin="0,0,10,0"/>

<TextBlock Text="Управление задачами"
Style="{StaticResource HeaderTextStyle}"/>

</StackPanel>

<!-- Основное содержимое -->

<GroupBox Grid.Row="1" Header="Информация о задачах"
Margin="10,0,10,10"

BorderBrush="#003366" BorderThickness="1"
FontWeight="Bold" Foreground="#003366">

<Grid>

<Grid.RowDefinitions>

<RowDefinition Height="Auto"/>

<RowDefinition Height="Auto"/>

<RowDefinition Height="Auto"/>

<RowDefinition Height="Auto"/>

<RowDefinition Height="Auto"/>

<RowDefinition Height="Auto"/>

<RowDefinition Height="*/>

<RowDefinition Height="Auto"/>

</Grid.RowDefinitions>


```

<Grid.ColumnDefinitions>
    <ColumnDefinition Width="Auto"/>
    <ColumnDefinition Width="*/>
    <ColumnDefinition Width="Auto"/>
    <ColumnDefinition Width="*/>
</Grid.ColumnDefinitions>

<!-- Название задачи -->
<Label            Grid.Row="0"            Grid.Column="0"
Content="Название:" Style="{StaticResource LabelStyle}"/>
    <TextBox            Grid.Row="0"            Grid.Column="1"
Grid.ColumnSpan="3"      Style="{StaticResource      TextBoxStyle}"
Name="txtTaskTitle"/>

<!-- Проект -->
<Label            Grid.Row="1"            Grid.Column="0"
Content="Проект:" Style="{StaticResource LabelStyle}"/>
    <ComboBox            Grid.Row="1"            Grid.Column="1"
Style="{StaticResource      ComboBoxStyle}"      Name="cmbProjects"
DisplayMemberPath="Name"/>

<!-- Исполнитель -->
<Label            Grid.Row="1"            Grid.Column="2"
Content="Исполнитель:" Style="{StaticResource LabelStyle}"/>
    <ComboBox            Grid.Row="1"            Grid.Column="3"
Style="{StaticResource      ComboBoxStyle}"      Name="cmbEmployees"
DisplayMemberPath="FullName"/>

<!-- Описание -->
<Label            Grid.Row="2"            Grid.Column="0"
Content="Описание:" Style="{StaticResource LabelStyle}"/>
    <TextBox            Grid.Row="2"            Grid.Column="1"
Grid.ColumnSpan="3" Style="{StaticResource TextBoxStyle}" Height="60"
TextWrapping="Wrap"      AcceptsReturn="True"
VerticalScrollBarVisibility="Auto" Name="txtDescription"/>

```

```

<!-- Статус -->
<Label          Grid.Row="3"          Grid.Column="0"
Content="Статус:" Style="{StaticResource LabelStyle}"/>
<ComboBox      Grid.Row="3"          Grid.Column="1"
Style="{StaticResource ComboBoxStyle}" Name="cmbStatus">
    <ComboBoxItem Content="Новая"/>
    <ComboBoxItem Content="В работе"/>
    <ComboBoxItem Content="На проверке"/>
    <ComboBoxItem Content="Завершена"/>
    <ComboBoxItem Content="Отложена"/>
</ComboBox>

```

```

<!-- Приоритет -->
<Label          Grid.Row="3"          Grid.Column="2"
Content="Приоритет:" Style="{StaticResource LabelStyle}"/>
<ComboBox      Grid.Row="3"          Grid.Column="3"
Style="{StaticResource ComboBoxStyle}" Name="cmbPriority">
    <ComboBoxItem Content="Низкий"/>
    <ComboBoxItem Content="Средний"/>
    <ComboBoxItem Content="Высокий"/>
    <ComboBoxItem Content="Критический"/>
</ComboBox>

```

```

<!-- Срок выполнения -->
<Label Grid.Row="4" Grid.Column="0" Content="Срок
выполнения:" Style="{StaticResource LabelStyle}"/>
<DatePicker    Grid.Row="4"          Grid.Column="1"
Style="{StaticResource DatePickerStyle}" Name="dpDeadline"/>

```

```

<!-- Затраченное время -->
<Label          Grid.Row="4"          Grid.Column="2"
Content="Затраченное время (ч):" Style="{StaticResource LabelStyle}"/>
<TextBox        Grid.Row="4"          Grid.Column="3"
Style="{StaticResource TextBoxStyle}" Name="txtHoursWorked"/>

```

```

<!-- Список задач -->

```

```

        <Label Grid.Row="5" Grid.Column="0" Content="Список
задач:" Style="{StaticResource LabelStyle}" FontWeight="Bold"/>
        <DataGrid
            Grid.Row="6"
            Grid.Column="0"
            Grid.ColumnSpan="4"
            Style="{StaticResource
                DataGridStyle}"
            Name="dgTasks"

            AutoGenerateColumns="False"
            CanUserAddRows="False" SelectionMode="Single">
            <DataGrid.Columns>
                <DataGridTextColumn
                    Header="Название"
                    Binding="{Binding Title}" Width="*" Foreground="#003366"/>
                <DataGridTextColumn
                    Header="Проект"
                    Binding="{Binding ProjectName}" Width="120" Foreground="#003366"/>
                <DataGridTextColumn
                    Header="Исполнитель"
                    Binding="{Binding
                        AssigneeName}"
                    Width="150"
                    Foreground="#003366"/>
                <DataGridTextColumn
                    Header="Статус"
                    Binding="{Binding Status}" Width="100" Foreground="#003366"/>
                <DataGridTextColumn
                    Header="Приоритет"
                    Binding="{Binding Priority}" Width="100" Foreground="#003366"/>
                <DataGridTextColumn
                    Header="Срок"
                    Binding="{Binding
                        Deadline,
                        StringFormat={ } {0:dd.MM.yyyy} }"
                    Width="100" Foreground="#003366"/>
            </DataGrid.Columns>
        </DataGrid>

        <!-- Кнопки действий -->
        <StackPanel
            Grid.Row="7"
            Grid.Column="0"
            Grid.ColumnSpan="4"
            Orientation="Horizontal"
            HorizontalAlignment="Right" Margin="5">
            <Button Content="Удалить" Style="{StaticResource
                ActionButtonStyle}" Click="BtnDelete_Click"/>
        </StackPanel>
    </Grid>
</GroupBox>

<!-- Кнопки сохранения/отмены -->

```

```

        <StackPanel      Grid.Row="2"      Orientation="Horizontal"
HorizontalAlignment="Right" Margin="10">
            <Button      Content="Сохранить"      Style="{StaticResource
ActionButtonStyle}" Click="BtnSave_Click"/>
            <Button      Content="Отмена"      Style="{StaticResource
CancelButtonStyle}" Click="BtnCancel_Click"/>
        </StackPanel>
    </Grid>
</Border>
</Border>
</Grid>
</Window>

```

TasksWindow.xaml.cs

```

using System;
using System.Collections.Generic;
using System.Data;
using System.Windows;
using System.Windows.Controls;
using MySql.Data.MySqlClient;

```

```

namespace MyWpfApp

```

```

{
    public partial class TasksWindow : Window
    {
        private      string      connectionString      =
"server=localhost;user=root;database=pochta;port=3306;password=";
        private int currentTaskId = -1;

        public TasksWindow()
        {
            InitializeComponent();
            LoadProjects();
            LoadEmployees();
            LoadTasks();
        }
    }
}

```

```

private void LoadProjects()
{
    try
    {
        using (MySQLConnection connection = new
MySQLConnection(connectionString))
        {
            connection.Open();
            string query = "SELECT ProjectID, Name FROM Projects";
            MySqlCommand cmd = new MySqlCommand(query,
connection);
            MySqlDataAdapter adapter = new MySqlDataAdapter(cmd);
            DataTable dt = new DataTable();
            adapter.Fill(dt);

            cmbProjects.ItemsSource = dt.DefaultView;
            cmbProjects.SelectedValuePath = "ProjectID";
        }
    }
    catch (Exception ex)
    {
        MessageBox.Show($"Ошибка при загрузке проектов:
{ex.Message}");
    }
}

```

```

private void LoadEmployees()
{
    try
    {
        using (MySQLConnection connection = new
MySQLConnection(connectionString))
        {
            connection.Open();
            string query = "SELECT EmployeeID, CONCAT(FirstName, ' ',
LastName) AS FullName FROM Employees";

```

```

        MySqlCommand cmd = new MySqlCommand(query,
connection);
        MySqlDataAdapter adapter = new MySqlDataAdapter(cmd);
        DataTable dt = new DataTable();
        adapter.Fill(dt);

        cmbEmployees.ItemsSource = dt.DefaultView;
        cmbEmployees.SelectedValuePath = "EmployeeID";
    }
}
catch (Exception ex)
{
    MessageBox.Show($"Ошибка при загрузке сотрудников:
{ex.Message}");
}

private void LoadTasks()
{
    try
    {
        using (MySqlConnection connection = new
MySqlConnection(connectionString))
        {
            connection.Open();
            string query = @"SELECT t.TaskID, t.Title, p.Name AS
ProjectName,
                        CONCAT(e.FirstName, ' ', e.LastName) AS
AssigneeName,
                        t.Status, t.Priority, t.Deadline
FROM Tasks t
LEFT JOIN Projects p ON t.ProjectID = p.ProjectID
LEFT JOIN Employees e ON t.AssigneeID =
e.EmployeeID";
            MySqlCommand cmd = new MySqlCommand(query,
connection);

```

```

        MySqlDataAdapter adapter = new MySqlDataAdapter(cmd);
        DataTable dt = new DataTable();
        adapter.Fill(dt);

        dgTasks.ItemsSource = dt.DefaultView;
    }
}
catch (Exception ex)
{
    MessageBox.Show($"Ошибка при загрузке задач:
{ex.Message}");
}
}

private void BtnAdd_Click(object sender, RoutedEventArgs e)
{
    // Сброс формы для добавления новой задачи
    currentTaskId = -1;
    txtTaskTitle.Clear();
    txtDescription.Clear();
    cmbStatus.SelectedIndex = 0;
    cmbPriority.SelectedIndex = 1;
    dpDeadline.SelectedDate = DateTime.Now.AddDays(7);
    txtHoursWorked.Clear();
}

private void BtnDelete_Click(object sender, RoutedEventArgs e)
{
    if (dgTasks.SelectedItem == null)
    {
        MessageBox.Show("Выберите задачу для удаления");
        return;
    }

    var result = MessageBox.Show("Вы уверены, что хотите удалить
выбранную задачу?", "Подтверждение",

```

```

        MessageBoxButton.YesNo,
        MessageBoxImage.Question);
        if (result == MessageBoxResult.Yes)
        {
            try
            {
                DataRowView row = (DataRowView)dgTasks.SelectedItem;
                int taskId = Convert.ToInt32(row["TaskID"]);

                using (SqlConnection connection = new
                MySqlConnection(connectionString))
                {
                    connection.Open();
                    string query = "DELETE FROM Tasks WHERE TaskID =
@TaskID";
                    MySqlCommand cmd = new MySqlCommand(query,
connection);
                    cmd.Parameters.AddWithValue("@TaskID", taskId);
                    cmd.ExecuteNonQuery();
                }

                LoadTasks();
            }
            catch (Exception ex)
            {
                MessageBox.Show($"Ошибка при удалении задачи:
{ex.Message}");
            }
        }

private void BtnSave_Click(object sender, RoutedEventArgs e)
{
    if (string.IsNullOrEmpty(txtTaskTitle.Text))
    {
        MessageBox.Show("Введите название задачи");
    }
}

```



```

        return;
    }

    if (cmbProjects.SelectedValue == null)
    {
        MessageBox.Show("Выберите проект");
        return;
    }

    if (cmbEmployees.SelectedValue == null)
    {
        MessageBox.Show("Выберите исполнителя");
        return;
    }

    if (dpDeadline.SelectedDate == null)
    {
        MessageBox.Show("Укажите срок выполнения");
        return;
    }

    try
    {
        using (MySqlConnection connection = new
        MySqlConnection(connectionString))
        {
            connection.Open();

            if (currentTaskId == -1)
            {
                // Добавление новой задачи
                string query = @"INSERT INTO Tasks (ProjectID, Title,
                Description, AssigneeID, Status, Priority, Deadline, CreatedDate)
                VALUES (@ProjectID, @Title, @Description,
                @AssigneeID, @Status, @Priority, @Deadline, NOW())";
                MySqlCommand cmd = new MySqlCommand(query,

```

```

connection);
        cmd.Parameters.AddWithValue("@ProjectID",
cmbProjects.SelectedValue);
        cmd.Parameters.AddWithValue("@Title", txtTaskTitle.Text);
        cmd.Parameters.AddWithValue("@Description",
txtDescription.Text);
        cmd.Parameters.AddWithValue("@AssigneeID",
cmbEmployees.SelectedValue);
        cmd.Parameters.AddWithValue("@Status",
((ComboBoxItem)cmbStatus.SelectedItem).Content);
        cmd.Parameters.AddWithValue("@Priority",
((ComboBoxItem)cmbPriority.SelectedItem).Content);
        cmd.Parameters.AddWithValue("@Deadline",
dpDeadline.SelectedDate);

        cmd.ExecuteNonQuery();

        // Если указано затраченное время, добавляем запись в
TimeLogs
        if (!string.IsNullOrEmpty(txtHoursWorked.Text) &&
            decimal.TryParse(txtHoursWorked.Text, out decimal
hours))
        {
            int newTaskId = (int)cmd.LastInsertedId;
            query = @"INSERT INTO TimeLogs (EmployeeID,
TaskID, HoursWorked, LogDate, Description)
                    VALUES (@EmployeeID, @TaskID,
@HoursWorked, NOW(), @Description)";
            cmd = new MySqlCommand(query, connection);
            cmd.Parameters.AddWithValue("@EmployeeID",
cmbEmployees.SelectedValue);
            cmd.Parameters.AddWithValue("@TaskID", newTaskId);
            cmd.Parameters.AddWithValue("@HoursWorked", hours);
            cmd.Parameters.AddWithValue("@Description",
"Первоначальная оценка времени");
            cmd.ExecuteNonQuery();

```

```

        }
    }
    else
    {
        // Обновление существующей задачи
        string query = @"UPDATE Tasks SET
            ProjectID = @ProjectID,
            Title = @Title,
            Description = @Description,
            AssigneeID = @AssigneeID,
            Status = @Status,
            Priority = @Priority,
            Deadline = @Deadline
            WHERE TaskID = @TaskID";
        MySqlCommand cmd = new MySqlCommand(query,
connection);
        cmd.Parameters.AddWithValue("@ProjectID",
cmbProjects.SelectedValue);
        cmd.Parameters.AddWithValue("@Title", txtTaskTitle.Text);
        cmd.Parameters.AddWithValue("@Description",
txtDescription.Text);
        cmd.Parameters.AddWithValue("@AssigneeID",
cmbEmployees.SelectedValue);
        cmd.Parameters.AddWithValue("@Status",
((ComboBoxItem)cmbStatus.SelectedItem).Content);
        cmd.Parameters.AddWithValue("@Priority",
((ComboBoxItem)cmbPriority.SelectedItem).Content);
        cmd.Parameters.AddWithValue("@Deadline",
dpDeadline.SelectedDate);
        cmd.Parameters.AddWithValue("@TaskID", currentTaskId);

        cmd.ExecuteNonQuery();
    }
}

LoadTasks();

```

```

        MessageBox.Show("Данные сохранены успешно");
    }
    catch (Exception ex)
    {
        MessageBox.Show($"Ошибка при сохранении задачи:
{ex.Message}");
    }
}

private void BtnCancel_Click(object sender, RoutedEventArgs e)
{
    this.Close();
}

private void DgTasks_SelectionChanged(object sender,
System.Windows.Controls.SelectionChangedEventArgs e)
{
    if (dgTasks.SelectedItem != null)
    {
        DataRowView row = (DataRowView)dgTasks.SelectedItem;
        currentTaskId = Convert.ToInt32(row["TaskID"]);
        txtTaskTitle.Text = row["Title"].ToString();
        txtDescription.Text = row["Description"].ToString();

        // Установка проекта
        foreach (DataRowView item in cmbProjects.Items)
        {
            if (item["Name"].ToString() == row["ProjectName"].ToString())
            {
                cmbProjects.SelectedItem = item;
                break;
            }
        }

        // Установка исполнителя
        foreach (DataRowView item in cmbEmployees.Items)

```

```

        {
            if (item["FullName"].ToString() ==
row["AssigneeName"].ToString())
            {
                cmbEmployees.SelectedItem = item;
                break;
            }
        }

// Установка статуса
foreach (ComboBoxItem item in cmbStatus.Items)
{
    if (item.Content.ToString() == row["Status"].ToString())
    {
        cmbStatus.SelectedItem = item;
        break;
    }
}

// Установка приоритета
foreach (ComboBoxItem item in cmbPriority.Items)
{
    if (item.Content.ToString() == row["Priority"].ToString())
    {
        cmbPriority.SelectedItem = item;
        break;
    }
}

// Установка срока выполнения
if (row["Deadline"] != DBNull.Value)
{
    dpDeadline.SelectedDate =
Convert.ToDateTime(row["Deadline"]);
}
}

```

}

}

}

Дипломный проект выполнен мной совершенно самостоятельно. Все использованные в проекте материалы и концепции из опубликованной литературы и других источников имеют ссылки на них. Дипломный проект прошел проверку на корректность заимствования в системе «Антиплагиат.ВУЗ».

Настоящим подтверждаю, что даю разрешение Университету «Синергия» на размещение полного текста моего дипломного проекта, отзыва на дипломный проект в электронно-библиотечной системе Университета «Синергия».



Заболотный Д.А.

подпись

Фамилия И.О.